Examen de cryptographie (3h)

Vous disposez de la moitié du temps pour résoudre les deux premiers exercices et de l'autre moitié pour résoudre l'exercice 3.
Durant la première partie de l'examen, vous n'avez pas accès aux documents ni aux ordinateurs. Pour la deuxième partie, vous devez utiliser un ordinateur et programmer sous sagemath dans votre environnement habituel. Votre fichier informatique doit être téléversé à la fin de l'épreuve sur

`https://im2ag-moodle.univ-grenoble-alpes.fr/course/view.php?id=528`

Exercice 1 : bits faciles et difficiles pour le logarithme discret

Soit p un grand nombre premier. Pour tout entier x compris entre 0 et $p-1$, son **bit de poids faible**, noté $\mathrm{lsb}(x)$ (*least significant bit*), est le dernier chiffre de x dans son écriture en base 2 ; on a donc

$$\mathrm{lsb}(x) = \begin{cases} 0 & \text{si } x \text{ est pair} \\ 1 & \text{si } x \text{ est impair} \end{cases}$$

Le **bit de poids fort** de x, noté $\mathrm{msb}(x)$ (*most significant bit*), est défini de manière un peu différente :

$$\mathrm{msb}(x) = \begin{cases} 0 & \text{si } x < \frac{p-1}{2} \\ 1 & \text{si } x \geq \frac{p-1}{2} \end{cases}$$

On considère un générateur g du groupe cyclique $(\mathbb{Z}/p\mathbb{Z})^*$, et un élément $h \in (\mathbb{Z}/p\mathbb{Z})^*$. On note $x \in [\![0, p-2]\!]$ le logarithme discret de h en base g, mais celui-ci n'est pas connu a priori.

1. Démontrer les équivalences :

   $$\mathrm{lsb}(x) = 0 \iff h \text{ est un carré dans } (\mathbb{Z}/p\mathbb{Z})^* \iff h^{(p-1)/2} = 1 \bmod p$$

   En déduire une méthode efficace de calcul de $\mathrm{lsb}(x)$ et en donner la complexité en fonction de p.

2. On considère l'écriture binaire du réel $\frac{x}{p-1}$:

   $$\frac{x}{p-1} = (0, y_1 y_2 y_3 \dots)_2 = \sum_{k \geq 1} y_k 2^{-k}$$

   (a) Démontrer que $y_1 = \mathrm{msb}(x)$, puis que pour tout $j \geq 1$,

   $$y_j = \mathrm{msb}\left(2^{j-1} x - (p-1) \sum_{k=1}^{j-1} y_k 2^{j-1-k} \right)$$

   (b) On suppose que l'on dispose d'une fonction `msbDL`, qui, pour toute entrée $g' \in (\mathbb{Z}/p\mathbb{Z})^*$, renvoie le bit de poids fort du logarithme discret de g' en base g.
   i. Montrer que $y_j = \texttt{msbDL}(h^{2^{j-1}})$ pour tout $j \geq 1$.
   ii. Donner une méthode permettant de retrouver le logarithme discret x de h en base g après $\lceil \log_2(p-1) \rceil$ appels à la fonction `msbDL`. On pourra commencer par montrer que pour tout $m \in \mathbb{N}$, on a l'égalité $(0, y_1 \dots y_m)_2 = \lfloor 2^m \frac{x}{p-1} \rfloor / 2^m$.

   (c) En déduire qu'il est possible de calculer (en temps polynomial en la taille de p) le bit de poids fort du logarithme discret d'un élément de $(\mathbb{Z}/p\mathbb{Z})^*$ si et seulement si il est possible de calculer (en temps polynomial en la taille de p) ce logarithme discret en entier.

