EPITA_ING1_2015_S2_FMSI

Fondements mathématiques pour la sécurité informatique

Corrigé ducontrôle du cours de Jean-Luc Stehlé

Juin 2013

Contrôle sans aucun document ni ordinateur. Total : 8 questions sur 2 pages. Durée : 1h30
Les calculettes sont autorisées.

Attention, les mauvaises réponses seront pénalisées par des points négatifs. En effet, dans la vie professionnelle d'un ingénieur, il vaut mieux avouer son ignorance que de raconter une bêtise.

Barème total sur 105 points : A=24 points ; B=26 points ; C=55 points

Remarques du correcteur :

Le sujet était trop long pour 1h30 ; Il aurait fallu prévoir 3h pour le contrôle.
Très peu d'étudiants ont abordé la partie C

Nous avons considéré que quelqu'un qui avait bien répondu aux questions A et B méritait la note maximale.
Il a donc été décidé de multiplier la note sur 105 selon barème initialement prévu par le coefficient 0.45, puis de faire un arrondi par partie entière supérieur pour obtenir une note finale sur 20.
Ainsi un étudiant qui aurait eu un total de 43 points sur 105 se retrouve avec une note finale de 20/20. Ceci pouvait s'obtenir uniquement avec les parties A et B (notées au total sur 50).

Ce barème est injuste pour les quelques très bons étudiants, car il aboutit dans certains cas à des notes supérieures à 20/20 qu'il faut bien écrêter à 20/20.

PARTIE A : Chiffrement par bloc avec chaînage. Total sur 24 points

Rappels de cours : On rappelle les divers modes de chaînage et/ou d'utilisation des algorithmes par bloc

ECB	(Electronic Code Book)
CBC	(Cipher Block Chaining)
CTS	(Cipher Text Stealing = Vol de texte chiffré)
CTR	(CounTeR = CompTeuR)

Question 1. On veut chiffrer (sans compression) un message de 50 octets.
De combien d'octets se composera le message chiffré selon le type de chiffrement utilisé ?

a) DES en mode ECB
b) DES en mode CBC
c) DES en mode CTS
d) DES en mode CTR
e) Triple DES en mode ECB
f) Triple DES en mode CBC
g) Triple DES en mode CTS
h) Triple DES en mode CTR
i) AES en mode ECB
j) AES en mode CBC
k) AES en mode CTS
l) AES en mode CTR

12 points

Question 2. Mêmes questions pour un message de 10 octets.

12 points

Rappel de cours : Un algorithme par blocs chiffre toujours un nombre entier de blocs.
- *Si le message est plus petit qu'un bloc, il faut compléter à un bloc (par exemple en rajoutant des bits à zéro) et on transmet minimum un bloc.*
- *Si le message est plus grand qu'un bloc, le mode CTS est une astuce consistant à ne pas transmettre entièrement l'avant dernier bloc, de façon que le texte chiffré ait la même longueur que le texte clair. Mais il faut bien sûr que le message soit plus long qu'un bloc.*
- *Exception : le mode CTR : l'algo de chiffrement sert à générer un masque XOR, et le texte chiffré a alors toujours la même longueur que le texte en clair.*

DES et triple DES utilisent des blocs de 8 octets, AES utilise des blocs de 16 octets.
D'où les réponses aux questions 1 et 2

Réponses question 1 : 56 56 50 50 / 56 56 50 50 / 64 64 50 50

Réponses question 2 : 16 16 10 10 / 16 16 10 10 / 16 16 16 10

PARTIE B : Chiffrement par AES. Total sur 26 points

Rappels de cours sur le système de chiffrement AES

On note Z/2Z [X] l'algèbre des polynômes à une variable sur le corps à deux éléments Z/2Z, et on considère qu'un octet en AES représente un élément du corps quotient de Z/2Z [X] par l'idéal <m> engendré par le polynôme $m[X] = X^8 + X^4 + X^3 + X + 1$. *Ce corps sera noté K dans la suite de ce contrôle. Un élément de K est représenté par la valeur hexadécimale entre accolades de l'octet correspondant. Par exemple {72} (soit en binaire 0111 0010) représente le polynôme* $X^6+X^5+X^4+X$, *modulo m[X] sur le corps Z/2Z*

