Projet

Vous trouverez ici quelques idées/propositions de projet. Il s'agira de proposer, d'analyser et d'implémenter un ou plusieurs protocoles de calcul multi-parties. Toute originalité est la bienvenue. Ce projet donnera lieu à un mini-rapport qui servira de base à un oral d'évaluation.

1. Le protocole Multiplication vu au chapitre **??** n'est pas sécurisé, car à l'étape 2, Bob peut dévier du protocole en envoyant une encryption W de son choix. Le TP 2 permet de résoudre ce problème. Il propose, en effet, une preuve intéractive sans divulgation de connaissance (zero-knowledge proof) permettant à Bob de prouver qu'il ne dévie pas du protocole. Il vous est demandé de réaliser ce TP et de proposer une implémentation sécurisée de Multiplication.

2. Exercice **??**.

3. Après de nombreuses années de mariage cryptographique, Bob a eu un n^{ieme} comportement malveillant vis à vis d'Alice lors d'un protocole houleux. Une mesure d'éloignement à été prononcée contre lui. Pour garantir ceci, il est affublé d'un bracelet numérique connecté (muni d'une petite capacité de calcul) et géolocalisé. Le "bracelet" connait sa position, i.e. ses coordonnées $(x_B, y_B) \in \mathbb{Z}^2$ dans un repère orthonormé arbitraire dont l'unité est le mètre. Les coordonnées d'Alice seront notées $(x_A, y_A) \in \mathbb{Z}^2$

   On supposera dans tout le TP que Bob (via son bracelet electronique) est honnête ou malhonnête passif signifiant qu'il respecte le protocole, e.g. le protocole est implémenté en hard sur son bracelet.

   Nous supposerons dans tout le TP qu'Alice a généré un couple de clés $(pk, sk) \longleftarrow \mathsf{Paillier.KeyGen}(\lambda)$ et que Bob (son bracelet) connait pk. Pour simplifier les notations, $[x]_{pk}$ (ou simplement $[x]$ lorsqu'il n'y aura aucune ambiguité sur la clé publique) désignera une encryption d'une valeur x avec la clé publique pk.

Q1 Il s'agira d'établir un protocole DistanceBob entre Alice et le bracelet, permettant à Alice de connaitre la distance $d_{AB} = \sqrt{(x_B - x_A)^2 + (y_B - y_A)^2}$ à laquelle Bob se trouve. Voici une version non-détaillée de ce protocole qu'il s'agira de détailler.

DistanceBob

(a) **Alice** envoie $[x_A]$ et $[x_B]$ à Bob

(b) **Bob** Envoie $[x_B^2 + y_B^2 - 2(x_A x_B + y_A y_B)]$ à Alice

(c) **Alice** retourne d_{AB}

Q2 Implémenter ce protocole.

Q3 Analyser la sécurité de ce protocole.

Q4 Proposer une variante DistanceBob100 du protocole précédent permettant à Alice de connaitre seulement si $d_{AB} \leq 100$ ou non (et non plus d_{AB}).

Q5 Analyser la sécurité de DistanceBob100.

Q6 Implémenter DistanceBob100.

Q7 Proposer (sans l'implémenter)/analyser une variante LocalisationBob100 permettant à Alice de connaitre (x_B, y_B) si et seulement si $d_{AB} \leq 100$.

Indice. On pourra utiliser le fait que $100/n$ est négligeable, i.e. $100/n \leq 2^{-\lambda}$.