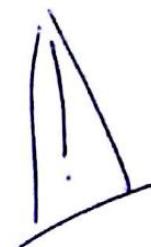


# Master – 1<sup>ère</sup> année – Informatique

## Cryptographie – Examen

15 janvier 2014



### Consignes :

- Durée : 2h.
- Documents interdits.
- Aucun accès à un téléphone portable, une calculatrice, un PDA ou tout autre dispositif électronique, connectable ou non.

### Exercice 1 (8 points) [Questions diverses]

1. Une recherche exhaustive sur les 56 bits d'une clé DES nécessite environ 56 heures. Combien de temps faudrait-il approximativement sur une clé de 64 bits ?  
 56 heures       64 heures       64 jours       plus d'un an
2. Alice a utilisé le chiffrement "one-time pad" pour envoyer un message  $m \in \{0, 1\}^{100}$  à Bob. Ils partageaient tous deux une clé aléatoire  $k \in \{0, 1\}^{100}$ . Charlie intercepte le chiffré  $c = m \oplus k$ . Quel est le temps nécessaire pour retrouver  $m$  ?  
 instantané       100 essais        $2^{100}$  essais       impossible
3. Une implémentation de RSA, utilisant des exposants public et privé aléatoires de la taille du module, annonce le temps de calcul suivant : 1 milliseconde pour chiffrer un message de 512 bits avec une clé de 512 bits. Sachant que cette implémentation utilise l'algorithme d'exponentiation vu en cours, quel temps nécessiterait le chiffrement RSA d'un message de 2048 bits avec une clé de 2048 bits (en millisecondes) ?  
 1       8       16       32       64       128
4. Afin de pouvoir distinguer ses communications personnelles de ses communications professionnelles, Alice utilise deux clés publiques RSA, ses correspondants utilisant l'une ou l'autre selon le type de communication. Afin d'accélérer la génération de clés, Alice ne choisit que trois grands nombres premiers  $p, q$  et  $r$  de 512 bits, qu'elle garde secrets. Ses deux modules RSA publics sont alors  $N_1 = pq$  et  $N_2 = qr$ . Alice choisit aléatoirement deux couples d'exposants privés et publics  $(d_1, e_1)$  et  $(d_2, e_2)$  vérifiant donc  $e_1 d_1 \equiv 1 \pmod{(p-1)(q-1)}$  et  $e_2 d_2 \equiv 1 \pmod{(q-1)(r-1)}$ . Quelle est la sécurité obtenue ?  
 impossible à déterminer       identique au RSA traditionnel       aucune sécurité

5. Une technique classique d'identification est le "mot de passe". Si on a confiance dans le serveur, cette technique

- est sûre une fois       résiste aux attaques passives       est zero-knowledge<sup>1</sup>

6. Un problème du mode CBC est qu'aucun parallélisme n'est possible. Il faut connaître le chiffré du bloc précédent. Une proposition est de séparer l'ensemble des blocs en par exemple deux groupes : ceux de numéro pair et ceux de numéro impair. On fait donc deux CBC en parallèle, l'un avec la clé  $k_1$ , et l'initialisation  $IV_1$ , l'autre avec  $k_2$  et  $IV_2$ . Comment garder la sécurité de CBC ?

- Je peux utiliser la même clé et le même IV  
 Je peux utiliser la même clé mais pas le même IV  
 Les clés doivent être différentes et les IV aussi  
 Je ne peux pas avoir la sécurité d'un unique CBC

7. Pour la signature de longs messages, on utilise habituellement une fonction de hachage destinée à transformer le message avant signature. Sachant qu'une borne supérieure pour une recherche exhaustive se situe vers  $2^{80}$ , quelle longueur de haché doit-on préconiser pour éviter les contrefaçons ?

- 40 bits       80 bits       128 bits       160 bits

8. Alice connaît la factorisation de  $n = pq$ , et elle seule. Pour prouver son identité, elle accepte, de qui le souhaite, un nombre  $y = x^2 \pmod{n}$ . Puis elle retourne une racine carrée de  $y$ , ce qu'elle seule peut calculer. Ce système d'authentification résiste

- à rien       aux attaques passives       est zero-knowledge

### Exercice 2 (Construction d'une fonction de hachage) (4 points)

1. Rappeler la définition d'une fonction à collisions fortes difficiles, d'une fonction à collisions faibles difficiles, d'une fonction à sens unique.

2. Soient  $\mathcal{E}$  un ensemble fini, et  $f_0$  et  $f_1$  deux permutations sur  $\mathcal{E}$  (i.e. deux fonctions bijectives sur  $\mathcal{E}$ ) à sens unique. On appelle *rencontre* entre  $f_0$  et  $f_1$  tout couple  $(a, b) \in \mathcal{E} \times \mathcal{E}$  tel que  $f_0(a) = f_1(b)$ . On dit que les fonctions  $f_0$  et  $f_1$  ne se rencontrent pas s'il est algorithmiquement difficile de trouver une rencontre entre  $f_0$  et  $f_1$ .

À partir d'une paire de telles fonctions à sens unique qui ne se rencontrent pas, et d'un élément  $e \in \mathcal{E}$  fixé, on construit une fonction  $h$  sur l'ensemble de toutes les chaînes binaires finies  $\{0, 1\}^*$  de la façon suivante :

$$h : \begin{array}{ccc} \{0, 1\}^* & \longrightarrow & \mathcal{E} \\ x = x_1 x_2 \dots x_k & \mapsto & f_{x_1}(f_{x_2}(\dots f_{x_k}(e) \dots)) \end{array}$$

Montrer que la fonction  $h$  ainsi construite est à collisions fortes difficiles.

<sup>1</sup>On rappelle qu'un algorithme d'authentification est dit "zero-knowledge" si le fait de l'utiliser (pour s'authentifier) ne révèle rien sur ses propres secrets.

$$\log_{3e} 800 = \frac{\ln 800}{\ln 3e}$$

$$\log(2^x)$$

### Exercice 3 (Algorithmie AES) (4 points)

1. Rappeler comment est défini le produit de deux octets dans l'algorithme AES.
2. Calculer  $\{C5\} \times \{B2\}$ .

### Exercice 4 (Chiffrement/déchiffrement RSA) (4 points)

Dans toute la suite, on pourra utiliser les résultats numériques suivants :

- $319 = 11 \times 29$ ;  $10^{11} = 263 \pmod{319}$ ;  $263^2 = 216 \times 319 + 265$ ;

- $133^3 = 12 \pmod{319}$ ;  $133^{25} = 133 \pmod{319}$ ;

- $11^2 = 121 \pmod{280}$ ;  $11^4 = 81 \pmod{280}$ ;  $11^8 = 121 \pmod{280}$ ;  $11^{16} = 81 \pmod{280}$ ;  $(11^{16}) \cdot (11^{-1}) \cdot (11^{-1}) \equiv 1 \pmod{280}$

- $95 = 64 + 31$ ;  $81 \times 11 = 51 \pmod{280}$ ;  $81 \times 121 = 1 \pmod{280}$ .

On considère la clé publique RSA  $(11, 319)$ , c'est-à-dire pour  $n = 319$  et  $e = 11$ .

1. Quel est le chiffrement avec cette clé du message  $M = 100$ ?
2. Calculer  $d$  la clé privée correspondant à la clé publique  $e$ .
3. Déchiffrer le message  $C = 133$ .

$$(n, e) \text{ clé publique}$$

$$d \text{ clé privée}$$

$$ed = 1 \pmod{n}$$

$$(e \cdot n) \cdot (p-1) \cdot (q-1)$$

$$C = M^e \pmod{n}$$

$$M = C^d \pmod{n}$$

$$\begin{aligned} ① \quad C &= 100^{11} \pmod{319} = 265 \\ &= (10^2)^{11} \pmod{319} \\ &= (10^2)^2 \pmod{319} \\ &= (263)^2 \pmod{319} \\ &= 265 \pmod{319} \\ &= 265 \end{aligned}$$

$$\begin{aligned} ② \quad M &= C^d \pmod{n} \\ &= 265^d \pmod{319} \\ &= 100 \pmod{319} \\ m &= p \cdot q \\ 319 &= 11 \times 29 \\ \varphi(n) &= (11-1) \times (29-1) \\ &= 280 \end{aligned}$$

$$\begin{aligned} e \cdot d &= 1 \pmod{\varphi(n)} \\ d \times 11 &= 1 \pmod{280} \\ \cancel{d} \times \cancel{11} &= 1 \pmod{280} \\ \cancel{11} &= 1 \pmod{280} \\ \cancel{11} &= -1 \pmod{280} \\ \cancel{11} &= 279 \pmod{280} \\ \cancel{11} &= 1 \pmod{280} \\ \text{on a } 81 \times 121 &= 1 \pmod{280} \\ \Rightarrow 81 \times 11 \times 11 &= 1 \pmod{280} \\ \Rightarrow \text{d'après } (81 \times 11 = 81 \pmod{280}) \quad & \\ \text{on a } 81 \times 11 &= 1 \pmod{280} \\ \Rightarrow d \times 11 &= 1 \pmod{280} \\ \text{et } d = 265^{-1} & \\ 3^0 & \\ n &= 133 \pmod{319} \\ &= 133 \times 133^{29} \pmod{319} \\ &= 133 \times (133^2)^{14} \pmod{319} \\ &= 133 \times (133^2)^2 \pmod{319} \\ &= 133^3 \pmod{319} \\ &= 12 \pmod{319} \quad \text{ID} \quad M = 12 \end{aligned}$$



## EXAMEN DE CRYPTOGRAPHIE

Master 1 Informatique

18 Janvier 2012  
(Durée : 2h)

- Le barème est donné uniquement à titre indicatif.
- Tous les documents physiques sont autorisés.
- Tous les types de calculatrice sont autorisées (sauf les téléphones).
- Les téléphones et les ordinateurs (et plus généralement tout les appareils communiquants) même utilisés comme calculette ou comme montre sont formellement interdits. Toute utilisation d'un téléphone sera considérée comme une tentative de fraude et traitée comme telle.

### Exercice 1 : Linear Feedback Shift Register — LFSR (6 pts)

Considérons un système de chiffrement à flot constitué d'un simple LFSR  $\mathcal{R}$  dont le polynôme caractéristique est :

$$P(X) = 1 + X + X^4$$

La clé secrète utilisée pour le chiffrement est  $k = k_3k_2k_1k_0 = 1101$  et constitue la graine du générateur pseudo-aléatoire correspondant au LFSR  $\mathcal{R}$ , produisant la suite chiffrante  $(s_i)_{i \in \mathbb{N}}$ .  
Un mot est transformé en binaire en prenant simplement comme encodage pour une lettre l'équivalent binaire de son rang dans l'alphabet, i.e.

A(1)	→	00001
B(2)	→	00010
C(3)	→	00011
	⋮	
X(24)	→	11000
Y(25)	→	11001
Z(26)	→	11010

Considérons le chiffré reçu suivant :

$$C = c_0 \dots c_{14} = 000001111001000.$$

**Question 1.1** Rappelez le principe du chiffrement à flot à l'aide d'un schéma. Décrivez précisément comment sont obtenus les bits  $c_0 \dots c_{14}$ .

**Question 1.2** De combien de lettres est formé le message clair ?

**Question 1.3** Quelle est la période maximale de  $\mathcal{R}$  ?

**Question 1.4** Représentez  $\mathcal{R}$  sur un schéma.

**Question 1.5** Donner la suite produite par ce LFSR.

**Question 1.6** Que pouvez-vous dire de la sécurité de ce système ?

## Exercice 2 : Un autre mauvais MAC (4 pts)

Considérons le schéma de MAC opérant sur des mots de longueur paire de taille  $2n$  ou plus. Soit  $E$  un algorithme de chiffrement par blocs (de  $n$  bits), et soit  $f$  une fonction de hachage (à collisions fortes difficiles) dont la sortie fait  $n$  bits. Alors, pour tout message  $m = m_1||m_2$  avec  $m_1$  et  $m_2$  de même taille, on obtient le MAC de la façon suivante :

$$\text{MAC}_k(m) = \begin{cases} E_k(f(m_1) \oplus f(m_2)) & \text{si } |m| > 2n \\ E_k(m_1 \oplus m_2) & \text{sinon } (|m| = 2n) \end{cases}$$

**Question 2.1** Montrez que ce MAC à la même faiblesse que celui vu en TD.

**Question 2.2** Comment modifier la construction du MAC pour éviter cette faiblesse ?

**Question 2.3** Montrez que si on connaît le MAC de  $m_1||m_1$ , alors il est possible de construire  $2^n - 1$  messages ayant le même MAC.  
s'il le  $2^n$  de la même taille

**Question 2.4** Comment modifier la construction du MAC pour éviter cette faiblesse ?

## Exercice 3 : Cryptosystème homomorphique de Paillier (10 pts)

En 1999, Pascal Paillier a proposé un schéma de chiffrement basé sur un problème cryptographiquement difficile : le problème des résidus composés. Dans la suite, le PGCD de deux nombres  $a$  et  $b$  sera noté  $a \wedge b$  et l'indicatrice d'Euler sera désigné par  $\varphi$ . Le fonctionnement du schéma de Paillier est détaillé ci-dessous :

**Génération des clés :**

1. On tire aléatoirement  $p$  et  $q$ , deux grands entiers premiers dont les tailles sont fixées par un paramètre de sécurité et tels que, si  $n = p * q$ , alors  $n \wedge \varphi(n) = 1$ ;
2. On calcule  $n = p * q$  et  $\varphi(n) = (p - 1)(q - 1)$ ;
3. On pose ensuite  $g = n + 1$ ;
4. On calcule  $\mu \equiv \left( \frac{(g^{\varphi(n)} \bmod n^2) - 1}{n} \right)^{-1} [n]$
5. La clé publique est la paire  $(n, g)$ ;
6. La clé privée est la paire  $(\varphi(n), \mu)$ .

