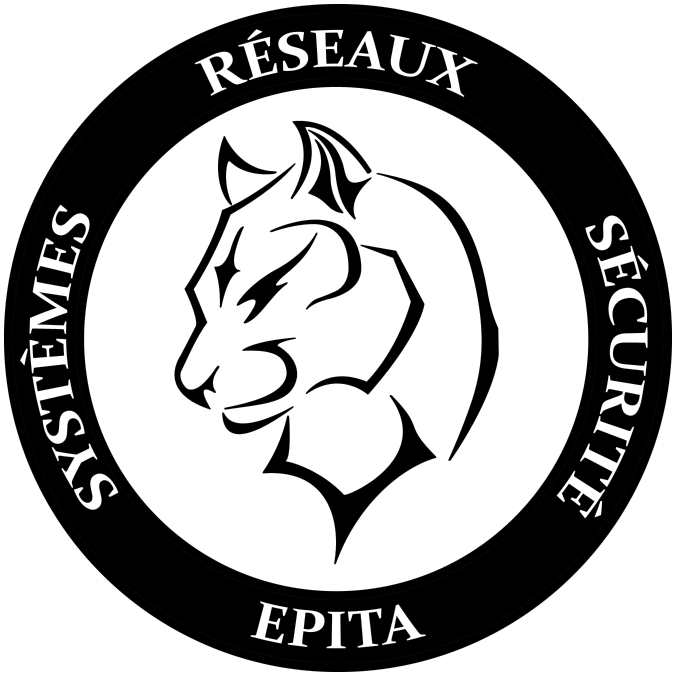
CRYPI

Rapport de Projet



Valentin LUCOTTE
Hector Colin
Léo Blanc-Di-Pasquale
Oscar Mrad

Sujet:
Réaliser une authentification par reconnaissance faciale en garantissant la confidentialité des données biométriques entre le client et le serveur.

Contents

1 Introduction

Le sujet de ce rapport correspond à de l'authentification par des données biométriques. L'enjeu de ce sujet est la confidentialité des données biométriques de l'utilisateur et du serveur :

De quelle manière pouvons nous réaliser de la reconnaissance faciale sans qu'aucun parti ne puisse récupérer les données de l'autre ?

La solution à cette question est la réalisation d'un algorithme de reconnaissance faciale mettant en place un protocole Secure Multiparty Computation (SMC), qui permet de garantir cette confidentialité.

Nous allons donc présenter le protocole utilisé et comment nous allons l'utiliser. Nous avons deux acteurs communiquant :

- Un client, semi-honnête, qui doit s'authentifier à l'aide d'une image.
- Un serveur, semi-honnête, qui authentifie les clients et possède une base de données d'images.

Le message de passe pas par un tier parti. Cependant, un man in the middle peut être pris en compte.

2 Présentation de SMC

2.1 Introduction

Les premiers protocoles SMC sont apparus à la fin des années 1970. En 1982, le calcul sécurisé a été formellement introduit sous la forme de calcul sécurisé à deux parties (2PC) par Andrew Yao, notamment pour le problème des millionnaires. En 1986, Yao a généralisé cette approche pour permettre tout calcul faisable. Le *Secure Multi-party Computation* ou SMC est une branche de la cryptographie qui permet à plusieurs entités de pouvoir calculer une fonction sur leurs entrées privées sans avoir à les dévoiler.

2.2 Objectifs

Les objectifs du calcul SMC sont centrés sur la confidentialité, la sécurité et l'intégrité des données partagées entre plusieurs parties. Plus particulièrement:

- La confidentialité des données: il faut s'assurer que les données des entités qui participent à la procédure de chiffrement ne fuitent pas.
- La sécurité contre les adversaires: il faut se protéger des personnes qui voudrait porter atteinte aux entreprises :
  - adversaires semi-honnêtes (honest-but-curious): Ces adversaires suivent le protocole correctement mais essaient de tirer des informations des messages reçus ;
  - adversaires malveillants (malicious): Ces adversaires peuvent dévier du protocole de manière arbitraire pour essayer de compromettre la confidentialité ou l'exactitude des données.

2.3 Fonctionnement

Le protocole SMC a deux modes de fonctionnement. Le premier est un mode où seulement deux parties sont utilisées pour assurer la sécurité des données. Le deuxième mode est un mode multipartite où plusieurs parties peuvent être utilisées.

2.3.1 Calcul bipartite

Le 2PC permet à deux parties de calculer une fonction sur leurs entrées sans révéler ces entrées l'une à l'autre. Ce type de protocole est particulièrement

utile pour des applications où seulement deux parties interagissent. Méthode :

- Préparation du circuit : Le circuit est converti en un circuit "brouillé" où les données des différents partis sont encodées.
- Évaluation : Le second client évalue le circuit brouillé sans apprendre les entrées spécifiques, obtenant ainsi les étiquettes correspondant aux données privées.
- Décryptage : Le récepteur envoie les étiquettes de sortie au créateur du circuit, qui les traduit en résultats compréhensibles.