3. On considère désormais un élément $a \in (\mathbb{Z}/p\mathbb{Z})^*$ dont l'ordre est un entier **impair** r.
   (a) En s'inspirant des questions précédentes, montrer que si l'on dispose d'une fonction `lsbDL` qui, pour toute entrée $b \in \langle a \rangle$, renvoie le bit de poids faible du logarithme discret en base a de b, alors il est possible de calculer le logarithme discret en base a d'un élément $h \in \langle a \rangle$ en $\lceil \log_2(r) \rceil$ appels à `lsbDL`.
   Indication : on pourra noter u un inverse de 2 modulo r, et $x = (x_m \dots x_1 x_0)_2$ le logarithme discret de h. Que vaut alors $\texttt{lsbDL}((h\, a^{-x_0})^u)$?
   (b) En déduire qu'il est possible de calculer (en temps polynomial en la taille de r) le bit de poids fort du logarithme discret d'un élément du sous-groupe $\langle a \rangle$ si et seulement si il est possible de calculer (en temps polynomial en la taille de r) ce logarithme discret en entier.

Exercice 2 : Décodage des codes de Reed-Solomon

On rappelle la définition suivante :

> Soient $k \leq n$ deux entiers, $\mathbb{F}_q$ un corps fini, et $x_1, \dots, x_n$ des éléments distincts de $\mathbb{F}_q$. Le code de Reed-Solomon sur $\mathbb{F}_q$ de paramètres (k, n) associé aux éléments $x_1, \dots, x_n$ est l'ensemble
>
> $$C = \{(P(x_1), \dots, P(x_n)) \mid P \in \mathbb{F}_q[X],\ \deg(P) < k\}$$

1. Rappeler pourquoi la distance minimale de C est égale à $n - k + 1$.

On note dans la suite $t = \lfloor \frac{n-k}{2} \rfloor$ la capacité de correction de C.

Soit $w = (y_1, \dots, y_n) \in (\mathbb{F}_q)^n$ un message reçu ; on suppose qu'il y a eu au plus t erreurs lors de la transmission. Décoder w revient alors à retrouver l'unique polynôme P de degré strictement inférieur à k tel que $P(x_i) = y_i$ pour au moins $n - t$ valeurs distinctes de i.

2. Soient $Q_0 = \sum_{j=0}^{n-t-1} c_j X^j$ et $Q_1 = \sum_{j=0}^{n-t-k} d_j X^j$ deux polynômes de $\mathbb{F}_q[X]$ de degrés strictement inférieurs à $n - t$ et à $n - t - k + 1$ respectivement.
   Montrer que, pour tout $i \in [\![1, n]\!]$, $Q_0(x_i) - y_i Q_1(x_i) = 0$ si et seulement si le $(2n-2t-k+1)$-uplet $(c_0, \dots, c_{n-t-1}, d_0, \dots, d_{n-t-k})$ est solution d'un système linéaire que l'on explicitera.
3. Montrer que ce système linéaire possède au moins une solution non nulle.
4. On suppose désormais que $(c_0, \dots, c_{n-t-1}, d_0, \dots, d_{n-t-k})$ est une solution non nulle du système linéaire ci-dessus, et donc que $Q_0(x_i) - y_i Q_1(x_i) = 0$ pour tout $i \in [\![1, n]\!]$.
   On note $I \subseteq [\![1, n]\!]$ le sous-ensemble des positions où il n'y a pas eu d'erreurs de transmission. Démontrer que $Q_0(x_i) - P(x_i) Q_1(x_i) = 0$ pour tout $i \in I$.
5. Montrer finalement que Q_1 est non nul et que $P = Q_0/Q_1$.
   Quelle est la complexité de cette méthode de décodage ?
6. Justifier que le polynôme Q_1 ainsi trouvé est un multiple de $\prod_{i \in [\![1,n]\!] \setminus I} (X - x_i)$ (parfois appelé polynôme localisateur d'erreurs).