**Chiffrement :**

Les messages sont des éléments de  $\mathbb{Z}/n\mathbb{Z}$ , pour chiffrer un message  $m$ , on choisit aléatoirement un élément  $r$  de  $\mathbb{Z}/n\mathbb{Z}$ . Le texte chiffré est alors :

$$c \equiv g^m r^n [n^2].$$

Dans toute la suite on notera  $\mathcal{E}$  la fonction de chiffrement.

**Déchiffrement :**

Pour déchiffrer un message, on calcule :

$$m \equiv \left( \frac{(c^{\varphi(n)} \bmod n^2) - 1}{n} \right) \cdot \mu [n].$$

Dans toute la suite on notera  $\mathcal{D}$  la fonction de déchiffrement.

### 3.1 Fonctionnement du schéma

**Question 3.1** Rappelez le principe de la cryptographie à clef publique. Vous prendrez soin de bien expliquer la différence et les avantages par rapport à la cryptographie symétrique.

**Question 3.2** Rappelez la définition d'un problème cryptographiquement difficile.

**Question 3.3** Dans ce cryptosystème, on utilise un entier naturel  $n$  qui est le produit de deux entiers premiers  $p$  et  $q$  que l'on doit choisir de sorte que  $n \wedge \varphi(n) = 1$ .

1. Rappelez la définition et la formule de l'indicatrice d'Euler ;
2. La condition est-elle vérifiée lorsque  $p = 2$  et  $q = 31$ ? Et lorsque  $p = 29$  et  $q = 31$ ?
3. Si  $n \wedge \varphi(n) = d \neq 1$ , que pouvez-vous dire de  $n \pmod{d}$ ? En déduire que  $d = p$  ou  $d = q$ ;
4. Toujours si  $n \wedge \varphi(n) = d \neq 1$ , que pouvez-vous dire de  $\varphi(n) \pmod{n}$ ? En déduire que  $p \mid (q - 1)$  ou  $q \mid (p - 1)$ .
5. En déduire une condition pour que  $n$  et  $\varphi(n)$  soient premiers entre eux.

**Question 3.4** Afin de montrer que ce schéma définit un cryptosystème, il faut montrer que :

$$\forall m \in \mathbb{Z}/n\mathbb{Z} : \mathcal{D}(\mathcal{E}(m)) = m.$$

1. Calculez la valeur de  $\varphi(15^2)$ ;
2. Donnez la valeur de  $\varphi(n^2)$  en fonction de  $n$  et de  $\varphi(n)$ ;
3. Déduisez-en que  $c^{\varphi(n)} \equiv g^{m\varphi(n)} \pmod{n^2}$ ;
4. Montrez que  $(1 + n)^\alpha \equiv 1 + n\alpha \pmod{n^2}$  avec  $\alpha \in \mathbb{Z}$ ;
5. Montrez que  $0 \leq 1 + n\varphi(n) < n^2$  puis que  $\mu \equiv (\varphi(n))^{-1} \pmod{n}$ ;
6. Montrez finalement que le schéma proposé par Paillier définit un cryptosystème.

**Question 3.5** Montrez que si on sait factoriser le produit de deux nombres premiers alors on casse complètement le cryptosystème de Paillier.

### 3.2 Vous avez dit « homomorphique » ?

Dans toute la suite  $m_1$  et  $m_2$  désigne deux messages qui seront chiffrés avec des valeurs aléatoires notées  $r_1$  et  $r_2$  respectivement.

**Question 3.6** Montrez que :

1.  $\mathcal{D}(\mathcal{E}(m_1, r_1) \cdot g^{m_2}) \equiv m_1 + m_2 \pmod{n}$ ;
2.  $\mathcal{D}(\mathcal{E}(m_1, r_1) \cdot \mathcal{E}(m_2, r_2)) \equiv m_1 + m_2 \pmod{n}$ ;
3.  $\forall k \in \mathbb{N} : \mathcal{D}(\mathcal{E}(m_1, r_1)^k) \equiv km_1 \pmod{n}$ .

**Question 3.7** En mathématiques, un homomorphisme de groupes est une application (notée  $f$  dans la suite) entre deux groupes  $(G_1, \oplus)$  et  $(G_2, \otimes)$  telle que :

$$\forall x, y \in G_1, f(x \oplus y) = f(x) \otimes f(y).$$

On note  $\mathcal{P}$  l'ensemble des messages clairs et  $\mathcal{C}$  l'ensemble des messages chiffrés. En admettant que  $\mathcal{P}$  et  $\mathcal{C}$  sont des groupes, montrez que la fonction de déchiffrement du schéma de Paillier définit un homomorphisme de groupes.

### 3.3 Utilisation des propriétés d'homomorphisme

De manière générale, ce type de propriété n'est pas souhaitable en cryptographie puisque cela rend le cryptosystème malléable (c'est à dire qu'il est possible de créer un nouveau chiffre  $c'$  valide à partir d'un chiffre  $c$ , sans déchiffrer ce dernier). Cependant, une telle propriété permet de réaliser des calculs sur des données sans avoir besoin de les révéler. Les applications sont très nombreuses et parmi lesquelles on trouve le vote électronique, par exemple, ou la délégation de calculs (calculs « In the Cloud » ou CaS – Computing as a Service).

**Question 3.8** Montrez comment Alice peut faire calculer  $a + b$  et  $k \times a$  à Bob, en ne lui révélant que  $\mathcal{E}(a)$ ,  $\mathcal{E}(b)$  et  $k$ .

**Question 3.9** Que manque-t-il pour qu'Alice puisse déléguer tous ses calculs à Bob ?

**Question 3.10** Du point de vue d'une entreprise, il existe souvent plusieurs types de traitements : les traitements généraux — comme par exemple le calcul des itches de paye — et les traitements relevant d'un « processus métier », correspondant au savoir-faire de l'entreprise — comme par exemple, le contrôle d'une machine-outil. Critiquez la méthode précédente, du point de vue de la délégation d'un « processus métier ». Que faudrait-il pour permettre une délégation complète des calculs ?

# Examen

## Master Informatique 1<sup>re</sup> année

Cryptographie - Examen

6 janvier 2010

Consignes :

- Durée : 2h.
- Documents interdits.
- Aucun accès à un téléphone portable, une calculatrice, un PDA ou tout autre dispositif électronique, connectable ou non.

### Questions de cours (6 points)

#### 1. L'algorithme DES *Déjà Encrypthon standard*

- Quelle taille de clé utilise-t-on pour le DES ?  $K \leq 56$  bits
- Quelle est la complexité d'une attaque par recherche exhaustive sur le DES :
  - Dans le pire cas ?
  - En moyenne ?
- Comment pouvons-nous améliorer cette attaque?
  - Quelle est la complexité dans le pire cas?
  - En moyenne?

#### 2. L'algorithme AES

- Quelles sont les tailles de l'entrée, de la sortie et de la clé dans l'algorithme AES ? Combien y a-t-il de tours ?
- Dans l'algorithme AES, rappeler comment est défini le produit de deux octets. [Rappel : le polynôme  $X^8 + X^4 + X^3 + X + 1$  intervient dans la définition]

#### 3. Fonctions de hachage

- Rappeler la définition d'une fonction à collisions fortes difficiles.
- On suppose qu'une fonction  $f$ , prenant en entrée des messages de taille quelconque, donne une sortie de  $n$  bits. Donner une condition nécessaire sur  $n$  pour que  $f$  soit à collisions fortes difficiles. Justifier votre réponse le plus précisément possible.

- i. Quelle est la complexité dans le pire cas?
- ii. En moyenne?

## 2. L'algorithme AES

- (a) Quelles sont les tailles de l'entrée, de la sortie et de la clé dans l'algorithme AES ? Combien y a-t-il de tours ?
- (b) Dans l'algorithme AES, rappeler comment est défini le produit de deux octets. [Rappel : le polynôme  $X^8 + X^4 + X^3 + X + 1$  intervient dans la définition]

## 3. Fonctions de hachage

- (a) Rappeler la définition d'une fonction à collisions fortes difficiles.
- (b) On suppose qu'une fonction  $f$ , prenant en entrée des messages de taille quelconque, donne une sortie de  $n$  bits. Donner une condition nécessaire sur  $n$  pour que  $f$  soit à collisions fortes difficiles.  
Justifier votre réponse le plus précisément possible.

125, 152, 156 :

IP faut trouver  
d' le P que

$$E(n) \times 1 + dx e = 1$$

**Exercice 1** (Une attaque sur le mode CBC) (5 points)

Une fonction de chiffrement sur des blocs de  $n$  bits  $E : (K, x) \mapsto y = E(K, x)$  est utilisée en mode CBC pour chiffrer un message  $m = (m_1, \dots, m_t)$ . On obtient un cryptogramme  $c = (IV, c_1, \dots, c_t)$ , où  $IV$  est le vecteur d'initialisation.

1. Montrer que si deux blocs du cryptogramme,  $c_i$  et  $c_j$  sont identiques, avec  $1 \leq i < j \leq t$ , alors on dispose d'une information sur les blocs de clairs  $m_i$  et  $m_j$  (montrer que  $m_i \oplus m_j$  est connu).
2. On suppose que les blocs de cryptogramme sont répartis aléatoirement. Quelle est la probabilité pour qu'il existe deux blocs de  $n$  bits égaux dans un cryptogramme de  $t$  blocs ? Quelle condition doit satisfaire la taille  $n$  des blocs pour que cette probabilité soit inférieure à  $1/1000$  sur des cryptogrammes d'un million de blocs ?

← **Exercice 2** (Construction d'une fonction de hachage) (4 points)

Soient  $\mathcal{E}$  un ensemble fini, et  $f_0$  et  $f_1$  deux permutations sur  $\mathcal{E}$  (i.e. deux fonctions bijectives sur  $\mathcal{E}$ ) à sens unique. On appelle *rencontre* entre  $f_0$  et  $f_1$  tout couple  $(a, b) \in \mathcal{E} \times \mathcal{E}$  tel que  $f_0(a) = f_1(b)$ . On dit que les fonctions  $f_0$  et  $f_1$  ne se rencontrent pas s'il est algorithmiquement difficile de trouver une rencontre entre  $f_0$  et  $f_1$ .

À partir d'une paire de telles fonctions à sens unique qui ne se rencontrent pas, et d'un élément  $e \in \mathcal{E}$  fixé, on construit une fonction  $h$  sur l'ensemble de toutes les chaînes binaires finies  $\{0, 1\}^*$  de la façon suivante :

$$h : \begin{array}{ccc} \{0, 1\}^* & \longrightarrow & \mathcal{E} \\ r = r_1 r_2 \dots r_k & \longmapsto & f_{r_1}(f_{r_2}(\dots f_{r_k}(e) \dots)) \end{array}$$

Montrer que la fonction  $h$  ainsi construite est à collisions fortes difficiles.



← **Exercice 3** (Signature RSA) (5 points)

1. Calculer le module  $n$  et l'entier  $\varphi(n) = (p - 1)(q - 1)$  associés aux nombres premiers  $p = 17$  et  $q = 23$ .
2. Quels sont les exposants secrets de signature associés aux exposants publics  $e = 11$  et  $e = 13$  ?
3. Quelle est la signature de  $m = 100$  ?
4. Vérifier que la vérification fonctionne.

$$111 p = 17, q = 23$$

$$\varphi(n) = (17 - 1)(23 - 1) = 352$$

$$n = 17 \times 23 = 391$$

$$e // d = ?$$

$$(ed = 1 \bmod \varphi(n)) = (11d = 1 \bmod 352)$$

$$d = \frac{1}{11} \bmod 352 = ?$$

$$d'' = \frac{1}{11} \bmod 352 = ?$$

$$3 // n = 100, c = ?$$

2

ip faut trouver  
de la P que

$$\mathcal{E}(n) \times t + dx = 1$$

**Exercice 1** (Une attaque sur le mode CBC) (5 points)

Une fonction de chiffrement sur des blocs de  $n$  bits  $E : (K, x) \mapsto y = E(K, x)$  est utilisée en mode CBC pour chiffrer un message  $m = (m_1, \dots, m_t)$ . On obtient un cryptogramme  $c = (IV, c_1, \dots, c_t)$ , où  $IV$  est le vecteur d'initialisation.

1. Montrer que si deux blocs du cryptogramme,  $c_i$  et  $c_j$  sont identiques, avec  $1 \leq i < j \leq t$ , alors on dispose d'une information sur les blocs de clairs  $m_i$  et  $m_j$  (montrer que  $m_i \oplus m_j$  est connu).
2. On suppose que les blocs de cryptogramme sont répartis aléatoirement. Quelle est la probabilité pour qu'il existe deux blocs de  $n$  bits égaux dans un cryptogramme de  $t$  blocs ? Quelle condition doit satisfaire la taille  $n$  des blocs pour que cette probabilité soit inférieure à  $1/1000$  sur des cryptogrammes d'un million de blocs ?