Dans le corps K les opérations d'addition et de multiplication sont l'addition et la multiplication des polynômes dans Z/2Z [X], modulo le polynôme m[X]. Ces opérations sont respectivement notées $\oplus$ *et* $\bullet$.

En AES, un mot de 32 bits (4 octets) représente un élément du quotient de l'algèbre K[X] des polynômes sur K, quotientée par l'idéal engendré par le polynôme X^4+1 . *Un tel élément est représenté par la suite des 4 octets, par ordre de degré décroissant.*

Par exemple {18}{AC}{62}{2A} représente le polynôme $\{18\}X^3+\{AC\}X^2+\{62\}X+\{2A\}$, *modulo* X^4+1 *sur le corps K .*

Question 3. On demande le résultat dans K des calculs suivants

a) {18} $\oplus$ {18}
b) {03} $\bullet$ {02}
c) {18} $\bullet$ {02}
d) {81} $\bullet$ {02}
e) {18} $\bullet$ {03}
f) {23} $\bullet$ {26}

Détail des calculs

a) 0001 1000 $\oplus$ 0001 1000 = 0000 0000 = {00}

b) 0000 0011
0000 0010
0 0000 011. = {06}

c) 0001 1000
0000 0010
0 0011 000. ={30}

d) 1000 0001
0000 0010
1 0000 001.
Pour tuer le bit X^8, il faut faire une réduction modulo <m> = 1 0001 1011
Pour ce faire, on ajoute ici (par l'opération $\oplus$) <m>
1 0000 0010
$\oplus$ 1 0001 1011
0001 1001 = {19} Cette question nécessitait plus de calculs et est comptée double

```
e) 0001 1000
   0000 0011
   0001 1000
 0 0011 000.
   0010 1000   ={28}

f)       0010 0011
         0010 0110
       0 0100 011.
      00 1000 11..
  0 0100 011.. ....
    0100 1010 1010
```

Pour tuer le bit de gauche (qui représente X^{10}) on fait une réduction modulo <m> = 1 0001 1011
Pour ce faire, on ajoute (par l'opération ⊕) m × X^2 (donc décalé de 2 bits vers la gauche

```
⊕   0100 0110 1100
    0000 1100 0110 ={C6}
```

Cette question nécessitait plus de calculs et est comptée double
Notons que ce résultat est réutilisé dans la question suivante

Réponses attendues : {00} {06} {30} {19} {28} {C6} **8 points** (questions d) et f) comptent double)

Question 4. La multiplication par {04} dans K est équivalente à

(A) Un décalage de deux bits dans le sens des poids forts
(B) Un décalage de deux bits dans le sens des poids faible
(C) Une permutation circulaire des bits
(D) Une multiplication par 4 modulo 256
(E) Autre chose

Décalage de 2 bits vers la gauche suivie d'une réduction modulo <m> **6points**
Expliquez en 3 lignes maximum la raison pour laquelle vous avez choisi cette réponse.

Question 5. Dans l'algèbre des mots de 32 bits utilisée en AES, c'est-à-dire l'algèbre des polynômes sur K , modulo X^4+1, on demande de calculer le produit des deux mots suivants :

