École normale supérieure **2022-2023**
Département d'informatique

Introduction à la cryptologie
TD n° 4 : Correction.

Exercice 1 (Calcul de produit à n joueurs)**.**

1. Soit x la clef secrète d'un chiffrement ElGamal, sur un groupe $\mathbb{G}$ généré par g. Soient (g^r, mg^{xr}) et $(g^{r'}, m'g^{xr'})$ deux chiffrements ElGamal. Alors le produit coordonnée par coordonnée des deux chiffrements est égal à $(g^{r+r'}, mm'g^{x(r+r')})$. C'est un chiffrement valide du message mm'.
2. Les n parties peuvent utiliser le protocole suivant.
   — Les parties génèrent de manière distribuée une paire clef publique/clef secrète ElGamal (voir slides du cours sur la cryptographie distribuée). La clef secrète est donc partagée entre les n parties.
   — Chaque participant i chiffre son message m_i en utilisant la clef publique du chiffrement (qui, comme son nom l'indique, est publique), et publie ce chiffré.
   — Les n participants calculent le produit des n messages chiffrés, puis déchiffrent ce produit de manière distribuée. Du fait que ElGamal est multiplicativement homomorphe, ils obtiennent le produit des m_i.

Exercice 2 (« Five-card trick »)**.**

1. Les deux dernières étapes de mélange du protocole assurent qu'Alice et Bob apprennent l'ordre des cartes modulo une permutation circulaire. Il y a deux cartes coeur et trois cartes trèfles. Si l'on quotiente les permutations de ces cinq cartes par les rotations circulaires, il y a seulement deux classes d'équivalence : une où les deux coeurs sont « consécutifs » (dispositions ♡♡♣♣♣, ♣♡♡♣♣, ♣♣♡♡♣, ♣♣♣♡♡, ♡♣♣♣♡), et une où ils ne le sont pas (dispositions ♡♣♡♣♣, ♣♡♣♡♣, ♣♣♡♣♡, ♡♣♣♡♣, ♣♡♣♣♡). On remarque qu'on est dans la première classe si et seulement si Alice et Bob ont tous les deux dit oui.
2. Si l'un au moins des deux a répondu non, les cartes observées à la fin du protocole sont un des 5 représentants de la seconde classe d'équivalence, tiré uniformément. La disposition des cartes ne contient donc pas d'autre information que le fait qu'un des deux a répondu non. Au passage, on peut remarquer que si Alice par exemple a répondu oui, alors en observant le résultat final elle peut déduire la réponse de Bob. Mais ceci est inévitable : cette information peut être déduite de tout protocole qui réalise la fonctionnalité souhaitée. Cette fuite d'information est intrinsèque à la fonctionnalité demandée. Le but d'un protocole cryptopgrahique multipartite est qu'il n'y ait pas d'autre fuite d'information que cette fuite inévitable.
3. C'est une question piège. Alice ou Bob, connaissant sa propre entrée et le résultat du XOR, peut toujours déduire l'entrée de l'autre participant. On ne peut donc pas être plus sécurisé que le protocole trivial où chaque participant révèle son entrée.

Exercice 3 (Calcul privé d'intersection)**.**

1. Ayant reçu les hachés A d'Alice, Bob peut librement essayer tous les noms qu'il souhaite pour voir s'ils sont dedans, sans se limiter aux noms B qu'il a envoyé plus tôt. Cette solution n'est pas satisfaisante.
2. Les éléments communs entre $\{H(x)^{ab} : x \in B\}$ et $\{H(x)^{ab} : x \in A\}$ correspondent aux éléments x communs entre Alice et Bob. On suppose que les calculs effectués par Alice et Bob préservent l'ordre des éléments, ce qui permet à Alice et Bob de retrouver ces x à partir des $H(x)^{ab}$ correspondants.