**Exercice 2** (Construction d'une fonction de hachage) (4 points)

Soient  $\mathcal{E}$  un ensemble fini, et  $f_0$  et  $f_1$  deux permutations sur  $\mathcal{E}$  (i.e. deux fonctions bijectives sur  $\mathcal{E}$ ) à sens unique. On appelle rencontre entre  $f_0$  et  $f_1$  tout couple  $(a, b) \in \mathcal{E} \times \mathcal{E}$  tel que  $f_0(a) = f_1(b)$ . On dit que les fonctions  $f_0$  et  $f_1$  ne se rencontrent pas s'il est algorithmiquement difficile de trouver une rencontre entre  $f_0$  et  $f_1$ .

À partir d'une paire de telles fonctions à sens unique qui ne se rencontrent pas, et d'un élément  $e \in \mathcal{E}$  fixé, on construit une fonction  $h$  sur l'ensemble de toutes les chaînes binaires finies  $\{0, 1\}^*$  de la façon suivante :

$$h : \begin{array}{ccc} \{0, 1\}^* & \longrightarrow & \mathcal{E} \\ r = x_1 x_2 \dots x_k & \mapsto & f_{x_1}(f_{x_2}(\dots f_{x_k}(e) \dots)) \end{array}$$

Montrer que la fonction  $h$  ainsi construite est à collisions fortes difficiles.



Montrer que la fonction  $h$  ainsi construite est à collisions fortes difficiles.



### Exercice 3 (Signature RSA) (5 points)

1. Calculer le module  $n$  et l'entier  $\varphi(n) = (p - 1)(q - 1)$  associés aux nombres premiers  $p = 17$  et  $q = 23$ .
2. Quels sont les exposants secrets de signature associés aux exposants publics  $e = 11$  et  $e = 13$  ?
3. Quelle est la signature de  $m = 100$  ?
4. Vérifier que la vérification fonctionne.

$$100 \rho = 17, q = 23$$

$$\varphi(n) = (17 - 1)(23 - 1) = 352$$

$$n = 17 \times 23 = 391$$

$$e // d = ?$$

$$\begin{cases} e \\ d \end{cases} = 1 \text{ mod } \varphi(n) = \begin{cases} 11 \\ d = 1 \text{ mod } 352 \end{cases}$$

$$d'' = \frac{1}{11} \text{ mod } 352 = ?$$

$$3 // m = 100, c = ?$$

# Master Informatique 1<sup>ère</sup> année

## Cryptographie - Examen

Consignes :

- Durée : 2h.
- Documents interdits.
- Aucun accès à un téléphone portable, une calculatrice, un PDA ou tout autre dispositif électronique, connectable ou non.

### Introduction

Ce devoir s'organise autour de trois parties.

Le barème de chaque partie est indiqué à titre indicatif.

Les différentes parties sont proposées par ordre croissant de difficulté.

Une attention particulière sera portée à la présentation.

La première partie comprend 10 questions de cours.

La deuxième partie traite d'un problème de cryptographie symétrique basé sur les MAC.

La troisième partie vous propose d'étudier un système de chiffrement asymétrique proposé par Merkle et Hellmann en 1978.

### 1 Question de cours (10 points)

Dans cette partie il vous est demandé de rappeler successivement :

1. Le théorème des restes chinois  
(son énoncé et sa démonstration dans le cas d'un système à deux équations);
2. Les 3 définitions permettant de caractériser la sécurité des fonctions de hachage;
3. Le théorème de Bezout;
4. L'algorithme d'Euclide;
5. Le théorème d'Euler;
6. Les modes ECB et CBC (des schémas seront particulièrement appréciés).

## 2 Le protocole de Merkle Hellman : Une histoire prometteuse... (6 points)

En 1978, Ralph Merkle et Martin Hellman proposèrent un système de cryptographie asymétrique basé sur le problème du sac à dos.

Dans sa forme décisionnelle (celle qui est la base du cryptosystème de Merkle-Hellman), le problème du sac à dos s'énonce de la manière suivante : "Connaissant  $a_1, \dots, a_m$   $m$  entiers et  $S$  un autre entier, existe-t-il un  $m$ -uplet  $x_1, \dots, x_m$  (où les  $x_i$  valent 0 ou 1) tel que

$$S = \sum_{i=1}^m x_i \cdot a_i$$

Le problème du sac-à-dos est un problème  $\text{NP}$ -complet. Le meilleur algorithme connu pour résoudre ce type de problème d'une façon générale a une complexité de l'ordre de  $\mathcal{O}(n2^{n/2})$ . Ce qui en fait un problème pratiquement insoluble dès que  $n$  est de l'ordre de 128. L'idée de Merkle-Hellman était très simple : ils sont partis du constat que certaines instances du problème du sac à dos étaient simples et notamment lorsque les coefficients  $a_i$  forment une suite super-croissante.

**Définition** Une suite  $(a_k)_{k \in \mathbb{N}} \in \mathbb{N}^\mathbb{N}$  (une suite d'entiers) est dite super-croissante lorsque

$$\forall i, a_i > \sum_{k=0}^{i-1} a_k$$

**Question 1** Dans le cas où on a un problème du sac à dos dont les coefficients forment une suite super croissante, proposez un algorithme permettant de résoudre l'instance de ce problème en temps polynomial.

En se basant sur une instance facile, ils choisirent de la complexifier en utilisant la transformation suivante :

Soient  $A, N, m \in \mathbb{N}$ ; on suppose que  $N > \sum_{i=1}^m a_i$ ,  $A < N$  et  $A \wedge N = 1$ .

On suppose de plus que les  $a_i$  forment une suite super-croissante.

Pour tout  $i$ , on définit  $b_i = Aa_i \pmod{N}$ .

La clé publique d'une personne est alors  $(b_i)_{i=1..m}$  et sa clé privée est  $(a_i)_{i=1..m}$ ,  $N$ ,  $A$ .

L'algorithme qu'ils publièrent permettait de chiffrer des blocs de  $m$  bits que l'on notera  $M = M_1 \dots M_m$  (les  $M_i$  sont des bits). Le texte chiffré,  $c$  est alors calculé ainsi :

$$c = \sum_{i=1}^m M_i b_i \pmod{N}$$

**Question 2** Montrer comment le destinataire d'un message est capable de déchiffrer.

**Question 3 : Application** Dans cette question, on suppose que la clé privée de Alice est :  $(a_1 = 2, a_2 = 3, a_3 = 6, a_4 = 13, a_5 = 25, N = 51, A = 5)$ . Vous devez calculer la clé publique, puis le chiffrement et le déchiffrement du message (11011).

Ce cryptosystème est bien plus efficace que le cryptosystème RSA, il est théoriquement plus simple et enfin, il est plus facile à implémenter. En effet, en supposant que la complexité d'une multiplication modulaire est de l'ordre de  $\mathcal{O}(\ln^2(n))$  (où  $n$  est la taille en bits du modulo), un chiffrement par l'algorithme RSA requiert  $\mathcal{O}(\ln^2(n))$  opérations (on considère que  $e$  est égale à  $2^{16} + 1$ ).

**Question 4** Donner la complexité du chiffrement et du déchiffrement pour le cryptosystème proposé par Merkle et Hellman (En considérant que l'addition compte pour 1 opération).

Partant de ces constatations, il est légitime de se demander pourquoi ce cryptosystème n'a pas été adopté aussi largement que RSA ? En fait, ce cryptosystème a été cassé en 1982 par Adi Shamir. Ceci signa l'arrêt de son utilisation.

En 1988, deux chercheurs japonais, Masakatu Morii et Masao Kasahara, proposèrent une version mutliplicative du cryptosystème de Merkle-Hellman en se basant sur le problème du sac à dos multiplicatif :

"Connaissant  $a_1, \dots, a_m$   $m$  entiers et  $P$  un autre entier, existe-t-il un  $m$ -uplet  $x_1, \dots, x_m$  (où les  $x_i$  valent 0 ou 1) tel que

$$\P P = \prod_{i=1}^m a_i^{x_i}.$$

**Question 5** Montrer que ce problème est facile lorsque les entiers  $a_i$  sont tous premiers deux à deux, i.e.,  $\forall i, j$  tels que  $i \neq j$ ,  $a_i \wedge a_j = 1$  (indication : calculer  $p_i = P \pmod{a_i}$ ).

De la même façon que Merkle et Hellman, ils cherchèrent une façon de transformer cette instance facile en une instance difficile. Pour cela ils utilisèrent une transformation analogue à la fonction utilisée dans le cryptosystème RSA :

Soit  $q \in \mathbb{P}$  ( $q$  est un nombre premier) tel que  $q > \prod_{i=1}^m a_i$  et de plus  $\forall i$ ,  $q \wedge a_i = 1$ .

Soient  $e, d \in \mathbb{Z}/q\mathbb{Z}$  tels que  $(q - 1) \wedge e = 1$  et  $d = e^{-1} \pmod{q - 1}$ .

La clé privée est alors  $(a_i)_{i=1..m}, q$  et la clé publique est  $(b_i)_{i=1..m}$  où  $\forall i$ ,  $b_i = a_i^e \pmod{q}$ .

De la même manière que pour le cryptosystème de Merkle et Hellman, ce cryptosystème sert à crypter des blocs de  $m$  bits, en gardant les mêmes notations,

$$c = \prod_{i=1}^m b_i^{M_i} \pmod{q}$$

**Question 6** Montrer comment le destinataire d'un message est capable de déchiffrer.

Cette version mutliplicative a très vite été complètement cassée ; elle n'a donc pas pu remplacer la version proposée par Merkle et Hellman. Dans la suite, on assimile la taille de clé du cryptosystème de Merkle-Hellman à la taille des  $a_i$  plus la taille des  $b_i$ .

**Question 7** Déterminer la taille des clés pour le système de Merkle-Hellman (on considère que tous les éléments de  $\mathbb{Z}/N\mathbb{Z}$  ont une taille de  $\ln(N)$  bits).

Une autre attaque a été proposée contre le système de Merkle-Hellman. Cette attaque est basée sur les réseaux euclidiens et il n'est pas question de l'aborder ici. Cependant, il est intéressant de retenir que sa complexité est de l'ordre de  $\mathcal{O}(d^5n \ln^3(n))$  (où  $n$  désigne la taille en bits des  $b_i$ ;  $d$  désigne le nombre de  $b_i$ ).

**Question 8** Déterminer la taille de clés nécessaire pour assurer une bonne sécurité si on souhaite utiliser le cryptosystème de Merkle-Hellman.

**Question 9** Conclure quand à l'utilisation pratique du cryptosystème proposé par Merkle et Hellman.

# Master – 1<sup>ère</sup> année – Informatique

## Cryptographie – Examen

14 janvier 2015

Consignes :

- Durée : 2h.
- Documents interdits.
- Aucun accès à un téléphone portable, une calculatrice, un PDA ou tout autre dispositif électronique, connectable ou non.

### Questions de cours

#### 1. DES

Expliquer pourquoi on utilise le triple DES. En particulier, quelles sont les raisons pour lesquelles on ne se contente pas du double DES. Expliquer en détail.

#### 2. Le mode de chiffrement CBC

Dans cette question, on considère un algorithme de chiffrement par blocs noté  $E$ , avec une clé notée  $K$ . On suppose que  $E_K : \{0, 1\}^n \mapsto \{0, 1\}^n$ , avec  $K \in \{0, 1\}^k$ . En d'autres termes,  $E_K(t)$  désigne le chiffré du message  $t$  de longueur  $n$ , selon la clé  $K$  de  $k$  bits.

- Faire le schéma du mode de chiffrement CBC.
- Pour un message clair de la forme  $x = x_1 || x_2 || \dots || x_\ell$ , où pour chaque  $i$ ,  $x_i \in \{0, 1\}^n$ , donner les formules qui expriment le message chiffré  $y$  en fonction du message clair.
- Expliquer comment on déchiffre le message chiffré  $y$ .
- Expliquer le rôle de la "valeur initiale" IV.

#### 3. Fonctions de hachage

- Que dit le paradoxe des anniversaires ? (donner une réponse la plus précise possible, avec si possible une formule)
- Quelle condition nécessaire donne-t-il sur la valeur de  $n$  pour qu'une fonction  $f : \{0, 1\}^* \rightarrow \{0, 1\}^n$  soit à collisions fortes difficiles ?

### Exercice 1 (*Fonction de hachage*)

1. Rappeler la définition d'une fonction :
    - à collisions fortes difficiles (= "collision-resistant")
    - à collisions faibles difficiles (= "second-preimage resistant")
    - à sens unique (= "one-way")

2. Soit  $f : \{0, 1\}^{2m} \rightarrow \{0, 1\}^n$  une fonction de hachage. Soit maintenant une deuxième fonction de hachage définie par

$$h : \begin{matrix} \{0, 1\}^{4m} \\ x_1 | x_2 \end{matrix} \longrightarrow \begin{matrix} \{0, 1\}^m \\ f(f(x_1)) | f(x_2) \end{matrix}$$

où  $\|$  désigne l'opération de concaténation. Montrer que si  $f$  est à collisions fortes difficiles, alors  $h$  est aussi à collisions fortes difficiles.

