Université Bordeaux 1 — **MHT839 – Master**
Mathématiques — Année 2011–2012

Devoir Surveillé, 4 avril 2012
Durée 2h.

Ce travail porte sur le problème de partage d'un secret.

Une personne détient un secret qu'elle souhaite léguer à un groupe de n individus, de telle sorte qu'un seul de ces individus ne puisse pas en profiter seul. Elle veut donc découper son secret en n parties, qui ne donneront accès au secret qu'une fois rassemblées. De plus, cette personne souhaite que si $n-1$ des légataires mettent leur partie du secret en commun, ils n'aient aucune information sur le secret lui-même. Elle va utiliser pour cela un algorithme d'évaluation d'un polynôme en n points, le décodage se fera alors par interpolation.

Plus précisément, soit K un corps fini de cardinal strictement supérieur à n, et soit $s \in K$ le secret à partager. Soit $A = K^{n-1}$. La fonction "partage" choisit au hasard un élément $a = (a_1 \ldots, a_{n-1}) \in A$, inconnu de tous, et forme le polynôme de $K[y]$

$$P_{a,s} = s + \sum_{i=1}^{n-1} a_i y^i.$$

Soit $t = (t_1, \ldots, t_n) \in K^n$, connu de tous, où les t_i sont non nuls et deux à deux distincts. Le l-ième légataire recevra l'élément $P_{a,s}(t_l)$ de K. Quand les légataires du secret mettent leurs parties du secret en commun, ils peuvent calculer $P_{a,s}$ par interpolation, puis $s = P_{a,s}(0)$.

1) (sur papier) Soient $s_1, \ldots, s_{n-1} \in K$. Quel résultat du cours permet-il d'affirmer que pour tout $k \in K$, il existe un unique $a \in K^{n-1}$ tel que pour tout $l \in \{1, \ldots, n-1\}$, $P_a(t_l) = s_l$ et tel que $P_a(0) = k$? Cela signifie que la connaissance de $n-1$ parties du secret ne donne aucun renseignement sur le secret lui-même.

2) Soit p un nombre premier. On prend ici $K = \mathbb{F}_p$. Soient $n < p$ un entier correspondant au nombre de légataires du secret et soit $s \in \mathbb{F}_q$ le secret à partager. Soit $t = [t_1, \ldots, t_n]$ où les $t_l \in \mathbb{F}_p$ sont non nuls et deux à deux distincts. Écrire une fonction `Partage` qui en entrée prend p, n, t et s et rend en sortie la liste des $P_{a,s}(t_l)$ (où a sera choisi au hasard à l'intérieur de la fonction).

3) Écrire une fonction `Decodage` qui retrouve le secret s à partir des t_l et des $P_{a,s}(t_l)$.

4) Appliquer les fonctions précédentes à $p = 10^7 + 19$, $n = 7$, $t = [1, 2 \ldots, n]$ et un secret s à choisir au hasard dans $\mathbb{F}_p$.

5) Soient p un nombre premier et $r \in \mathbb{N} \setminus \{0\}$ tels que $p^r > n$ (n est toujours le nombre de légataires du secret). Soit Q un polynôme irréductible unitaire de degré r de $\mathbb{F}_p[x]$ et soit $K = \mathbb{F}_p[x]/(Q) \simeq \mathbb{F}_{p^r}$. On note b l'image de x dans K (défini grâce à la fonction `alias`).

Écrire une procédure `PartageFq` suivant l'algorithme de partage d'un secret décrit ci-dessus. Cette procédure prendra en entrées n, p, r, b, une liste $t = [t_1, \ldots, t_n]$ d'éléments non nuls de K deux à deux distincts et le secret $s \in K$. Il donnera en sortie la liste $[s_1, \ldots, s_n]$ où pour tout l, l'élément s_l est la valeur en t_l de $P_{a,s}$. Remarquons que si $a = [a_1, \ldots, a_{n-1}]$ est la liste d'éléments qu'il faut prendre au hasard dans K, chacun des a_i est un polynôme en b de degré inférieur ou égal à $r-1$. Il en va de même de chacun des s_l obtenus.

6) Vérifier que $Q = x^{14} + x^5 + 1$ est un polynôme irréductible de $\mathbb{F}_2[x]$.