3. Soit x tel que $\{x\} = B \setminus A$. Par abus de notation, on appelle ici x la sortie de la fonction de hachage, plutôt que son entrée. On peut donc modéliser x comme un élément uniforme au départ. Au cours du protocole, Alice envoie des valeurs $x_1, \ldots, x_n$ tirées uniformément à Bob, et apprend $x_1^b, \ldots, x_n^b$. Elle apprend aussi x^b. On suppose que tous les éléments considérés sont distincts de l'élément neutre. Alice se demande alors si l'élément x est égal à une valeur $y \neq 1$ quelconque qui l'intéresse. Posons $x = x_1^{a'}$ pour un certain a'. Le point clef est que la question d'Alice, à savoir si $y = x_1^{a'}$, est exactement équivalente à se demander si $(x_1, x_1^b, y, x_1^{a'b})$ est un quadruplet Diffie-Hellman. On a supposé que Diffie-Hellman est difficile dans $\mathbb{G}$, c'est donc une question difficile. Cependant, Alice a des informations supplémentaires : elle connaît aussi les paires (x_i, x_i^b) pour $i > 1$. Un point un peu subtil est que cela ne lui apprend rien de plus : en effet, à partir de (x_1, x_1^b), Alice peut générer des paires (z, z^b) à volonté toute seule, en calculant simplement $(x_1^r, (x_1^b)^r)$ pour r uniforme.
4. En utilisant la propriété d'homomorphisme additif du chiffrement : $E(P(b)\cdot r_b + b) = r_b \sum E(\alpha_i) b^i + E(b)$. Noter qu'on utilise ici une multiplication externe par des éléments r_b et b^i, mais cette opération est bien définie tant que ces valeurs se plongent dans $\mathbb{Z}$: à partir du moment où un chiffrement est additivement homomorphique, il y a une action naturelle de $\mathbb{Z}$ sur les chiffrés, définie par $nE(x) = \sum_{i=1}^{n} E(x)$ et $(-1) \cdot E(x) = E(x)^{-1}$. Dans le cas d'El Gamal tel qu'utilisé dans l'énoncé, la multiplication externe par v revient à transformer le chiffré (c_1, c_2) en (c_1^v, c_2^v).
5. Lorsque l'élément $b \in B$ est aussi dans A, la quantité $E(P(b) \cdot r_b + b)$ renvoyée par Bob est égale à $E(b)$. Alice déchiffre et s'en rend compte en voyant que la valeur obtenue est dans A. Dans le cas contraire, $E(P(b) \cdot r_b + b)$ est le chiffré d'une valeur uniformément aléatoire, qui n'est pas dans A (on suppose que l'espace des chiffrés est exponentiellement grand).
6. Lorsque $b \notin A$, comme on l'a remarqué plus haut, l'élément renvoyé par Bob est le chiffré d'une valeur uniformément aléatoire. Il ne contient donc aucune information sur b. On peut noter que cette garantie est absolue, sans hypothèse calculatoire.

Exercice 4 (Dîner de cryptologues)**.**

1. Il suffit de calculer la somme modulo 2 des valeurs publiées par les dîneurs. L'un d'entre eux a payé ssi cette somme vaut 1. En effet cette somme est égale à la somme des p_i, puisque tous les autres termes apparaissent deux fois.
2. Soit $P = [\![1, n]\!] \setminus K$ l'ensemble des cryptologues qui ne sont pas cooptés par le KGB. Le but est de montrer que le KGB ne peut rien déduire sur les p_i pour $i \in P$, sauf ce qui est impliqué par l'entrée du protocole connue du KGB, à savoir les p_i pour $i \in K$, et sa sortie, à savoir $s = \sum p_i$. Autrement dit, le protocole ne révèle rien de plus au KGB que ce qui est impliqué par son bon fonctionnement.

   Si P est de taille 0 ou 1, le KGB sait déjà tout avec s et $(p_i)_{i\in K}$, donc la question est triviale. De même, si aucun membre de P n'a payé le repas, le KGB le sait à partir de s et $(p_i)_{i\in K}$, et n'a rien non plus à apprendre. Le cas intéressant est celui où $|P| \geq 2$ et un des membres de P a payé le repas. Dans ce cas le KGB voudrait savoir lequel. Soit $i \neq j \in P$, et supposons que le cryptologue i a payé.

   Alors supposons qu'on effectue la chose suivante : on inverse p_i, p_j, et $x_{i,j}$. Par cette transformation, toutes les informations observées par les membres de K sont inchangées. Cependant, c'est j qui a payé et non plus i. On obtient ainsi une bijection entre les choix des $x_{i,j}$ et $(p_i)_{i\in P}$ compatibles avec ce qu'a observé K, et où i a payé, et ceux où j a payé. Comme la distribution des $x_{i,j}$ est par ailleurs uniforme, on déduit que la probabilité que i a payé, ou que j a payé, conditionnée aux observations des membres de K, est identique.
3. Non : si un membre non-KGB est assis entre deux membres du KGB, ceux-ci apprennent son p_i. Si $n > 3$ ça ne devrait pas être le cas.

4. Non, le problème de la question précédente est inévitable : si le graphe n'est pas complet, soit s un sommet d'arité inférieure à $n-1$, alors si les sommets adjacents à s sont contrôlés par le KGB, celui-ci apprend le p_i du sommet s. Or si le KGB ne contrôle pas d'autre sommet, on est dans le cas $|P| \geq 2$, donc le KGB ne devrait pas pouvoir déduire la valeur de ce p_i.