## Exercice 2 (Algorithme AES)

- Rappeler comment est défini le produit de deux octets dans l'algorithme AES. [Rappel : le polynôme  $X^8 + X^4 + X^3 + X + 1$  intervient dans la définition.]
  - Calculer  $\{D5\} \times \{E9\}$ .
  - Rappeler le théorème de Bezout et l'algorithme d'Euclide. ?.
  - Soit  $p = 3, q = 13, n = pq = 39$  et  $e = 29$ .
    - Calculer  $d$  tel que  $ed \equiv 1 \pmod{\varphi(n)}$ .
    - Chiffrer le message  $m = 2$  et vérifier le résultat en le déchiffrant.
  - On suppose qu'on intercepte le chiffré  $c = 10$ , qui a été obtenu par chiffrement RSA avec la clé publique  $n = 35$  et  $e = 5$ . Quel est le message clair ?
  - Soit  $n = pq = 851$  est un produit de deux nombres premiers. On sait que  $\varphi(n) = 792$ . Retrouver les deux facteurs premiers  $p$  et  $q$  de  $n$ .
  - Plus généralement, expliquer comment retrouver de façon générale  $p$  et  $q$  à partir de  $n (= pq)$  et  $\varphi(n)$ .

$$\begin{aligned}
 20/a \cdot \varphi(m) &= 2^{24} = 2^4 \\
 c &= 2^{24} \mod 39 \\
 &= 32 \\
 \text{so } j - c &= 32 \\
 n &= 32^d \mod 39 \\
 &= 32^5 \mod 39 \\
 &= 2 \\
 m &= 35 = \frac{2^9}{3} \\
 e(n) &= (c_p - 1)(q-1) = 24
 \end{aligned}$$

$$e \cdot d = 1 \pmod{24}$$

$$r \cdot d = 1 \pmod{24}$$

$$24 \cdot u + r \cdot d = 1$$

$$u = 1 \quad d = r$$

$$F = 10^5 \pmod{35} = 5$$

$$40 / m = p \cdot q$$

$$\varphi(n) = (p-1)(q-1)$$

$$= p \cdot q - p \cdot q + 2$$

$$= n - p \cdot q + 2$$

$$\Rightarrow p + q = n - \varphi(n) + 2 \\ = 251 - 792 + 2$$

$$\begin{cases} p+q = 61 \\ n = p \cdot q \end{cases}$$

$$\text{on } (\alpha + d)(\alpha - \beta) = \alpha^2 - \alpha(d + \beta) + d \cdot \beta$$

$$(\alpha - p)(\alpha - q) = \alpha^2 - \alpha(p + q) + p \cdot q$$

$$= \alpha^2 - \alpha(1 + n - \varphi(n)) + \underline{n}$$

$$\Rightarrow \alpha^2 - \alpha(1 + 251 - 792) + 251$$

$$\Rightarrow \alpha^2 - 60\alpha + 251$$

$$\Rightarrow \alpha = \frac{b - b \pm \sqrt{b^2 - 4ac}}{2a}$$

$$= \frac{60 \pm \sqrt{60^2 - 4 \times 251}}{2}$$

$$= \frac{60 \pm 14}{2}$$

$$= 8, 9 = 37 \quad d - q = 2, 3$$

$$511 n = p \cdot q + \varphi(n) \cdot (p-1) + 1$$

$$pq = n; p+q = 1 + \varphi(n) + 1$$

$$(\alpha + \beta)(\alpha - \beta) = \alpha^2 - \alpha(d + \beta) + d \cdot \beta$$

$$(\alpha - p)(\alpha - q) = \alpha^2 - \alpha(p + q) + p \cdot q \\ = \alpha^2 - \alpha(1 + n - \varphi(n)) + \underline{n}$$

# Cryptographie – Master 1

Examen final

Mercredi 23 Janvier 2013  
Durée : 3h

- La consultation de documents est strictement interdite, une annexe reprenant certains résultats du cours est cependant fournie à la fin du sujet;
- Les exercices marqués d'une étoile (\*) indiquent des questions plus difficiles que les autres. Si vous ne trouvez pas la réponse tout de suite, n'hésitez pas à la passer et y revenir par la suite;
- Matériel électronique : seules les calculatrices non programmables sont autorisées;
- Tout autre matériel électronique est interdit (téléphones portables y compris même ceux pouvant être utilisés comme calculatrice!);
- Le barème sur 40 n'est donné qu'à titre indicatif.

## Exercice 1 : Bref QCM (4 pts)

Voici un questionnaire à choix multiples, il vous est demandé de cocher la bonne réponse. Toute réponse fausse donne lieu à pénalité.

Numéro de code barre :

1) On travaille dans l'anneau des polynômes à coefficients dans  $\mathbb{Z}/2\mathbb{Z}$ .  
Quel est le reste de la division euclidienne de  $X^5 + X^3 + 1$  par  $X^2 + 1$ ?

- 0  
 X

$$\begin{array}{r} \cancel{X^5} \\ \cancel{X^3} \\ \hline X+1 \end{array}$$

$$L = L_{133}$$

$$\varphi(n) = (p-1)(q-1)$$

$$\varphi(40501) = 1$$

2) Sachant que  $n = 499$  est un nombre premier, que vaut  $\varphi(n)$ ?

- 498  
 500

$$\cancel{499}$$

Aucune de ces valeurs

3) Sachant que  $n = 40501 = 101 \cdot 401$  et que 101 et 401 sont deux nombres premiers, combien de nombres sont inversibles dans  $\mathbb{Z}/n\mathbb{Z}$ ?

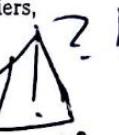
- 39999  
 40001

$$\cancel{40000}$$

Aucune de ces valeurs

$$n - p - q = 39999$$

$$=$$



$$\begin{aligned}\varphi(n) &= (2^2 - 1)(3^1 - 1)(5^1 - 1) \\ &= 2 \times 2 \times 4 \\ &= 16\end{aligned}$$

4) Sachant que  $n = 60$ , que vaut  $\varphi(n)$ ?

- 16  
 45

$$\cancel{38}$$

$$55$$

$$13$$

$$10^{12}$$

$$10^4$$

$$\begin{array}{r} 60 \\ | \quad 8 \\ | \quad 30 \\ | \quad 15 \\ | \quad 5 \\ | \quad 1 \end{array}$$

$$\begin{aligned}(2^2 - 2^1)(3^1 - 3^0)(5^1 - 5^0) &= 2 \times 2 \times 4 \\ (4 - 2)(3 - 1)(5 - 1) &= 2 \times 2 \times 4\end{aligned}$$

$$\begin{array}{cccc} x & x & x & x \\ \downarrow & \downarrow & \downarrow & \downarrow \\ 10 & 10 & 10 & 10 \end{array}$$

$$\begin{array}{c} 10 \\ \times 10 \\ \hline 100 \end{array}$$

6) Dans une salle de 120 places, combien existe-t-il de façons différentes de disposer 90 copies de sorte que chaque étudiant puisse trouver une place avec une copie et qu'aucune place ne contienne plus d'une copie ?

$$\cancel{\binom{120}{90}} = C_{120}^{90} = \frac{120!}{90!(120-90)!}$$

$120^{90}$

$\frac{120!}{90!}$

7) Lors d'un examen de Compilation, le nombre de copies rendues était un multiple de 5. Le lendemain, la même classe devrait rendre 10 copies de moins lors de l'épreuve de Cryptographie et ce nombre devrait être un multiple de 7. D'après quel théorème peut-on déterminer le nombre de copies rendues à l'épreuve de Compilation sachant qu'il est compris entre 70 et 100 ?

Théorème des restes chinois

Théorème de Gauss

Théorème d'Euler

Théorème de Bézout

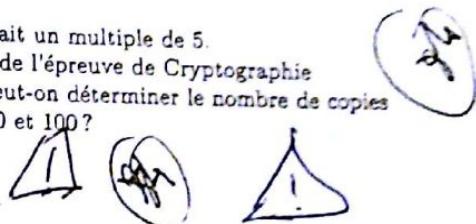
8) Combien y a-t'il de solutions à l'équation  $3x \equiv 1 [7]$  ?

0

2

1

Aucune de ces valeurs



## Exercice 2 : Fonctions de hachage (6 pts)

### Une première fonction

On considère la fonction de hachage qui renvoie systématiquement les  $n$  premiers bits de son entrée, éventuellement complétée par des 0.

Question 2.1 Cette fonction de hachage est-elle :

1. résistante à la première pré-image ?
2. résistante à la seconde pré-image ?
3. résistante aux collisions ?

Justifier.

### Une seconde fonction

On considère maintenant la fonction de hachage qui renvoie la somme (modulo 2) des bits de son entrée (i.e. un bit de parité).

Question 2.2 À partir de combien de messages hachés différents peut-on espérer trouver une collision avec une probabilité d'au moins une chance sur deux ?

Question 2.3 Cette fonction de hachage est-elle :

1. résistante à la première pré-image ?
2. résistante à la seconde pré-image ?
3. résistante aux collisions ?

Justifier.

$6 \bmod 2$

$6 \bmod 7$

$$6 \begin{array}{|l} \hline 2 \\ \hline \end{array}$$

$$6 \begin{array}{|l} \hline b \\ \hline a \\ \hline \end{array}$$

$\lambda$

$3 \cdot 2$

$3 \equiv 1 [7]$

$6 \equiv 1 [7]$

$17 = 1 \cdot 6 + 1$

### Une dernière pour la route

On considère la fonction de hachage  $H$  décrite ci-dessous.

1. On découpe un message  $m$  en  $n$  blocs de 56 bits :  $m = m_1|m_2|\dots|m_n$ ;
2. On pose  $h_0 = m_0 = 0$ ;
3. La valeur  $h_{i+1}$  est calculée de la façon suivante :

$$h_{i+1} = DES_{K(m_i)}(h_i) \oplus m_{i+1}$$

où  $K$  est une fonction qui prend en entrée 56 bits et retourne une clé  $DES$  valide (cette fonction ne fait qu'ajouter les bits de parité nécessaires pour que  $K(m_i)$  soit considéré comme une clé  $DES$  valide).

4. Enfin,  $H(m) = h_n$ .

Lorsque la taille du message  $m$  n'est pas un multiple de 56, on le complète avec des 0 jusqu'à obtenir une taille multiple de 56 bits.

**Question 2.4** Faire un schéma représentant la fonction de hachage  $H$ .

**Question 2.5** Donner la taille d'un haché (la taille de  $H(m)$ ).

**Question 2.6** Combien de hachés doit-on calculer pour avoir une probabilité de collision de l'ordre de  $1 - e^{-8}$  ?

**Question 2.7** Si on génère  $2^{60}$  messages aléatoires, et que l'on calcule ensuite leur haché par la fonction  $H$ , quelle est la probabilité d'obtenir une collision ?

### Exercice 3 : Chiffrement à flot (6 pts)

Considérons le procédé de génération d'un flux pseudo-aléatoire obtenu en effectuant le XOR du flux de sortie de deux LFSR, comme décrit ci-après :

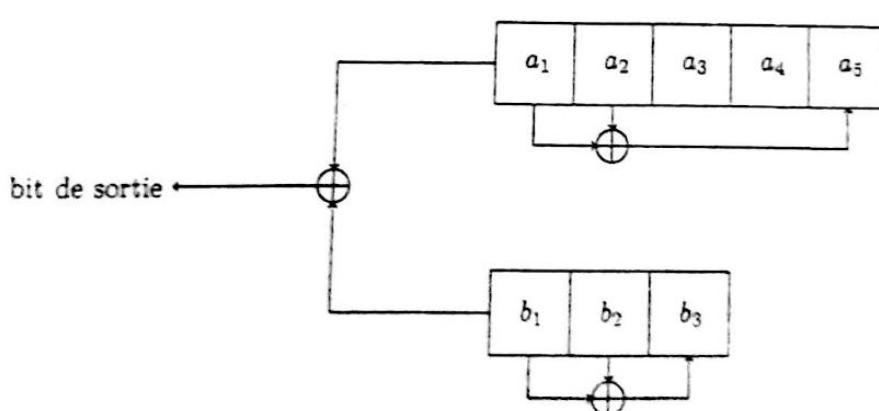


FIGURE 1 – Un générateur pseudo aléatoire basé sur deux LFSR

**Question 3.1** On suppose que les deux LFSR sont initialisés de sorte que  $(a_1, a_2, a_3, a_4, a_5) = (1, 1, 0, 1, 1)$  et  $(b_1, b_2, b_3) = (1, 0, 1)$ . Donner les états successifs des registres internes des deux LFSR lors de la génération d'un flot de 20 bits et donner le flux ainsi produit. On pourra remplir les états internes à l'aide du tableau ci-après que l'on prendra soin de rendre avec la copie :