Exercice 3 : Générateurs congruentiels linéaires

Les générateurs congruentiels linéaires (LCG) forment une famille très simple de PRNG, qui s'appuie sur les suites arithmético-géométriques. Étant donnés des paramètres a, c, m et une graine X_0, un tel générateur calcule les termes successifs de la suite (X_n) vérifiant la relation

$$X_{n+1} = aX_n + c \mod m$$

On peut ensuite renvoyer soit directement la suite (X_n) des états du générateurs, soit seulement une partie des bits des X_n. Un tel générateur a l'avantage d'être rapide et très simple, donc intéressant dans un environnement contraint (système embarqué) ou si l'on n'a pas besoin d'un pseudo-hasard très robuste. Cependant, la qualité d'un tel PRNG dépend énormément du choix des paramètres.

On considère donc une suite $(X_n)_{n\in\mathbb{N}}$, à valeurs dans $\{0, \dots, m-1\}$, vérifiant pour tout $n \in \mathbb{N}$ l'égalité $X_{n+1} = aX_n + c \mod m$.

1. Écrire en SageMath une fonction `LCG(X0,a,c,m,N)` qui renvoie la liste des N premières valeurs de la suite (X_n). Exemple :
```
>>> LCG(0,2,1,5,9)
[0,1,3,2,0,1,3,2,0]
```

I. Période : exemples

2. (a) Rappeler pourquoi la suite (X_n) est ultimement périodique, de période $T \leq m$.
   (b) Justifier que si a est premier avec m, alors la suite (X_n) est en fait périodique.

3. Un premier exemple.
   On considère le générateur donné par la relation de récurrence $X_{n+1} = 96\,X_n + 47 \mod 1309$, avec $X_0 = 1$.
   (a) Écrire un programme (suite d'instructions) permettant de trouver la période de cette suite.
   (b) Pour tout $n \in \mathbb{N}$, on pose $Y_n = X_n \mod 7$ avec $Y_n \in \{0, \dots, 6\}$ et $Z_n = X_n \mod 11$ avec $Z_n \in \{0, \dots, 10\}$.
   Afficher la liste des 300 première valeurs des suites (Y_n) et (Z_n) et donner (sans justification) leur période.
   (c) Pour tout $n \in \mathbb{N}$, on pose $U_n = X_n \mod 10$ avec $U_n \in \{0, \dots, 9\}$.
   Afficher la liste des 300 premières valeurs de la suite (U_n). Quelle(s) différence(s) peut-on observer avec le comportement des suites (Y_n) et (Z_n) ?
   Déterminer en quelques lignes de code la période de (U_n) (on pourra utiliser sans justification qu'il s'agit d'un diviseur de la période de la suite (X_n)).

4. Un deuxième exemple.
   Le générateur *RANDU*, introduit dans les années 1960 par IBM, utilisait la relation de récurrence $X_{n+1} = 65539X_n \mod 2^{31}$.
   (a) En partant d'une valeur de X_0 impaire de votre choix, afficher la liste des 200 premières valeurs de la suite (X_n), écrite en base 2. On utilisera la commande `n.digits(2)` pour obtenir cette écriture binaire, du bit de poids faible vers le bit de poids fort.
   (b) Observer la suite formée par les premiers bits, les deuxièmes bits, etc. Que peut-on constater ? Ce comportement persiste-t-il pour une autre valeur de X_0 ?
   (c) La suite (X_n) ainsi générée est-elle une bonne suite pseudo-aléatoire ?

II. Période : théorie

On considère toujours une suite $(X_n)_{n\in\mathbb{N}}$, à valeurs dans $\{0,\dots,m-1\}$, vérifiant pour tout $n\in\mathbb{N}$ l'égalité $X_{n+1} = aX_n + c \mod m$.

5. (a) Soit d un diviseur de m. Montrer que la suite $(X_n \mod d)_{n\in\mathbb{N}}$ (à valeurs dans $\{0,\dots,d-1\}$ ou $\mathbb{Z}/d\mathbb{Z}$) vérifie aussi une relation de récurrence.
   (b) En déduire que la suite $(X_n \mod d)_{n\in\mathbb{N}}$ est ultimement périodique, de période $T_d \leq d$.
   (c) Expliquer les comportements observés aux questions 3.b et 4.b.