Avantages : Étant donnée que le nombre de partis est de 2 l'efficacité et le nombre de tours est maximal par rapport à des adversaires semi-honnêtes.

2.3.2 Protocoles multipartites

Le MPC généralise le SMC à plus de deux parties. Il permet à plusieurs acteurs de calculer conjointement une fonction sur leurs entrées sans les révéler entre eux. Méthode :

- Partage de Secret : Les données sont divisées en parts distribuées parmi les participants.
- Évaluation de Circuit Arithmétique : Les fonctions sont définies comme des circuits arithmétiques, avec des portes d'addition et de multiplication.
- Reconstruction : Les résultats sont combinés pour obtenir le résultat final sans révéler les parts individuelles.

2.4 Applications

Le SMC est crucial pour les situations où plusieurs parties souhaitent collaborer sur des données sensibles sans compromettre la confidentialité. Il permet de réaliser des calculs sécurisés dans des domaines tels que :

- finance sécurisée: Calcul de statistiques financières sans divulguer les données individuelles ;
- recherche médicale: Analyse de données médicales sans révéler les informations des patients ;
- enchères électroniques: Permettre des enchères sécurisées où les soumissions restent privées.

3 Description détaillée du protocole SMC choisi

3.1 Protocole

Le protocole que nous avons choisi est le Garbled Circuit Protocol. Il s'agit d'un protocole à deux parties créé par Andrew Yao qui consiste en ces étapes :

- On créé un circuit permettant de réaliser l'opération que nous souhaitons.
- Alice chiffre son input et le circuit avec sa clé, et les transmet à Bob.
- Bob chiffre son input et actionne le circuit. Il renvoit le résultat à Alice.
- Alice déchiffre le résultat.
- Les deux communiquent leurs résultats.

Alice ne connait que le résultat de l'opération, et Bob ne peut pas déchiffrer la donnée envoyé par Alice, ni le résultat de l'opération.

3.2 Utilisation

Dans notre cas, Alice est le client, Bob est le serveur. Le circuit est le calcul de la distance euclidienne entre les vecteurs faciales des images. Le client commence par générer une paire de clé de chiffrement, une publique et une privée. Il extrait les vecteurs de son image et les chiffre avec sa clé publique. Il envoi alors son nom, les données chiffrées et la clé publique au serveur. Le serveur extrait les images correspondant au nom récupéré, les vectorise et chiffre les résultats. Il calcule alors la distance euclidienne entre ces vecteurs et ceux fournis et transmet le résultat au client. Celui ci déchiffre le résultat et le communique au serveur. A aucun moment un parti peut déchiffrer les images ou les vecteurs issus de ces images de l'autre parti.

Nous avons mis tout cela en pratique dans un code Python à l'aide des différentes librairies que nous utilisons ci dessous.

4 Critères et processus de sélection de la bibliothèque SMC

4.1 MPyC

MPyC est une librairie qui permet de mettre en place un Multi Party Computation entre deux ou plusieurs utilisateurs distants. Il permet de sécuriser les données qui vont être transmis, les transmettre, et les traiter en réalisant des opérations programmé par avance. Il s'agit d'une librairie reconnu pour réaliser des SMC et est notre choix final pour faire l'authentification. Le chiffrement des données est homomorphe ce qui permet de réaliser n'importe quel type d'opération.

4.2 Face recognition

Face recognition est une libraire basé sur dlib qui s'occupe de réaliser simple ce que dlib peut réaliser plus laborieusement. Il peut extraire des visages au sein d'une photo, directement comparer des visages ou bien extraire des vecteurs de visages. Son taux de succès sur la base de donné LFW est de 99.38%.

4.3 Non retenu

4.3.1 PyFhel

Pyfhel est une bibliothèque qui nous fournit un moyen de chiffrer les données tout en nous permettant de réaliser les opérations du garbled circuit. Elle permet de faire un chiffrement homomorphe complet, c'est-à-dire qu'elle permet de calculer sur des données chiffrées. Complet signifie qu'elle peut effectuer aussi bien des additions et des soustractions que des multiplications et des divisions.
C'est un critère nécessaire étant donné que les calculs pour récupérer la distance euclidienne nécessitent des multiplications. Elle ne peut donc pas se contenter d'un chiffrement partiellement homomorphe ne gérant que les additions et les soustractions.
Après avoir trouvé la bibliothèque MPyC, il a été décidé de mettre cette solution de côté pour se baser sur une solution déjà reconnue comme viable plutôt que de nous appuyer sur un procédé que nous avons réalisé nous-mêmes. Le prototype est cependant toujours disponible dans le code source.

5 Résultats des expériences menées

Nous avons deux types d'authentification :

- Un utilisant seulement l'image.
- Un avec nom et image.