Etat	$a_1$	$a_2$	$a_3$	$a_4$	$a_5$	$b_1$	$b_2$	$b_3$	Bit de sortie
Initial	1	1	0	1	1				
	1	0							-
	0								
	1								
	1								
	0								
	1								
	1								
	0								
	1								
	1								
	0								
	1								
	1								
	0								
	1								
	1								
	0								
	1								
	1								
	0								
	1								
	1								
	0								
	1								
	1								
	0								
	1								
	1								
	0								
	1								
	1								
	0								
	1								
	1								
	0								
	1								
	1								
	0								
	1								
	1								
	0								
	1								
	1								
	0								
	1								
	1								
	0								
	1								
	1								
	0								
	1								
	1								
	0								
	1								
	1								
	0								
	1								
	1								
	0								
	1								
	1								
	0								
	1								
	1								
	0								
	1								
	1								
	0								
	1								
	1								
	0								
	1								
	1								
	0								
	1								
	1								
	0								
	1								
	1								
	0								
	1								
	1								
	0								
	1								
	1								
	0								
	1								
	1								
	0								
	1								
	1								
	0								
	1								
	1								
	0								
	1								
	1								
	0								
	1								
	1								
	0								
	1								
	1								
	0								
	1								
	1								
	0								
	1								
	1								
	0								
	1								
	1								
	0								
	1								
	1								
	0								
	1								
	1								
	0								
	1								
	1								
	0								
	1								
	1								
	0								
	1								
	1								
	0								
	1								
	1								
	0								
	1								
	1								
	0								
	1								
	1								
	0								
	1								
	1								
	0								
	1								
	1								
	0								
	1								
	1								
	0								
	1								
	1								
	0								
	1								
	1								
	0								
	1								
	1								
	0								
	1								
	1								
	0								
	1								
	1								
	0								
	1								
	1								
	0								
	1								
	1								
	0								
	1								
	1								
	0								
	1								
	1								
	0								
	1								
	1								
	0								
	1								
	1								
	0								
	1								
	1								
	0								
	1								
	1								
	0								
	1								
	1								
	0								
	1								
	1								
	0								
	1								
	1								
	0								
	1								
	1								
	0								
	1								
	1								
	0								
	1								
	1								
	0								
	1								
	1								
	0								
	1								
	1								
	0								
	1								
	1								
	0								
	1								
	1								
	0								
	1								
	1								
	0								
	1								
	1								
	0								
	1								
	1								
	0								
	1								
	1								
	0								
	1								
	1								
	0								
	1								
	1								

# Master - 1<sup>re</sup> année - Informatique

## Cryptographie - 2<sup>nde</sup> session

15 juin 2011

Durée totale : 2h. Tous documents interdits.

### Questions de cours (5 points)

#### 1. Le mode de chiffrement CBC

Dans cette question, on considère un algorithme de chiffrement par blocs noté  $E$ , avec une clé notée  $K$ . On suppose que  $E_K : \{0, 1\}^n \mapsto \{0, 1\}^n$ , avec  $K \in \{0, 1\}^k$ . En d'autres termes,  $E_K(t)$  désigne le chiffré du message  $t$  de longueur  $n$ , selon la clé  $K$  de  $k$  bits.

- Faire le schéma du mode de chiffrement CBC.
- Pour un message clair de la forme  $x = x_1 || x_2 || \dots || x_\ell$ , où pour chaque  $i$ ,  $x_i \in \{0, 1\}^n$ , donner les formules de récurrence qui permettent d'exprimer le message chiffré  $y$  en fonction du message clair.
- Expliquer comment on déchiffre le message chiffré  $y$ .
- Expliquer le rôle de la "valeur initiale" IV.

#### 2. Fonctions de hachage

- Donner la définition d'une fonction à collisions fortes difficiles, d'une fonction à collisions faibles difficiles, d'une fonction à sens unique.
- Que dit le paradoxe des anniversaires ?
- Quelle condition nécessaire donne-t-il sur la valeur de  $n$  pour qu'une fonction  $f : \{0, 1\}^\infty \rightarrow \{0, 1\}^n$  soit à collisions fortes difficiles ?

### Exercice 1 (Questions diverses) (5 points)

- Une recherche exhaustive sur les 56 bits d'une clé DES nécessite environ 56 heures. Considérons un autre algorithme, qui utilise quant à lui une clé de 64 bits. Combien de temps faut-il pour faire une recherche exhaustive sur cette clé de 64 bits ?

56 heures       64 heures       64 jours       plus d'un an

Exercice 3 (RSA avec un modulo commun) (5 points)

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  $e_2$ .

- Montrer qu'Alice peut déchiffrer les messages adressés à Bob.
- Montrer qu'un attaquant Charlie peut déchiffrer un message envoyé à la fois à Alice et à Bob, à condition d'avoir  $\text{pgcd}(e_1, e_2) = 1$ .
- Application numérique : supposons que  $n = 35$ ,  $e_1 = 7$  et  $e_2 = 17$ . On suppose que - pour un même message clair donné  $x$  - les messages chiffrés envoyés respectivement à Alice et à Bob sont  $y_1 = 13$  et  $y_2 = 23$ . Trouver le message clair  $x$ .

1°/

$$\begin{array}{c} \text{Alice} \\ n, e^{(n)} \\ (e_1, d_1) \end{array} \xrightarrow{\hspace{1cm}} c_1 = m_1^{e_1} \pmod{n}$$

$$\begin{array}{c} \text{Bob} \\ n, e^{(n)} \\ (e_2, d_2) \end{array} \quad \begin{array}{c} \text{Alice} \\ n, e^{(n)} \\ (e_1, d_1) \end{array} \xrightarrow{\hspace{1cm}} c_2 = m_2^{e_2} \pmod{n}$$

2°/

$$\text{pgcd}(e_1, e_2) = 1 \Rightarrow e_1 \text{ et } e_2 \text{ sont premiers entre eux}$$

$$e_1 + e_2 = 1$$

$$c_1 = m^{e_1} \pmod{n}$$

$$c_2 = m^{e_2} \pmod{n}$$

$$\text{is n-pot de m-pot}$$

$$c_1^u \cdot c_2^v = m^{ue_1 + ve_2} \pmod{n}$$

$$= m \pmod{n}$$

$$c_1^u \cdot c_2^v = m \pmod{n}$$

$$\begin{aligned} d &= -8 \pmod{35} \\ &= 33 \pmod{35} \end{aligned}$$

$$d : 33$$

$$3^{\circ}/ e_1 = 7 \quad d \cdot e_2 = 17$$

$$\text{on a } d \cdot d \equiv 1$$

$$7 \cdot u + 12 \cdot d = 1$$

$$\Rightarrow \boxed{d = 23}$$

$$\Rightarrow \begin{array}{c} 13^7 \cdot 23^2 \pmod{35} \\ 13^7 \cdot 23^2 = m \pmod{35} \end{array}$$

$$\begin{aligned} m &= 13^7 \cdot 23^2 \pmod{35} \\ &= 13 \cdot 2 \pmod{35} \end{aligned}$$

$$= 34$$

18

$$N^{P_i, C_i} \text{ mod } n$$

$$(c_1, d)$$

$$c_1 = n^{e_1} \text{ mod } n$$

$$\begin{array}{l} \text{Bob's} \\ N^{P_2, C_2} \text{ mod } n \\ (c_2, d) \end{array}$$

$$\begin{array}{l} c_2 = n^{e_2} \text{ mod } n \\ c_2 = n^d \text{ mod } n \end{array}$$

so

$$p \text{ and } (c_1, c_2) = d \Rightarrow c_1, c_2 \text{ are not multiples of } p$$

$$c_1 \cdot c_2 \equiv 1$$

$$c_1 = n^{e_1} \text{ mod } n$$

$$c_2 = n^{e_2} \text{ mod } n$$

$$\begin{aligned} & \text{if } n \text{ is prime, } c_1 \cdot c_2 \equiv n^{e_1+e_2} \text{ mod } n \\ & c_1 \cdot c_2 \equiv n^{e_1+e_2} \text{ mod } n \end{aligned}$$

$$\begin{aligned} & \equiv n^{d+e_2} \text{ mod } n \\ & \equiv 33 \text{ mod } 35 \end{aligned}$$

(d: 33)

$$c_1 \cdot c_2 = n \text{ mod } n$$

$$3^0 / e_1 = 7 \cdot d \cdot c_2 = 17 \Rightarrow \cancel{13^5 \cdot 25} \text{ mod } 35$$

$$\text{on } u \cdot d \cdot d^{-1} q$$

$$2 \cdot u + 17 \cdot d = 1$$

$$\Rightarrow \cancel{13^5 \cdot 25}$$

$$\begin{aligned} & 13^5 \cdot 23 = u \text{ mod } 35 \\ & 13^5 \cdot 23 \text{ mod } 35 \\ & 13 \cdot 23 \text{ mod } 35 \\ & 13 \cdot 2 \text{ mod } 35 \end{aligned}$$

Exercice 3 (RSA avec un module commun) (5 points)

On suppose que deux entités Alice et Bob utilisent un schéma de chiffrement RSA avec le même module  $n$  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, à condition d'avoir  $\text{pgcd}(e_1, e_2) = 1$ .
3. Application numérique : supposons que  $n = 35$ ,  $e_1 = 7$  et  $e_2 = 17$ . On suppose que pour un même message clair donné  $x$  – les messages chiffrés envoyés respectivement à Alice et à Bob sont  $y_1 = 32$  et  $y_2 = 23$ . Trouver le message clair  $x$ .

# Interrogation

Cryptographie – Master 1

Mercredi 21 Novembre 2012  
Durée : 1h30

- La consultation de documents est strictement interdite ;
- Matériel électronique : seules les calculatrices non programmables sont autorisées ;
- Tout autre matériel électronique est interdit (téléphones portables y compris même ceux pouvant être utilisés comme calculatrice !) ;
- Le barème n'est donné qu'à titre indicatif.

*Les questions suivies d'une étoile (\*) sont des questions ouvertes qui ne se rapportent pas directement à votre cours ou à vos TD. Ne passez pas trop de temps dessus si vous ne voyez pas quitte à y revenir par la suite. D'autre part, le sujet est estimé comme long et le barème sera réévalué en conséquence. Ne cherchez pas à tout faire rapidement. Prenez le temps nécessaire pour répondre correctement à chaque question.*

## 1 Quelques questions de cours (2,5 pts)

**Question 1 :** Sous quel nom est connu l'algorithme Lucifer ?

**Question 2 :** Rappelez ce que signifie l'acronyme DES.

**Question 3 :** Quel est le nom du successeur du DES ?

**Question 4 (\*) :** A quelles familles de chiffrement (chiffrement symétrique/asymétrique, chiffrement par bloc/à flot) appartient RC4 ?

**Question 5 :** Rappelez en quelques lignes le fonctionnement du mode de chiffrement OFB (vous pourrez illustrer votre propos par un schéma).

## 2 Le protocole WPS (total : 10,5 pts)

Suite aux récentes lois pour la diffusion des œuvres et la protection des droits sur internet, tout acte de téléchargement illégal de contenu est considéré comme imputable au propriétaire de la connexion ayant servi au méfait. Si le méfait n'a pas été commis par le propriétaire de la connexion internet, ce dernier peut cependant être poursuivi pour "défaut de sécurisation" de son réseau. Ainsi, de plus en plus de personnes sécurisent leur réseau Wi-Fi avec des protocoles plus robustes que le protocole WEP (le protocole WPA-PSK par exemple). Une protection complémentaire consiste à employer des mots de passe longs, variés et complexes. Partant du principe qu'il est fastidieux de saisir ces clés et qu'il est préférable d'employer des mots de passe dénués de sens (pour éviter d'être sensible aux attaques par dictionnaire), un nouveau protocole a été conçu : le Wi-Fi Protected Setup (WPS).

Lors de la connexion d'un appareil à un routeur, le protocole WPS permet à l'appareil de s'authentifier par la saisie d'un mot de passe à 8 chiffres appelé code PIN. Lorsque le code est correct, le routeur doit envoyer à l'appareil la clé du réseau ; le cas échéant, un message d'erreur est renvoyé. Pour éviter qu'un intervenant extérieur ne puisse récupérer ni le code PIN saisi, ni la clé du réseau, le protocole WPS chiffre ses communications suite à un échange de clés de type Diffie-Hellman (ce protocole sera évoqué en cours ultérieurement).

Dans cet exercice, nous considérerons que le réseau est chiffré au moyen du protocole WPA-PSK et étudions le protocole WPS pour mettre en avant une grave vulnérabilité publiée le 26 décembre 2011 par Stefan Viehböck.

## Partie 1 – Attaque par recherche exhaustive (5,5 pts)

Dans cette partie on se place dans le cas d'une recherche du code PIN par recherche exhaustive. Pour cela, nous allons étudier les 3 contextes suivants :

### Contexte 1 : Attaque sur 8 chiffres

On se place dans le cas où l'attaquant décide d'essayer toutes les combinaisons de 8 chiffres possibles pour trouver le code PIN associé au réseau WPA-PSK.

### Contexte 2 : Attaque sur 8 chiffres avec connaissance d'une relation les liant

On se place cette fois dans le cas où l'attaquant sait que le code PIN du protocole WPS est en fait composé de 7 chiffres suivi d'un huitième calculé en fonction des autres (au moyen de la fonction HMAC qui sera détaillée dans la partie 3).