P = {23}{00}{01}{02} Q = {26}{00}{2F}{01}

12 points

Calcul du produit des deux polynômes **Puis réduction modulo X^4+1**

```
Cst  : {02}•{01}                                  = {02}     ⊕ {C0} = {C2}
X¹   : {02}•{2F} ⊕ {01}      = {5E} ⊕ {01}        = {5F}     ⊕ {00} = {5F}
X²   : {2F}                                       = {2F}     ⊕ {C6} = {E9}
X³   : {23} ⊕ {02}•{26}      = {23} ⊕ {4C}        = {6F}            = {6F}
X⁴   : {23}•{2F} ⊕ {26}      = {E6} ⊕ {26}        = {C0}
X⁵   : {00}                                       = {00}
X⁶   : {23}•{26}                                  = {C6}
```

Détail des calculs intermédiaires

N.B. : Dans les multiplications • , on a toujours intérêt à mettre en second facteur le terme qui a le moins de bits à 1.

```
X¹:  {2F}      =   0010 1111
     {02}      =   0000 0010
   {2F}•{02}   = 0 0101 111.  = {5E}  (Pas de réduction modulo <m> nécessaire ici)
     ⊕ {01}    =   0000 0001
               =   0101 1111  = {5F}

X³:  {26}      =   0010 0110
     {02}      =   0000 0010
   {26}•{02}   = 0 0100 110.  = {4C} (Pas de réduction modulo <m> nécessaire ici)
     ⊕ {23}    =   0010 0011
               =   0110 1111  = {6F}
```

```
X⁴:  {2F}      =      0010 1111
     {23}      =      0010 0011
                      0010 1111
                 0 0101 111.
               0 0101 111. ....
                 0101 1001 0001   Il faut faire une réduction modulo <m> pour tuer les bits X⁸ et X¹⁰
  ⊕ <m>×X²       .100 0110 11..   Ici on ajoute <m> décalé de 2 bits vers la gauche ce qui tue X¹⁰
  ⊕ <m>             1 0001 1011   Ici on ajoute tout simplement  <m> ce qui tue X⁸
                    1110 0110  = {E6}
    ⊕ {26}      =      0010 0110
                =      1100 0000  = {C0}

X⁶:  {23}•{26} = {C6} déjà calculé à la question 3f.
```

Réponse attendue : {6F}{E9}{5F}{C2}

PARTIE C : Estimation de temps de calcul *Total sur 55 points*

Dans toute la suite, on ne demande pas une valeur précise, mais un ordre de grandeur. Toute réponse à ±25% sera considérée comme correcte. Mais on demande, pour chaque question, quelques lignes de brèves explications justifiant comment on a trouvé ce résultat.

On travaille avec un processeur 32 bits, cadencé à 4 GHz. On appellera multiplication élémentaire l'opération consistant à multiplier deux entiers non signés à 32 bits pour fournir un résultat stocké sur deux registres de 32 bits. Dans tous les calculs d'ordre de grandeur des temps de calcul, on ne tiendra compte que du nombre de multiplications élémentaires. On négligera donc les additions (qui en général seront simultanées aux multiplications, car les registres résultats fonctionneront comme des accumulateurs) ainsi que les calculs d'indice, les transferts registre mémoire etc. On admettra qu'une multiplication élémentaire se fait en moyenne en 7 tops d'horloge (y compris les additions dans les registres résultats, calculs d'indices, …)

On veut mettre en œuvre un système de chiffrement à clé publique RSA. On travaille dans des arithmétiques modulo N, où N est un entier à n=2048 bits, dont le bit de poids fort est égal à 1 (donc $2^{2047} \le N < 2^{2048}$). On estime qu'une réduction modulo N prend approximativement le même temps que la multiplication de deux entiers à n bits. En conséquence, on estimera qu'une multiplication modulo N nécessite un temps de calcul équivalent approximativement au double de celui d'une multiplication de deux grands nombres (à n bits).

On utilisera en RSA un exposant public égal à $d = 2^{16}+1$ et un exposant privé c de l'ordre de grandeur de N, ayant approximativement autant de bits à 1 que de bits à 0 .