5.1 Image uniquement

Le calcul est très lent du fait que nous parcourons toute la base de données à chaque authentification. Réaliste, ce n'est pas la méthode idéale pour s'identifier ; cependant, cela peut être appliqué dans le cas de la reconnaissance faciale, lorsque le client n'est pas une personne mais une caméra de surveillance, une porte électronique à reconnaissance faciale ou un portillon automatique dans un aéroport.

Sur un échantillon de 50 photos différentes, comprenant des photos de la même personne et de différentes personnes, nous avons obtenu 100% de réussite dans la comparaison des images. Cependant, nous nous rendons bien compte que sur un échantillon beaucoup plus grand, il devrait y avoir des faux positifs et des faux négatifs. En effet, nous avons pu observer que des photos très semblables d'une même personne ont une distance euclidienne de 0,3, ce qui indique clairement la ressemblance de l'image. Cependant, en moyenne, deux images différentes d'une même personne auront une distance proche de 0,5, tandis que deux images de personnes différentes auront une distance de 0,75. L'écart étant relativement petit, il ne serait pas surprenant de voir une distance euclidienne passer en dessous de 0,6 si deux personnes se ressemblent beaucoup, ou au-dessus si la qualité de deux photos d'une même personne est très différente. Certaines personnes dans la base de données ne possèdent qu'une unique photo, ce qui ne permet pas de corriger ces éventuelles erreurs.

5.2 Nom et image

Plus rapide, le serveur va vérifier dans la base de données les images d'un nom donné, et comparera tout cela avec les vecteurs de la photo d'un client. Nous parlons d'un temps d'exécution d'environ 10 secondes. C'est un cas d'authentification bien plus probable pour des applications par exemple. Le taux de succès est le même que précédemment sur l'échantillon que nous avons, et les mêmes problématiques s'imposent.

6 Défis rencontrés pendant le projet

Étant donné que nous avons 2 parties qui veulent communiquer entre-eux, nous pensions que le *Garbled Circuit Protocole* serait la meilleure option. Nous pensons que c'est toujours la meilleure solution, mais pas à implémenter. En effet, lors de nos recherches nous avons trouvé plusieurs bibliothèques pour implémenter le *Garble Circuit Protocole* mais il fallait s'occuper de la partie communications entre le client et le serveur. Nous sommes donc partis sur le MPC (avec la bibliothèque MPyC).

6.1 Numpy

Lors de nos tests nous avons utilisé la version la plus récente de numpy qui est la 2.0. Or, une mise à jour majeure à eu lieu du passage à la 2.0 et la bibliothèque que nous avons utiliser au début ne fonctionnait pas. Nous sommes donc passé à la version 1.24 (la dernière avant la 2.0) et le problème a été résolus.

6.2 Reconnaissance faciale

Nous voulions utiliser une librairie de reconnaissance d'image simple, face_recognition, afin de ne pas nous embêter avec cette aspect là de l'application. Malheureusement elle n'est pas compatible avec les données chiffrées. C'est pour cela que nous avons donc dû comparer à la main les vecteurs extraits des visages en calculant leur distance euclidienne.

6.3 Chiffrements

Afin d'effectuer des opérations sur des données chiffrées nous avions besoin d'un type spécifique de chiffrement que l'on nomme homomorphe. Qui peut donc réaliser des opérations de type additions ou multiplications et que nous puissions retrouver le résultat attendu en le déchiffrant.
Au début nous pensions utiliser la méthode de chiffrement de Paillier, mais il se trouve qu'elle n'est pas entièrement homomorphe, elle ne peut que réaliser des additions, et ne peut pas être multiplié par autre chose que des scalaires. Nous avons cependant trouvé plusieurs librairies correctes pouvant le faire, Pyfhel par exemple, et MPyC peut aussi l'intégrer.

7 Étapes futures recommandées pour améliorer le projet

7.1 Base de donnée

L'une des améliorations que nous pourrions faire est la sécurisation de la base de donnée. En effet pour le moment nous nous basons sur le fait que les deux parties soit à moitié bienveillant. Mais le serveur peut être piraté, et il est important de trouver une solution pour que les images soient sécurisées. Si nous rajoutions un tier de confiance nous pourrions faire en sorte que le serveur n'ait à aucun moment accès aux images en claire, et seul le tier de confiance pourrait avoir la clé privée.

7.2 Déploiement à grande échelle

Nous pourrions déployer à grand échelle. Dans le cas actuel le serveur ne pourrait communiquer qu'avec un client à la fois, mais si nous changions la méthode de communications et effectuons de la parallélisation des tâches nous pourrions avoir un serveur pouvant subvenir à cette tache.

7.3 Protocole SMC

Il était possible de réaliser un protocole plus sûr en utilisant plusieurs parties, mais nous devant nous contenter de seulement deux partis, nous n'avions pas eu d'autres choix que de nous servir aussi du chiffrement homomorphe, obligatoire pour réaliser des opérations en gardant toute confidentialité.