### Contexte 3 : Attaque en plusieurs temps

Finalement, on se place dans le même contexte que le contexte 2 sauf que l'attaquant sait que la procédure d'authentification du protocole WPS a lieu comme suit :

- L'appareil envoie les 4 premiers chiffres du code PIN au routeur ;
- Le routeur répond à l'appareil si les 4 chiffres saisis sont corrects ou non ;
- Si tel est le cas, l'appareil envoie alors au routeur les 4 derniers chiffres du code PIN ;
- Le routeur répond à l'appareil si les 4 chiffres saisis sont corrects ou non. Lorsqu'ils le sont, il retourne en plus la clé du réseau WPA-PSK.

### Questions

Pour chacun de ces contextes, il vous est demandé de répondre aux questions suivantes :

- a) Combien de codes PIN doivent être testés dans le pire des cas ?
- b) Combien de codes PIN doivent être testés en moyenne ?
- c) Combien de temps prendra ce type d'attaque dans le pire des cas si l'on ne peut tester qu'un code PIN par seconde.
- d) Combien de temps prendra ce type d'attaque dans le pire des cas si l'on ne peut tester qu'un code PIN toute les 30 secondes.
- e) Combien de temps prendra ce type d'attaque dans le pire des cas si l'on ne peut tester qu'un code PIN par minute.

Note : Il n'est pas demandé de donner le nombre de codes PIN à tester sous forme développée (ex :  $2^5 \cdot 3^2 \cdot 7$  est une réponse sous une forme acceptable). Les temps obtenus devront être exprimés en secondes, puis au format jours-heures-minutes-secondes (exemple : 1j 3h 52' 43"). Pour répondre à ces questions, vous recopierez et compléterez le tableau suivant :

	Contexte 1	Contexte 2	Contexte 3
Nombre d'essais dans le pire des cas			
Nombre d'essais en moyenne			
Temps de l'attaque pour 1 essai par seconde			
Temps de l'attaque pour 1 essai par 30s			
Temps de l'attaque pour 1 essai par minute			

## Partie 2 – Analyse de la fonction HMAC (5 pts)

Pour qu'un code PIN  $x = (x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8)$  soit valide pour le protocole WPS, il faut et il suffit que  $x_8$  soit égal à  $\text{HMAC}(x) = (-(3.x_1 + x_2 + 3.x_3 + x_4 + 3.x_5 + x_6 + 3.x_7) \bmod 10)$ .

**Question 1 :** Donner (dans un tableau) les images des codes PIN suivants : 0023456, 0123456, 0223456, 0323456, 0423456, 0523456, 0623456, 0723456, 0823456 et 0923456. Que remarquez-vous ?

### Question 2 (\*) :

- a) Démontrer que pour des valeurs de  $x_1, x_3, x_4, x_5, x_6$  et  $x_7$  fixées, le calcul des images par la fonction HMAC de  $x$  lorsque  $x_2$  prend successivement pour valeur les nombres de 0 à 9 ne donnera qu'une et une seule fois chaque entier de 0 à 9.

Note : On pourra remarquer que pour un certain code PIN  $y$  que l'on précisera, on a :

$$\text{HMAC}(x) = \text{HMAC}(y) - x_2 \bmod 10.$$

- b) En déduire le nombre de codes PIN valides pour lesquels  $x_8$  vaut 1.
- c) En déduire un critère statistique intéressant en cryptographie sur la fonction HMAC.

# Master - 1<sup>re</sup> année - Informatique

## Cryptographie - Examen (niveau agrégé)

15 juin 2005

Durée totale : 2h. Tous documents interdits.

### Questions de cours (5 points)

#### 1.- Le mode de chiffrement CBC

Dans cette question, on considère un algorithme de chiffrement par blocs noté  $E$ , avec une clé notée  $K$ . On suppose que  $E_K : \{0, 1\}^n \rightarrow \{0, 1\}^n$ , avec  $K \in \{0, 1\}^k$ . En d'autres termes,  $E_K(t)$  désigne le chiffré du message  $t$  de longueur  $n$ , selon la clé  $K$  de  $k$  bits.

- Faire le schéma du mode de chiffrement CBC.
- Pour un message clair de la forme  $x = x_1 || x_2 || \dots || x_\ell$ , où pour chaque  $i$ ,  $x_i \in \{0, 1\}^n$ , donner les formules de récurrence qui permettent d'exprimer le message chiffré  $y$  en fonction du message clair.
- Expliquer comment on déchiffre le message chiffré  $y$ .
- Expliquer le rôle de la "valeur initiale" IV.

#### 2. Fonctions de hachage

- Donner la définition d'une fonction à collisions fortes difficiles, d'une fonction à collisions faibles difficiles, d'une fonction à sens unique.
- Que dit le paradoxe des anniversaires ?
- Quelle condition nécessaire donne-t-il sur la valeur de  $n$  pour qu'une fonction  $f : \{0, 1\}^\infty \rightarrow \{0, 1\}^n$  soit à collisions fortes difficiles ?

### Exercice 1 (Questions diverses) (5 points)

1. Une recherche exhaustive sur les 56 bits d'une clé DES nécessite environ 56 heures. Considérons un autre algorithme, qui utilise quant à lui une clé de 64 bits. Combien de temps faut-il pour faire une recherche exhaustive sur cette clé de 64 bits ?

56 heures       64 heures       64 jours       plus d'un an

— — — — — attaques passives

□ est zero-knowledge<sup>1</sup>

**Exercice 2** (Chiffrement/déchiffrement RSA) (5 points)

Dans toute la suite, on pourra utiliser les résultats numériques suivants :

- $319 = 11 \times 29$  ;  $10^{11} = 263 \pmod{319}$  ;  $263^2 = 216 \times 319 + 265$  ;  
 $\uparrow \uparrow \uparrow = 5 \times 2 + 1$
- $133^3 = 12 \pmod{319}$  ;  $133^{25} = 133 \pmod{319}$  ;
- $11^2 = 121 \pmod{280}$  ;  $11^4 = 81 \pmod{280}$  ;  $11^8 = 121 \pmod{280}$  ;  $11^{16} = 81 \pmod{280}$  ;
- $95 = 64 + 31$  ;  $81 \times 11 = 51 \pmod{280}$  ;  $81 \times 121 = 1 \pmod{280}$ .

On considère la clé publique RSA  $(11, 319)$ , c'est-à-dire pour  $n = 319$  et  $e = 11$ .

1. Quel est le chiffrement avec cette clé du message  $M = 100$  ?
2. Calculer  $d$  la clé privée correspondant à la clé publique  $e$ .
3. Déchiffrer le message  $C = 133$ .

<sup>1</sup>On rappelle qu'un algorithme d'authentification est dit "zero-knowledge" si le fait de l'utiliser (pour s'authentifier) ne révèle rien sur ses propres secrets.

# Cryptographie – Master 1

Examen de rattrapage

Lundi 17 Mars 2013  
Durée : 2h

- La consultation de documents est strictement interdite;
- Certaines questions peuvent être plus difficiles que les autres. Si vous ne trouvez pas la réponse tout de suite, n'hésitez pas à passer à la question suivante et y revenir éventuellement par la suite;
- Matériel électronique : seules les calculatrices non programmables sont autorisées;
- Tout autre matériel électronique est interdit (téléphones portables y compris même ceux pouvant être utilisés comme calculatrice!);
- Le barème (sur 20) n'est donné qu'à titre indicatif.

## Exercice 1 : Code EAN13 (5 pts)

Les codes-barres apposés sur de nombreux articles du commerce représentent une suite de 13 chiffres appelé *code EAN13*. Parmis ces 13 chiffres se trouvent 12 chiffres nécessaires à l'identification du produit et un 13<sup>e</sup> chiffre calculé à partir des précédents. Ce dernier chiffre appelé *clé de contrôle* et vise à détecter les mauvaises lectures du code-barre (code-barre abimé, lecteur optique défaillant, ...). Ces types de codes-barres est généralement représenté comme suit :



Considérons les 13 chiffres d'un code-barre :  $a_1 b_1 a_2 b_2 a_3 b_3 a_4 b_4 a_5 b_5 a_6 b_6 c$ .

On peut alors calculer la clé de contrôle  $c$  au moyen de la formule  $c = -(a + 3b) \bmod 10$  avec  $a = a_1 + a_2 + a_3 + a_4 + a_5 + a_6 \bmod 10$  et  $b = b_1 + b_2 + b_3 + b_4 + b_5 + b_6 \bmod 10$ .

**Question 1.1** Combien y a-t'il de codes-barres (valides) possibles ?

**Question 1.2** Voici les 12 premiers chiffres de trois codes-barre différents. Donner la clé de contrôle de chacun d'entre eux.

- 0000 0000 0000
- 2222 2222 2222
- 0123 4567 8901

**Question 1.3** Voici quelques codes-barres dont un chiffre manque (représenté par la lettre  $X$ ). Dire pour chacun d'eux quelles sont la ou les valeurs possibles de  $X$  pour que le code soit valide :

- $X111\ 2222\ 3333\ 4$
- $1X11\ 2222\ 3333\ 4$

**Question 1.4** Est-il toujours possible de retrouver un chiffre manquant ? Justifier votre réponse en le démontrant ou en fournissant un contre-exemple.

**Question 1.5** On se place dans le cas général. Dire pour chacun des cas suivants si la clé de contrôle d'un code-barre reste toujours valide. Vous justifierez votre réponse en fournissant une démonstration ou un contre-exemple.

- Cas où le premier et le troisième chiffre ont été inversés ;
- Cas où l'un des douze premiers chiffres du code (seulement un) est modifié ;
- Cas où deux des douze premiers chiffres du code sont modifiés.

## Exercice 2 : RDS-TMC (5 pts)

Traffic Message Channel, ou TMC, est une norme européenne qui permet de diffuser des informations de circulation aux automobilistes, généralement via le système RDS de la radio FM<sup>1</sup>. Les messages TMC sont essentiellement composés de valeurs numériques représentant les localisations et les messages d'avertissement dans une base de données publique. Dans certains pays, ces informations sont chiffrées à l'aide d'un schéma de **chiffrement symétrique** que l'on notera  $E$  dans la suite ; celui-ci manipule des messages et des chiffrés de 2 octets en utilisant des clefs de 2 octets. Les clefs sont stockées dans les récepteurs GPS ; chaque jour seul le numéro de la clef à utiliser leur est transmise.

**Question 2.1** On note  $M$  le nombre de messages clairs possibles et  $K$  le nombre de clefs. Combien valent  $M$  et  $K$  ?

**Question 2.2** Quelle est la taille (espace mémoire) nécessaire pour stocker les chiffrés de tous les messages  $m$  avec toutes les clefs  $k$ , dans un tableau défini selon le modèle suivant ?

	$k_1$	$k_2$	$\dots$	$k_K$
$m_1$	$E(m_1, k_1)$	$E(m_1, k_2)$	$\dots$	$E(m_1, k_K)$
$m_2$	$E(m_2, k_1)$	$E(m_2, k_2)$	$\dots$	$E(m_2, k_K)$
$\vdots$	$\vdots$	$\vdots$	$\dots$	$\vdots$
$m_M$	$E(m_M, k_1)$	$E(m_M, k_2)$	$\dots$	$E(m_M, k_K)$

En pratique, certaines des informations transmises par TMC peuvent facilement être connues (travaux de longue durée, par exemple).

**Question 2.3** Expliquer comment cette table peut être utilisée pour retrouver la clef lors d'une attaque à clair connu.

Le fonctionnement du chiffrement  $E$  est illustré sur la figure 1. Nous détaillons à présent l'exemple donné dans cette figure. Les valeurs indiquées par  $d$  seront des valeurs exprimées en décimal et celles indiquées par  $h$  seront exprimées sous forme hexadécimale. Lorsque l'on aura besoin d'utiliser de réaliser des opérations avec des nombres binaires, il faudra bien entendu convertir les nombres en binaire si ils n'y sont pas, en gardant en tête que le bit de poids le plus fort sera celui le plus à gauche. Descriptif de la figure :

<sup>1</sup> Source : [http://fr.wikipedia.org/wiki/Traffic\\_Message\\_Channel](http://fr.wikipedia.org/wiki/Traffic_Message_Channel)

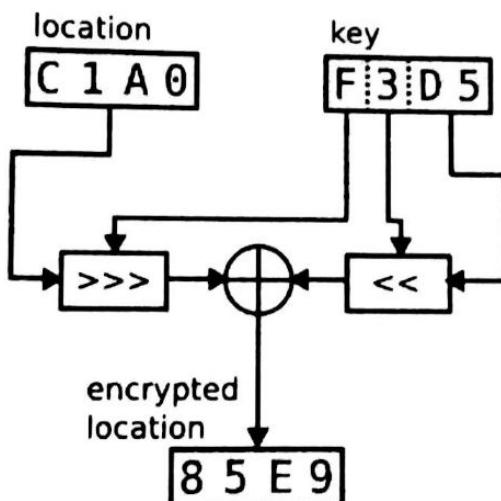
- La clef  $k$  de 16 bits est séparée en 3 paramètres distincts :

$$k = (\alpha \parallel \beta \parallel \gamma)$$

où  $\alpha$  et  $\beta$  sont chacun formés de 4 bits,  $\gamma$  est formé de l'octet restant et où  $\parallel$  désigne l'opérateur de concaténation. Dans l'exemple de la figure 1 où  $k = F3D5_h$ , on a  $\alpha = F_h = 15_d$ ,  $\beta = 3_d$  et  $\gamma = D5_h$  ;

- $\gamma$  est complété par des 0 à gauche pour former une variable sur deux octets puis est décalé de  $\beta$  bits vers la gauche ; on note cette opération  $\delta = \gamma << \beta$ . Dans notre exemple,  $\delta = D5_h << 3 = 06A8_h$  ;
- Une permutation circulaire<sup>2</sup> de  $\alpha$  bits vers la droite est appliquée sur les 2 octets du message  $m$ , on note cette opération  $m >>> \alpha$ . Dans l'exemple de la figure 1 où  $m = F3D5_h$ , on a  $m >>> 15 = 8341_h$  ;
- Le chiffrement de  $m$  à l'aide de la clef  $k$  est le résultat du ou exclusif entre le résultat des deux permutations :

$$\begin{aligned} E(m, k) &= (m >>> \alpha) \oplus \delta \\ \text{avec } \delta &= (\gamma << \beta) \end{aligned}$$



Source : <http://windytan.blogspot.fi/2013/05/a-determined-hacker-deciphers-rds-tmc.html>

FIGURE 1 – Chiffrement des données TMC

**Question 2.4** Reproduire le dessin et placer  $\alpha$ ,  $\beta$ ,  $\gamma$ ,  $\delta$ ,  $m$  et  $E(m, k)$  sur les flèches.