Appliquer la fonction `PartageFq` à $n = 7$, $p = 2$, $r = 14$, b l'élément algébrique sur $\mathbb{F}_2$ défini par Q, une liste t convenable et s un secret choisi au hasard dans K. Décoder le résultat obtenu.

7) [Partage avec une erreur] On s'intéresse maintenant à un problème différent. On suppose qu'il pourrait y avoir un problème de transmission dans le partage, et on souhaite que les légataires des parties du secret puissent retrouver celui-ci s'il y a au plus une erreur. On pose

alors $A = K^{n-3}$ (où $n \geqslant 3$). Soit $a = (a_1 \ldots, a_{n-3})$ un élément pris au hasard dans A, le polynôme $P_{a,s}$ est maintenant

$$P_{a,s} = s + \sum_{i=1}^{n-3} a_i y^i.$$

La fonction de partage se fait comme précédemment. Pour retrouver le secret une fois les n parties de ce secret réunies, on peut procéder de la manière suivante.

On commence par faire une interpolation sur les n parties du secret. Le polynôme ainsi obtenu est égal à $P_{a,s}$ si et seulement s'il est de degré inférieur ou égal à $n-3$: c'est le cas où il n'y aurait pas d'erreur. S'il y a une erreur, on enlève tour à tour une partie du secret, et on fait une interpolation sur les $n-1$ parties restantes. Le polynôme $P_{a,s}$ est alors le seul des polynômes ainsi obtenus qui soit de degré inférieur ou égal à $n-3$.

a) **(sur papier)** Expliquer pourquoi cet algorithme fonctionne.

b) Programmer la fonction `Decode1Err` de décodage correspondante, dans le cas où $K = \mathbb{F}_p$, p étant un nombre premier.

c) Soit s un nombre secret, inférieur à 1000. Soit $p = 1009$. Ce secret est partagé entre 7 personnes. On applique le partage pour $t = [1, \ldots, 7]$. Après transmission du partage, on obtient $[35, 275, 11, 764, 757, 864, 122]$, mais il y a eu une erreur dans la transmission. Calculer s.

8) [Partage avec m erreurs] On s'intéresse maintenant au cas où il y aurait au plus m erreurs, où $n \geqslant 2m+1$. On pose alors $A = K^{n-2m-1}$ et pour $a = (a_1 \ldots, a_{n-2m-1}) \in A$,

$$P_{a,s} = s + \sum_{i=1}^{n-2m-1} a_i y^i.$$

Pour retrouver le secret, on peut généraliser l'algorithme précédent. On essaie comme ci-dessus les cas où il n'y aurait pas d'erreurs, puis le cas où il y en aurait une. Ensuite, on essaie le cas où il y aurait deux erreurs en ôtant tour à tour tous les couples possible de valeurs obtenues, et on continue ainsi, jusqu'à obtenir un polynôme de degré inférieur ou égal à $n - 2m - 1$. Soit s un nombre secret, inférieur à 1000. Soit $p = 1009$. Ce secret est partagé entre 10 personnes. On applique le partage pour $t = [1, \ldots, 10]$. Après transmission du partage, on obtient $[26, 23, 643, 996, 111, 313, 89, 516, 463, 451]$. On suppose qu'il y a au plus trois erreurs. Calculer s en utilisant la méthode ci-dessus.

9) Cette méthode est coûteuse en temps. Le théorème suivant propose une autre méthode, plus efficace.

Théorème 1. *Supposons donnés n couples $(t_i, s_i) \in K^2$, les t_i étant supposés deux à deux distincts. Soit $k = n-2m$. Alors il existe Q_0 et Q_1 dans $K[y]$ tels que $Q_1 \neq 0$, $\deg Q_0 \leqslant n-1-m$, $\deg Q_1 \leqslant n-k-m$ et tels que pour tout i*

$$Q_0(t_i) + s_i Q_1(t_i) = 0.$$

De plus, s'il existe $P \in K[y]$ de degré inférieur ou égal à $k-1$ tel que $P(t_i) = s_i$ pour au moins $n-m$ valeurs de i, alors Q_1 divise Q_0 et $P = -Q_0/Q_1$.

Calculer s de la question précédente en utilisant le théorème 1. On pourra au besoin utiliser les commandes `msolve` et `assign`.

10) (sur papier) Démontrer le théorème 1.