6. Un cas particulier : $c = 0$ *(générateur de Lehmer).*
   (a) Donner la forme du terme général de la suite lorsque $c = 0$ et $a \wedge m = 1$.
   Que peut-on dire de la période de la suite ?
   (b) Soit $\prod_{i=1}^{r} p_i^{\alpha_i}$ la décomposition en nombres premiers distincts de m. Montrer que le maximum des périodes (à m fixé impair, et toujours pour $c = 0$) vaut $\mathrm{ppcm}(\varphi(p_1^{\alpha_1}),\dots,\varphi(p_r^{\alpha_r}))$.
   Comment choisir a et X_0 pour obtenir cette période ?

7. Un cas particulier : $m = p$ premier.
   (a) Montrer que si $a \neq 1 \mod p$, alors il existe un entier b tel que pour tout $n\in\mathbb{N}$,
   $$X_n = a^n(X_0 - b) + b \mod p.$$
   Que peut-on alors en déduire sur la période de la suite ?
   (b) Démontrer que la suite (X_n) est de période p si et seulement si $a = 1 \mod p$ et $p \nmid c$.
   Est-ce que la suite de période maximale ainsi obtenue est une bonne suite pseudo-aléatoire ?

III. Quelques faiblesses

8. Points et plans.
   Pour certaines applications comme le calcul d'intégrales multiples par méthode de Monte-Carlo, on a besoin d'interpréter les valeurs (X_n) renvoyées comme les coordonnées de points de l'espace, en posant $A_n = (X_{3n}, X_{3n+1}, X_{3n+2}) \in \{0,\dots,m-1\}^3$.
   On considère à nouveau le générateur *RANDU*, utilisant la relation de récurrence $X_{n+1} = 65539X_n \mod 2^{31}$.
   (a) En partant d'une valeur impaire de X_0, générer la liste des 1000 points $A_0,\dots,A_{999}$. On pourra représenter un point soit comme un triplet `(x,y,z)` soit comme une liste à trois éléments `[x,y,z]`.
   (b) À l'aide des commandes `points` ou `point3d`, qui prennent toutes deux en paramètres une liste de points, afficher le nuage de points correspondant et l'observer sous différents angles.
   En cas de difficultés d'affichage, on pourra "renormaliser" les points pour les mettre dans $[0,1]^3$ en divisant toutes les coordonnées par 2^{31}.
   Les points semblent-ils uniformément répartis dans le cube de côté de longueur 2^{31} ?
   (c) En remarquant que $65539 = 2^{16} + 3$, montrer que $X_{n+2} = 6X_{n+1} - 9X_n \mod 2^{31}$ pour tout $n\in\mathbb{N}$.
   (d) Pour $k\in\mathbb{Z}$, on note P_k le plan affine d'équation $9x - 6y + z = 2^{31}k$.
   En utilisant la question précédente et les valeurs prises par la suite (X_n), montrer que pour tout $n\in\mathbb{N}$, le point $A_n = (X_{3n}, X_{3n+1}, X_{3n+2}) \in \mathbb{R}^3$ appartient à l'union $\bigcup_{-5\leq k\leq 9} P_k$.
   Faire le lien avec la question (b).

9. Paradoxe des anniversaires.
   On suppose qu'un générateur congruentiel linéaire renvoie la suite des (X_n), de période T proche de m.
   Combien de temps faut-il attendre pour observer une répétition dans la liste des valeurs produites ? Expliquer pourquoi ce comportement n'est pas typique d'une vraie suite aléatoire à valeurs dans $\{0, \dots, m-1\}$.
   Quelle modification simple permettrait d'avoir un comportement plus aléatoire ?