Calculs préliminaires utiles pour les questions suivantes : Temps de calcul d'une multiplication dans Z/NZ

Une multiplication élémentaire prend 7 tops d'horloge soit 1.75×10^{-9} secondes.
Une « grande » multiplication c'est-à-dire une multiplication de deux nombres à 2048 bits (stockés sur q=64 mots) représente $q^2=4096$ multiplications élémentaires soit environ 7×10^{-6} secondes.
Une multiplication modulo N nécessite en plus une réduction modulo N (approximativement égal à celui d'une grande multiplication),

Donc au total, une multiplication modulo N nécessite environ 14×10^{-6} secondes.

Question 6. Quels sont approximativement les temps de calcul

a) pour le chiffrement d'un message de 1 Mo (Megaoctet) utilisant la clé publique du destinataire ?

Élever, modulo N, x à la puissance $2^{16}+1$ nécessite 17 multiplications modulo N (on élève 16 fois de suite au carré puis on multiplie par x), soit un temps de $16 \times 14\times10^{-6}$ secondes soit environ 224×10^{-6} secondes.
Cette opération permet de traiter un bloc, soit 2048 bits soit 256 octets. Pour traiter 1 Mo, soit 4096 blocs, il faut donc environ un peu plus de 4000 fois plus soit environ 1 seconde.

b) pour le déchiffrement de ce message par le destinataire (en utilisant sa clé privée).

Élever, modulo N, x à la puissance c nécessite un nombre de multiplications égal au nombre de bits de c (soit 2048) augmenté du nombre de bits à 1 de cet exposant (soit environ 1024), soit environ 3072 là, où dans la question a) il en suffisait de 17 , donc 3072/17 fois plus de temps qu'à la question précédente soit environ 180 secondes, soit 3 minutes.

Les calculs ont été faits approximativement et de tête, et en arrondissant à chaque étape, mais on est sûr d'être relativement proche du résultat exact et certainement moins loin que ±25%

Réponses attendues : 1 seconde, 3 minutes 15+15 points

Question 7. On veut chiffrer un flux de données par RSA en utilisant la clé publique du destinataire. Quel débit maximal, en Megabits par seconde, peut-on atteindre en supposant que toute la puissance du processeur est dédiée au chiffrement ?

On vient de voir (question 6a) que le chiffrement d'un Mégaoctet (8 Megabits) nécessitait 1 seconde.

Réponse attendue :8 Megabit par seconde,
5 points

Question 8. Les messages sont signés par une signature comportant une empreinte (hashcode ou checksum ou autre) du message, suivi d'informations comme le nom du signataire, la date de cette signature et divers autres données. L'ensemble de la signature fait 384 octets. Cette signature est chiffrée par l'expéditeur avec sa clé privée et sera déchiffrée par le destinataire en utilisant la clé publique.

a) Quel est le temps de calcul (de chiffrement) de cette signature par l'expéditeur ?
b) Quel est le temps nécessaire au destinataire, lors de la vérification, pour déchiffrer cette signature ?

On utilise un algorithme RSA de chiffrement déchiffrement qui fonctionne sur des blocs de 2048 bits (soit 256 octets).
Pour chiffrer 384 octets il faut donc chiffrer 2 blocs (et ce même si un utilise un chaîne de type CTS pour économiser de la bande passante lors de la transmission)

En refaisant les calculs de la question 6, on sait que pour traiter avec la clé publique (ici déchiffrer) un bloc, il faut environ 225×10^{-6} secondes. Vérifier (question b) nécessite de traiter deux blocs, et il faut donc environ 0.450 milliseconde

Pour chiffrer (question a) 3072/17 fois plus de temps (revoir 6b) soit environ 86 millisecondes

Réponse attendue : a) 86 millisecondes, b) 0.5 millisecondes

10 + 10 points

Rappel : On ne demandait pas une valeur précise, mais un ordre de grandeur, les calculs pouvant être faits de tête pour ceux qui n'avaient pas de calculette. Toute réponse à ±25% sera considérée comme correcte.