Intro à la Crypto 2017
Mathématiques et MSIAM

Mars, controle continu

Exercice 1

On rappelle le petit théorème de Fermat.

Si p est premier, alors pour tout x non nul de $\mathbb{Z}/p\mathbb{Z}$, on a $x^{p-1} \equiv 1\,[mod\, p]$.

1 – p est un nombre premier. Expliquer comment savoir si un element de $\mathbb{Z}/p\mathbb{Z}$ est le carré d'un autre, ou pas. Le calcul est-il rapide (et pourquoi) ?

2 – p est un nombre premier congru à 3 modulo 4. En essayant $x^{\frac{p+1}{4}}$, expliquer s'il est facile ou pas de calculer les deux racines carrées de x si c'est un carré. Le calcul est-il rapide (et pourquoi) ?

Exercice 2

On considère le code correcteur d'erreur de $(\mathbb{F}_2)^n$ donné par la matrice de contrôle d'une seule ligne :

$$(\alpha, \alpha^2, \ldots, \alpha^{2^n-1})$$

où $\alpha \in \mathbb{F}_{2^n}$ est primitif dans $\mathbb{F}_{2^n}^*$

1 – Quels sont les parametres du code ? (dimension, longueur, force)

2 – Quelle est selon vous la difficulté pour mettre en place ce code ?

3 – Décrire ce code pour $n = 2$, et pour $n = 3$ (selon temps disponible).

Exercice 3 – TP
Vous êtes Charles, un observateur indiscret d'un processus entre Alice et Bob. Ces derniers utilisent un générateur pseudo-aléatoire qui est un registre à décalage. Vous avez eu l'information que le nombre de registre qu'ils utilisent est 10.
Vous avez intercepté 20 valeurs conécutives de la suite pseudo-aléatoire.

$$0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 1, 1, 0, 0, 1, 0, 1, 0, 1, 0$$

Votre mission est de retrouver la matrice du système.

Il circule l'anecdote suivante. Les machines à sous de certains casinos utilisaient ce genre de générateurs pseudo-aléatoires. Certaines personnes audacieuses ont acheté une vieille machine, l'ont testée pendant longtemps « gratuitement » pour trouver la periode du processus. Ils en ont déduit un petit nombre de valeurs possibles pour m. Puis ces personnes sont allées dans un casino utilisant des machines de la même marque, et ont déterminé les coefficients des registres de l'une d'elles grâce à un début de jeu (comme dans le TP), puis ont anticipé les valeurs sorties dans la suite. L'histoire ne dit pas combien de temps ils ont pu jouer avant d'être virés du casino...

Exercice 4 – TP
1 – Remplir une matrice M à coefficients dans $\mathbb{F}_2$ de 15 colonnes non nulles toutes differentes. C'est la matrice de contrôle d'un code C.

Créer (à volonté) un vecteur colonne aléatoire, à coefficients dans $\mathbb{F}_2$ de 15 coordonées.

Tester s'il est dans le code C.

S'il ne l'est pas, en supposant qu'il n'est qu' à distance de Hamming 1 du code, trouver l'élément du code qui lui est le plus proche.

Pour rendre votre TP
Nommer votre feuille de calcul des deux noms de famille de votre binôme, suivie de l'année. (Exemple : MartinDupont2017)

Exportez votre feuille de calcul : revenir à l'accueil de Sage en cliquant sur « Sage The Sage Notebook » en haut à gauche, selectionner la case de votre feuille, et cliquer « download » au dessus de « active worksheets ». Cela sauve un fichier .sws dans le repertoire des téléchargements.

Envoyez moi ce fichier à l'adresse francois.dahmani@univ-grenoble-alpes.fr avec comme sujet « [CC Crypto 2017] MartinDupont2017 »

Examen 2017 -- Introduction à la crypto.

*M1-Maths-M1-MSIAM ; **2 heures**, une feuille A4 manuscrite recto-verso autorisée. Ce sujet comporte 2 pages, et 3 exercices.*

EXERCICE 1

1 -- Dans la processus suivant, décrivez ce qui se passe, dites si les calculs sont rapides ou pas, à quoi sert le processus total, et à quoi s'attendre comme résultat. (Rappel randint(a,b) renvoie un entier aléatoire dans [a,b])

2 -- Indiquez deux maladresses du rédacteur de ces lignes (indiquez les numeros de ligne et la raison). Proposez des corrections pour ces deux problèmes. --

```
M=64
a=randint(1,2^M);   P1=next_prime(a)
a2=randint(1,2^M);  P2=next_prime(a2)
N=P1*P2; print N
PhiN=euler_phi(N)
e=3
while (gcd(e,PhiN)!=1):
    e=e+1
print e
V= xgcd(e,PhiN)
d0=V[1]
M=11111111111111111111
Me=mod(M^e,N)
Med = mod(Me^d0,N)
print(Me)
print(Med)
```

EXERCICE 2 (theorique).

**Dans un protocole RSA, Alice veut envoyer le même message M à Bob, Charles, et David.
Les clés publiques de Bob, Charles et David sont (n=nB, e=3), (nC, 3), (nD, 3).
Le message M est un nombre inferieur à chaque nB, nC, nD. Elle decide de le transmettre selon RSA sans autre modification (pas de metadonnées, pas de padding, juste mod(M^3, nB), mod(M^3, nC), et mod(M^3, nD)).**

Eve a tout vu, tout entendu, et est très savante.

1 -- Eve peut-elle savoir facilement si deux nombres parmi nB, nC, nD ont un facteur premier commun ? Peut-elle calculer facilement M dans ce cas ?

2 -- On suppose que nB, nC, nD n'ont pas de facteur premier commun. Expliquer comment Eve peut retrouver le message M facilement. --

```
# Rappel, pour l'exercice suivant
Integers(57)
```

```
Ring of integers modulo 57
```

EXERCICE 3

1 -- Expliquer le détail des calculs suivants (3 prochaines cases de SageMath), en expliquant clairement ce qui se passe.

2 -- Quel problème essaye-t-on de résoudre ? Citer des protocoles dans lesquels cette attaque est pertinente.

3 -- Il est possible que les resultats A et B ne soient pas égaux. Expliquer pourquoi, dans quelle circonstances cela se produit (ou cela ne se produit pas) et donner une identité qui est tout de même verifiée.

4 -- Selon vous quel est le principal problème si le parametre M est trop grand ? --

```
M=32
a=randint(1,2^M); P=next_prime(a); print(P)
R=Integers(P); alpha=R.random_element(); print(alpha)
x=randint(1,P-1); beta=alpha^x; print(beta)
```

```
573903767
519997285
223374740
```

```
m=floor(sqrt(P))+1
gamma=alpha
L=[alpha]
for j in range (0, m):
    gamma=gamma*alpha
    L.append(gamma)
```

```
gamma=beta
hit=False
for r in range (0, m):
    for j in range (0, len(L)):
        if gamma==L[j]:
            hit=true
            k=j
            s=r
    gamma=gamma*alpha^(-m)
if hit:
    A=(k+1)+m*s ; B=x
    print(A) ; print(B)
```