**Question 2.5** Expliquer, éventuellement à l'aide d'un dessin, comment sont déchiffrées les données à partir de la clef.

**Question 2.6**

- Calculer le chiffré de  $1234_h$  avec la clef  $5678_h$ .
- Soit  $c = 9BCD_h$  le chiffré d'un message  $m$  avec la clef  $DEF1_h$ . Retrouver  $m$ .

**Question 2.7** Quel est le chiffré d'un message  $m$  à l'aide de la clef  $k = 0$  ?

2. À la différence d'un décalage, dans une permutation circulaire, les bits < sortants > de la variable ne sont pas perdus mais réintroduits à l'autre bout. Par exemple, sur 8 bits, une permutation circulaire de 1 bit vers la gauche de 11110001 donne 11100011

**Question 2.8** Chiffrer le message  $m = FFFF_h$  avec les clefs suivantes :

- $k_1 = 2F0F_h$  ;
- $k_2 = 2F07_h$  ;
- $k_3 = 29FF_h$  ;

**Question 2.9** Dans le cas où  $m = FFFF_h$  et où  $c = 5FFF_h$ . Que pouvez-vous dire :

- de la valeur de  $\alpha$  ?
- de la valeur de  $\beta$  ?
- de  $\gamma$  ?

Pouvez-vous en déduire la clé ?

**Question 2.10** Chiffrer le message  $m = 0000_h$  et déchiffrer le message  $c = FFFF_h$  avec les clefs suivantes :

- $k_1 = A5FF_h$  ;
- $k_2 = 28FF_h$  ;
- $k_3 = E0FF_h$  ;
- $k_4 = A9FF_h$  ;
- $k_5 = ADFF_h$ .

**Question 2.11** À l'aide de la question précédente, expliquer comment retrouver  $\beta$  et  $\gamma$  par une attaque à clairs choisis lorsque  $\beta < 8$ .

**Question 2.12** Même question à l'aide d'une attaque à chiffrés choisis.

**Question 2.13** Expliquer comment retrouver  $k$  lors d'une attaque à clairs ou chiffrés choisis dans le cas où  $\beta$  et  $\gamma$  sont connus. Quel sera le coût d'une telle opération (en nombre de chiffrements/déchiffrements) ?

**Question 2.14** Que pensez-vous de la sécurité du chiffrement de RDS-TMS ?

### Exercice 3 : RSA (5 pts)

Dans la suite, on désigne par module RSA un entier  $n$  produit de deux entiers premiers différents notés respectivement  $p$  et  $q$ .

On vous donne accès à trois fonctions d'une API :

- `inv_mod(x, n)` qui renvoie  $a = \begin{cases} x^{-1} \bmod n \text{ avec } a \in [0, n - 1] \text{ lorsque } x \text{ est inversible dans } (\mathbb{Z}/n\mathbb{Z}) \\ -1 \text{ sinon} \end{cases}$  ;
- `powmod(e, x, n)` qui renvoie  $x^e \bmod n$  ;
- `prime_number(t)` qui renvoie un entier premier sur  $t$  bits.

En utilisant les trois fonctions précédentes, vous devez répondre aux questions ci-dessous.

**Question 3.1** Écrire un algorithme en pseudo-code qui permet de générer une paire de clés RSA. Votre algorithme prendra en entrée un nombre entier  $t$  et renverra une paire de clés RSA dont le module est sur  $t$  bits.

**Question 3.2** Écrire un algorithme en pseudo-code qui permet de chiffrer un message.

**Question 3.3** Écrire un algorithme en pseudo-code qui permet de déchiffrer un message.

**Question 3.4** Dans le cryptosystème RSA, quel est l'ensemble contenant les messages clairs ?

**Question 3.5** Dans le cryptosystème RSA, quel est l'ensemble contenant les messages chiffrés ?

**Question 3.6** On vous donne la clé publique RSA suivante : ( $e = 3, n = 55$ ).

1. Calculer le chiffrement du message  $m = 10$ .
2. Calculer la clé privée correspondant à la clé publique donnée ci-dessus.
3. Déchiffrer le message  $c = 1$ .

**Question 3.7**

1. Si on utilise le cryptosystème RSA tel qu'il a été décrit en cours, que se passe-t-il lorsque l'on envoie deux fois le même message ?
2. Que pensez-vous de l'utilisation telle quelle de ce cryptosystème dans la pratique ?

## Exercice 4 : Diffie-Hellman (5 pts)

**Définition :** Un groupe est un couple formé d'un ensemble  $G$  et d'une loi de composition interne  $\star$  sur cet ensemble telle que :

1.  $\forall g_1, g_2 \in G, g_1 \star g_2 \in G$  ;
2.  $\star$  est une loi associative, i.e.,  $\forall g_1, g_2, g_3 \in G, (g_1 \star g_2) \star g_3 = g_1 \star (g_2 \star g_3)$  ;
3. Il existe un élément neutre  $e$  pour la loi  $\star$ , i.e.,  $\exists e \in G, \forall g \in G, g \star e = e \star g = e$  ;
4. Tout les éléments sont inversibles :  $\forall g \in G, \exists h \in G : g \star h = h \star g = e$ .

**Question 4.1** À l'aide de la définition précédente, vérifier que le couple  $((\mathbb{Z}/n\mathbb{Z}), +)$ , où  $n$  est un entier strictement positif, est un groupe.

On notera plus simplement la composition d'un élément avec lui-même de la façon suivante :

**Cas d'un groupe en notation multiplicative**  $\underbrace{g \times g \times \cdots \times g}_{i \text{ fois}} = g^i$  ;

**Cas d'un groupe en notation additive**  $\underbrace{g + g + \cdots + g}_{i \text{ fois}} = i \cdot g$ .

**Définition :** On appelle générateur d'un groupe  $G$  un élément  $g$  de ce groupe tel que (en notation multiplicative) :

$$\forall h \in G, \exists x \in \mathbb{Z} : h = g^x.$$

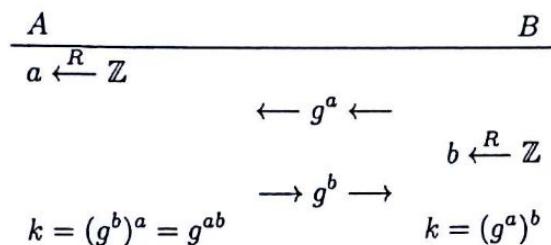
**Question 4.2** Vérifier que 1 est un générateur du groupe  $((\mathbb{Z}/n\mathbb{Z}), +)$ .

**Le protocole de Diffie et Hellman** Ce protocole permet à deux personnes de s'accorder publiquement sur une clef commune qui pourra, par exemple, servir à chiffrer les communications entre eux à l'aide d'un chiffrement symétrique.

**Notations :** Dans la suite on notera  $G$  un groupe et  $g$  un générateur de ce groupe. D'autre part, la notation  $x \xleftarrow{R} E$  signifie que  $x$  est une valeur tirée aléatoirement dans l'ensemble  $E$ .

En utilisant une notation multiplicative (donc pour une loi de composition interne a priori non commutative), le protocole de Diffie et Hellman entre  $A$  et  $B$  est décrit ci-dessous.

Alice (resp. Bob) tire aléatoirement un nombre *secret*  $a$  (resp.  $b$ ), élève  $g$  à la puissance  $a$  (resp.  $b$ ) dans  $G$  et transmet le nombre  $A = g^a$  (resp.  $B = g^b$ ) à Bob (resp. Alice) par un canal public. À la fin de cet échange, Alice peut éléver le nombre  $B$  qu'il a reçu de Bob à la puissance  $a$  pour obtenir  $K = B^a = (g^b)^a = g^{ab}$ . De même, Bob peut calculer  $A^b = (g^a)^b = g^{ab} = K$ . On donne ci-dessous une représentation schématique de ce protocole :



**Question 4.3** En vous inspirant de la description schématique précédente qui utilise une notation multiplicative, donner une description schématique du protocole de Diffie et Hellman en utilisant une notation additive.

**Le problème du logarithme discret :** Soit  $G$  un groupe et  $g$  un générateur de ce groupe. Soit  $h$  un élément de ce groupe. Le logarithme discret en base  $g$  de  $h$ , noté  $\ln_g(h)$ , est l'entier  $x$  tel que  $h = g^x$ .

**Question 4.4** Reformuler le problème du logarithme discret en notation additive.

**Question 4.5** Montrer que si on sait facilement calculer les logarithmes discrets dans un groupe  $G$ , alors le protocole de Diffie et Hellman n'est pas sécurisé dans ce groupe.

*Remarque :* Dans votre preuve vous prendrez soin d'utiliser une notation additive.

**Question 4.6** Donner le logarithme discret en base  $g$  de  $h = 5$  dans les deux situations suivantes :

1.  $((\mathbb{Z}/2006303739164321853044305842577017\mathbb{Z}), +)$  et  $g = 1$  ;
2.  $((\mathbb{Z}/14\mathbb{Z}), +)$  et  $g = 3$ .

*Remarque :* Avec votre réponse, vous prendrez soin de bien détailler les calculs qui vous ont permis d'arriver au résultat.

**Question 4.7** À la lumière des questions précédentes, que pensez-vous de l'utilisation d'un groupe de la forme  $((\mathbb{Z}/n\mathbb{Z}), +)$  pour le protocole de Diffie et Hellman ?

# Master – 1<sup>ère</sup> année – Informatique

## Cryptographie – Examen

15 janvier 2014

C  
C

Consignes :

- Durée : 2h.
- Documents interdits.
- Aucun accès à un téléphone portable, une calculatrice, un PDA ou tout autre dispositif électronique, connectable ou non.

### Exercice 1 (8 points) [Questions diverses]

1. Une recherche exhaustive sur les 56 bits d'une clé DES nécessite environ 56 heures. Combien de temps faudrait-il approximativement sur une clé de 64 bits ?

- 56 heures       64 heures       64 jours       plus d'un an

2. Alice a utilisé le chiffrement "one-time pad" pour envoyer un message  $m \in \{0, 1\}^{100}$  à Bob. Ils partageaient tous deux une clé aléatoire  $k \in \{0, 1\}^{100}$ . Charlie intercepte le chiffré  $c = m \oplus k$ . Quel est le temps nécessaire pour retrouver  $m$  ?

- instantané       100 essais        $2^{100}$  essais       impossible

3. Une implémentation de RSA, utilisant des exposants public et privé aléatoires de la taille du module, annonce le temps de calcul suivant : 1 milliseconde pour chiffrer un message de 512 bits avec une clé de 512 bits. Sachant que cette implémentation utilise l'algorithme d'exponentiation vu en cours, quel temps nécessiterait le chiffrement RSA d'un message de 2048 bits avec une clé de 2048 bits (en millisecondes) ?

- 1       8       16       32       64       128

4. Afin de pouvoir distinguer ses communications personnelles de ses communications professionnelles, Alice utilise deux clés publiques RSA, ses correspondants utilisant l'une ou l'autre selon le type de communication. Afin d'accélérer la génération de clés, Alice ne choisit que trois grands nombres premiers  $p$ ,  $q$  et  $r$  de 512 bits, qu'elle garde secrets. Ses deux modules RSA publics sont alors  $N_1 = pq$  et  $N_2 = qr$ . Alice choisit aléatoirement deux couples d'exposants privés et publics  $(d_1, e_1)$  et  $(d_2, e_2)$  vérifiant donc  $e_1 d_1 \equiv 1 \pmod{(p-1)(q-1)}$  et  $e_2 d_2 \equiv 1 \pmod{(q-1)(r-1)}$ . Quelle est la sécurité obtenue ?

