**Projet Crypto**

**Q.2** Nous souhaitons mettre en place un protocole DistanceBob permettant de calculer la distance entre Alice et le bracelet :

Pour cela on suit le protocole suivant :

Une image contenant texte, Police, blanc, reçu

Description générée automatiquement

Ainsi on cherche à retrouver cette distance en récupérant les positions respectives de Alice et Bob sans que Bob ne puisse retrouver les coordonnées de Alice.

*Voir fonction DistanceBob() dans le fichier python associé*

Pour cela, on encrypte les coordonnées de Alice, on encrypte par partie :

en commençant par car on connaît les coordonnées de Bob.

• A l’aide des propriétés d’homomorphie, on encrypte les termes et [en utilisant le produit par constante car les coordonnées de Bob sont connues.

• Propriété homomorphique à nouveau, on encrypte [

• On encrypte [ à l’aide du produit par constante

• On utilise la fonction oppose qui vient du

• On regroupe nos encryptages et on obtient

• A ce moment-là on peut alors décrypter notre résultat, on récupère ainsi la valeur .

Or on remarque que . Ainsi on ajoute au résultat précédent, on prend la racine et l’on obtient notre distance .

**Q.3**

• Confidentialité de la position d'Alice : Le protocole ne révèle pas la position d'Alice à Bob, car seules les coordonnées chiffrées de la position d'Alice ([] et []) sont envoyées à Bob. Tant que le système de chiffrement Paillier utilisé est sécurisé et que Bob ne dispose pas de la clé secrète sk, la position d'Alice reste confidentielle.

• Confidentialité de la position de Bob : Le protocole ne révèle pas directement la position réelle de Bob à Alice. Cependant, Alice peut estimer la distance entre elle-même et le bracelet de Bob en déchiffrant et en effectuant des calculs sur les valeurs chiffrées envoyées par Bob. A l’aide de la clé secrète, elle pourrait ainsi déterminer les coordonnées de Bob.

• Il est possible pour un attaquant de manipuler les valeurs chiffrées échangées entre les deux parties, ce qui peut conduire à des résultats incorrects ou trompeurs pour la distance calculée. Pour garantir l'intégrité des données, il serait nécessaire d'ajouter des mécanismes de vérification d'intégrité (mot de passe…)

**Q.4**

On peut imaginer un protocole différent permettant de calculer si ou non sans qu’elle ne puisse connaître les coordonnées de Bob ni la distance exacte entre les deux.  
Ainsi il faut que Alice envoie ses coordonnées encryptées à Bob et qu’à partir de ces informations Bob puisse calculer directement la distance entre lui et Alice qu’il élèvera au carré. Voici l’idée :

• Alice envoie ses coordonnées encryptées avec la clé publique pk ainsi que ses coordonnées qu’elle élève au carré.

• Bob, avec les informations reçus, peut alors calculer la distance au carré en calculant séparément les différents termes puis on calcule la distance au carré à l’aide des propriétés homomorphiques (cette valeur est appelé bob)

• Afin de ne pas pourvoir décrypter les coordonnées de Bob, on effectue une vérification sur une boucle de 10 000 valeurs en générant un nombre aléatoire entre 1 et 100 que l’on va multiplier par la valeur de l’encryption reçu.

• Si toutes les valeurs sont positives, c’est que la distance est bien supérieure à 100, ainsi Alice ne peut pas connaître les coordonnées de Bob ni la distance entre les deux et inversement.

**Q.5**

• Confidentialité de la position d'Alice : Le cryptosystème de Paillier assure la confidentialité des coordonnées d’Alice en cryptant les valeurs de . Bob ne peut pas récupérer les coordonnées d’Alice sans connaître .

• Confidentialité de la position de Bob : Alice ne peut pas déterminer les coordonnées exactes de Bob car elle ne peut pas connaître la distance exacte qu’il y a entre les deux.

• Il n’est cependant pas garantit que Bob ne puisse pas effectuer d’attaques actives et altérer les données/ envoyer de fausses données. Il est essentiel de mettre en place un protocole afin que Bob se doive de prouver l’encryption de ses coordonnées sans qu’Alice ne puisse pour autant recevoir ces informations.

**Q.6**

*Voir fonction DistanceBob100() du fichier python associé*