- impossible à déterminer       identique au RSA traditionnel       aucune sécurité

Q  
60  
120

5. Une technique classique d'identification est le "mot de passe". Si on a confiance dans le serveur, cette technique
- est sûre une fois       résiste aux attaques passives       est zero-knowledge<sup>1</sup>
6. Un problème du mode CBC est qu'aucun parallélisme n'est possible. Il faut connaître le chiffré du bloc précédent. Une proposition est de séparer l'ensemble des blocs en par exemple deux groupes : ceux de numéro pair et ceux de numéro impair. On fait donc deux CBC en parallèle, l'un avec la clé  $k_1$ , et l'initialisation  $IV_1$ , l'autre avec  $k_2$  et  $IV_2$ . Comment garder la sécurité de CBC ?
- Je peux utiliser la même clé et le même IV  
 Je peux utiliser la même clé mais pas le même IV  
 Les clés doivent être différentes et les IV aussi  
 Je ne peux pas avoir la sécurité d'un unique CBC
7. Pour la signature de longs messages, on utilise habituellement une fonction de hachage destinée à transformer le message avant signature. Sachant qu'une borne supérieure pour une recherche exhaustive se situe vers  $2^{80}$ , quelle longueur de haché doit-on préconiser pour éviter les contrefaçons ?
- 40 bits       80 bits       128 bits       160 bits
8. Alice connaît la factorisation de  $n = pq$ , et elle seule. Pour prouver son identité, elle accepte, de qui le souhaite, un nombre  $y = x^2 \pmod{n}$ . Puis elle retourne une racine carrée de  $y$ , ce qu'elle seule peut calculer. Ce système d'authentification résiste
- à rien       aux attaques passives       est zero-knowledge

### Exercice 2 (Construction d'une fonction de hachage) (4 points)

- Rappeler la définition d'une fonction à collisions fortes difficiles, d'une fonction à collisions faibles difficiles, d'une fonction à sens unique.
  - Soient  $\mathcal{E}$  un ensemble fini, et  $f_0$  et  $f_1$  deux permutations sur  $\mathcal{E}$  (i.e. deux fonctions bijectives sur  $\mathcal{E}$ ) à sens unique. On appelle rencontre entre  $f_0$  et  $f_1$  tout couple  $(a, b) \in \mathcal{E} \times \mathcal{E}$  tel que  $f_0(a) = f_1(b)$ . On dit que les fonctions  $f_0$  et  $f_1$  ne se rencontrent pas s'il est algorithmiquement difficile de trouver une rencontre entre  $f_0$  et  $f_1$ .
- À partir d'une paire de telles fonctions à sens unique qui ne se rencontrent pas, et d'un élément  $e \in \mathcal{E}$  fixé, on construit une fonction  $h$  sur l'ensemble de toutes les chaînes binaires finies  $\{0, 1\}^*$  de la façon suivante :

$$h : \begin{array}{ccc} \{0, 1\}^* & \longrightarrow & \mathcal{E} \\ x = x_1 x_2 \dots x_k & \mapsto & f_{x_1}(f_{x_2}(\dots f_{x_k}(e) \dots)) \end{array}$$

Montrer que la fonction  $h$  ainsi construite est à collisions fortes difficiles.

<sup>1</sup>On rappelle qu'un algorithme d'authentification est dit "zero-knowledge" si le fait de l'utiliser (pour s'authentifier) ne révèle rien sur ses propres secrets.

$p_0 + p_1 = 1$   
 $p_0 + p_1 = 1$

**Exercice 3** (Algorithme AES) (4 points)

1. Rappeler comment est défini le produit de deux octets dans l'algorithme AES.
2. Calculer  $\{C5\} \times \{B2\}$ .

**Exercice 4** (Chiffrement/déchiffrement RSA) (4 points)

Dans toute la suite, on pourra utiliser les résultats numériques suivants :

- $319 = 11 \times 29$  ;  $10^{11} = 263 \pmod{319}$  ;  $263^2 = 216 \times 319 + 265$  ;
- $133^3 = 12 \pmod{319}$  ;  $133^{25} = 133 \pmod{319}$  ;
- $11^2 = 121 \pmod{280}$  ;  $11^4 = 81 \pmod{280}$  ;  $11^8 = 121 \pmod{280}$  ;  $11^{16} = 81 \pmod{280}$  ;
- $95 = 64 + 31$  ;  $81 \times 11 = 51 \pmod{280}$  ;  $81 \times 121 = 1 \pmod{280}$ .

On considère la clé publique RSA  $(11, 319)$ , c'est-à-dire pour  $n = 319$  et  $e = 11$ .

1. Quel est le chiffrement avec cette clé du message  $M = 100$  ?
2. Calculer  $d$  la clé privée correspondant à la clé publique  $e$ .
3. Déchiffrer le message  $C = 133$ .

$$y \equiv x^e \pmod{n}$$
$$y \equiv x^d \pmod{n}$$
$$x^7 \equiv x^6 + x^2 + 1 \pmod{n}$$
$$x^7 \equiv x^5 + x^4 + x \pmod{n}$$

**Consignes :**

- Durée : 1h30.
- Documents interdits. Aucun accès à une calculatrice, un téléphone portable, un smartphone, ou tout autre dispositif électronique, connectable ou non.

**Exercice 1** (5 points)

1. Quelle est la taille de la clé de l'algorithme DES ? Cette taille est-elle suffisante ? Expliquer.
2. Vous recevez un message chiffré  $y$  de 64 bits. Tout ce que vous savez est que c'est le résultat du chiffrement d'un certain message  $x$  (que vous ne connaissez pas) par l'algorithme DES. Sachant cela, combien  $y$  a-t-il (au maximum) de possibilités pour le message  $x$  ? Expliquer.
3. Même question en remplaçant le DES par le Triple-DES.

**Exercice 2** (5 points)

1. Rappeler comment fonctionne le mode de chiffrement CBC.
2. Expliquer quel est le rôle de la "valeur initiale" IV.
3. Montrer que, dans ce mode CBC, un attaquant peut modifier un bloc du chiffré de façon à ce que seuls deux blocs du clair soient modifiés.

**Exercice 3** (5 points)

1. Quelles sont les tailles de l'entrée, de la sortie et de la clé dans l'algorithme AES ? Combien y a-t-il de tours ?
2. Rappeler comment est défini le produit de deux octets dans l'algorithme AES.  
[Rappel : le polynôme  $X^8 + X^4 + X^3 + X + 1$  intervient dans la définition.]
3. Calculer  $\{54\} \times \{7E\}$ .

**Exercice 4** (5 points)

1. Donner la définition d'une fonction à collisions fortes difficiles.
2. Que dit le paradoxe des anniversaires ? Donner le plus de détails que vous pouvez.
3. Soit  $f : \{0, 1\}^m \rightarrow \{0, 1\}^m$  une fonction quelconque. On définit  $h : \{0, 1\}^{2m} \rightarrow \{0, 1\}^m$  de la manière suivante. Pour toute valeur  $x \in \{0, 1\}^{2m}$  :
  - on écrit  $x = x_1 || x_2$  avec  $x_1, x_2 \in \{0, 1\}^m$ .
  - on pose alors  $h(x) = f(x_1) \oplus f(x_2)$ .

Montrer que  $h$  n'est pas à collisions fortes difficiles.

**Question 3.1.3 :** Que pouvez-vous dire sur la sécurité de deux clefs  $PK = (e, N)$  et  $PK' = (e', N')$  telles que  $N = p \cdot q \cdot r$  et  $N' = p \cdot q \cdot r'$  ?

**Question 3.1.4 :** Que vaut le chiffré du message  $M = 6$  avec la clef  $PK_1$  ?

**Question 3.1.5 :** Que vaut le clair du chiffré  $C = 5$ , obtenu à l'aide de la clef  $PK_2$  ?  
(Indication : Pour réaliser cela, vous pouvez calculer  $d_2 = e_2^{-1} \pmod{N_2}$ .)

### 3.2 Restes Chinois

On va s'intéresser à la résolution d'un système de congruences de la forme suivante :

$$(S) \left\{ \begin{array}{l} (E_1) : x \equiv a \pmod{p} \\ (E_2) : x \equiv b \pmod{q} \end{array} \right.$$

où  $p$  et  $q$  sont deux entiers premiers entre eux.

D'après le théorème de Bézout, on sait qu'il existe deux entiers  $u$  et  $v$  tels que  $p \cdot u + q \cdot v = 1$ . On rappelle que par définition,  $d \equiv e \pmod{l}$  se traduit par : il existe un entier  $k$  tel que  $d = e + k \cdot l$ .

**Question 3.2.1 :** Réécrivez le système  $(S)$  en utilisant la définition précédente. Les deux équations ainsi obtenues seront notées  $(E'_1)$  et  $(E'_2)$ .

**Question 3.2.2 :** Calculer  $(E'_1) \cdot (q \cdot v) + (E'_2) \cdot (p \cdot u)$ . En déduire que

$$x \equiv a \cdot v \cdot q + b \cdot u \cdot p \pmod{(p \cdot q)}.$$

Dans la suite on va voir en quoi ce type de résolution présente un intérêt cryptographique.

**Question 3.2.4 :** Rappelez le fonctionnement du cryptosystème RSA (Génération des clefs, Chiffrement et Déchiffrement).

**Question 3.2.5 :** Afin d'accélérer le chiffrement (et le déchiffrement), on utilise l'algorithme suivant dans lequel  $\ell_i$  désigne le bit en position  $i$  de l'entier  $\ell$  de  $k$  bits :

```

Data:  $x, \ell, n$  avec  $\ell = \sum_{i=0}^{k-1} \ell_i 2^i$ 
Result:  $x^\ell \pmod{n}$ 
result =  $x$ ;
for  $i = k - 1$  down to 0 do
    result =  $result^2 \pmod{n}$ ;
    if  $\ell_i = 1$  then
        result =  $x \times result \pmod{n}$ ;
    end
end
return result

```

Algorithm 1: Algorithme d'exponentiation rapide

Donnez le nombre de multiplications modulaires effectuées par cet algorithme.

Pour déchiffrer un message  $C = M^e \pmod{N}$ , on peut résoudre le système modulaire suivant :

$$(S) \left\{ \begin{array}{l} M \equiv C^d \pmod{p} \\ M \equiv C^d \pmod{q} \end{array} \right.$$

puis reconstituer le résultat à la manière de la question 3.2.3.

Dans la suite vous supposerez que la multiplication modulaire modulo un entier  $n$  a une complexité en  $O(\ln^2 n)$ . On supposera également que les entiers premiers  $p$  et  $q$  dont le produit constitue le module RSA considéré, ont la même taille en bits.

**Question 3.2.6 :** Quel est le facteur d'accélération offert par l'utilisation de la résolution de système modulaire par rapport à un déchiffrement modulo  $N$  (on négligera le temps de reconstruction) ?

Dans la suite on admettra que l'on peut résoudre tout système modulaire lorsque les équations ont leurs modulos tous premiers entre eux.

**Question 3.2.7 :** En faisant les mêmes hypothèses qu'à la question 3.2.6, quel est le facteur d'accélération offert par l'utilisation de la résolution de système modulaire par rapport à un déchiffrement modulo  $N$  (on négligera le temps de reconstruction) dans le cas de RSA multiprime à trois premiers ?

### 3.3 Considérations sur la PKI

Considérons l'utilisation de ce schéma dans le cadre du déploiement d'une infrastructure cryptographique dans laquelle  $\ell$  entités doivent pouvoir communiquer de manière sécurisée.

**Question 3.3.1 :** Si l'on utilise un système de chiffrement basé sur de la cryptographie à clef secrète, quel sera le nombre  $N$  de clefs permettant aux  $\ell$  entités de pouvoir communiquer entre elles deux à deux ?

**Question 3.3.2 :** Même question si on utilise de la cryptographie à clef publique que vaudra alors le nombre  $N$  de clefs à générer ?

**Question 3.3.3 :** À la lumière de la question précédente, quel type de cryptographie vous semble le plus adapté à la situation ?

On suppose dans la suite que l'on va utiliser le schéma RSA multiprime à trois premiers afin de permettre aux  $\ell$  entités de communiquer entre elles. Chacun des  $\ell$  utilisateurs du système possédera donc une paire de clefs publique et privée  $(PK_u, SK_u)$  de la forme  $PK_u = (e, N = p \cdot q \cdot r)$  et  $SK_u = (d, N)$  telles que  $d = e^{-1} \pmod{N}$ .

**Question 3.3.4 :** En s'appuyant sur la question 3.1, donnez le nombre d'entiers premiers différents à générer afin de garantir une sécurité satisfaisante.

On va maintenant s'intéresser à la faisabilité du déploiement d'une telle infrastructure : existe-t-il suffisamment d'entiers premiers pour déployer une telle infrastructure ? Le nombre de nombres premiers inférieurs à un entier  $n$  sera noté  $\pi(n)$ . On considérera dans la suite que  $\pi(n) = \frac{n}{\ln_2(n)}$ .

**Question 3.3.5 :** À l'aide de la formule précédente, donnez le nombre d'entiers premiers compris entre  $2^m$  et  $2^{m+1}$ .

**Question 3.3.6 :** À l'aide du résultat de la question précédente et en considérant que les entiers premiers qui constituent les clefs RSA multiprime sont sur 680 bits, vous semble-t-il possible d'utiliser le cryptosystème RSA multiprime à trois premiers si  $\ell = 10^6$  ?

### 3.4 Question subsidiaire

**Question 3.4 :** À votre avis, pourquoi ce cryptosystème n'est-il pas utilisé ?