Secure identification for the Internet of Things

Marzieh Gheisari

DOCTORAT BRETAGNE LOIRE / MATHSTIC

UNIVERSITÉ DE RENNES 1

THÈSE DE DOCTORAT DE

L'UNIVERSITÉ DE RENNES 1

ÉCOLE DOCTORALE N° 601
Mathématiques et Sciences et Technologies de l'Information et de la Communication
Spécialité : *Signal, Image, Vision*

Par

Marzieh Gheisari

Secure identification for the Internet of Things

Thèse présentée et soutenue à Rennes, le 7 Juillet 2021
Unité de recherche : Inria Rennes - Bretagne Atlantique

Rapporteurs avant soutenance :

Marta Gomez Barrero	Professor, Hochschule Ansbach, Germany
Patrick Bas	Senior Researcher, CNRS, Lille

Composition du Jury :

Examinateurs :	Valerie Viet Triem Tong	Professor, CentraleSupelec, Rennes
	Boris Škorić	Associate Professor, Eindhoven University of Technology, Netherlands
	Zekeriya Erkin	Associate Professor, Delft University of Technology, Netherlands
Dir. de thèse :	Laurent Amsaleg	Senior Researcher, CNRS, Rennes
Co-dir. de thèse :	Teddy Furon	Researcher, Inria, Rennes

Acknowledgement

I am deeply grateful to Dr. **Laurent Amsaleg** for being extremely supportive and for all the help in writing this thesis. I would like to take this opportunity to express my gratitude for his guidance and his profound understanding of everything, and his kind consideration. Thanks, Laurent, for making my Ph.D. easier.

My special thanks go to Dr. **Teddy Furon**; without his help, I would not be able to come up with this outcome. I am grateful to him for all his efforts and ideas that helped me throughout my doctoral study. Thank you, Teddy, for guiding me with a great deal of patience. It was a pleasure working with you.

I appreciate the members of my doctoral committee, the reviewers Dr. **Marta Gomez-Barrero** and Dr. **Patrick Bas**, and also the examiners, Dr. **Valerie Viet Triem Tong**, Dr. **Boris Skoric** and Dr. **Zekeriya Erkin**, for their precious time, shared positive insights and guidance.

I would like to thank all the people that I met in the Linkmedia team for all the discussions and fun times we had over the past three years. Special thanks to my colleague Hanwei Zhang for her beautiful friendship. Dear Hanwei, the days when we sat together and listened to each other's stories will be in my mind forever. I also would like to give thanks to Behrooz, whom I collaborated with on the project, for all the scientific and non-scientific conversations and his kind comments.

There are no words to express my deep appreciation towards my mother, Fatemeh, and my father, Nadali, to whom I owe a lot. They have always supported me throughout my life while leaving me free to make all my decisions. Thank you for all your sacrifices and unconditional support. My gratitude also extends to my sisters, Mehri and Maryam, and to my brothers, Mehdi and Mohammad, who have always been there, and I can turn to for advice. Finally, I have to thank my inspiration, my beloved husband, Javad. We have been so lucky to do our Ph.D. at the same time and in the same lab. It was very exciting to go through this journey together. You were giving me motivation every day and stood by me through the good times and bad. I am looking forward to the next chapter of my life with you.

Résumé en français

Motivations

L'internet des objets (IoT) est l'une des principales tendances technologiques qui ont émergé ces dernières années. L'internet des objets désigne le réseau de milliards de dispositifs physiques dotés de capteurs et de logiciels. Ces dispositifs vont des objets domestiques courants, dont les ressources de traitement et de communication sont très faibles, voire inexistantes, aux outils industriels dotés de ressources sophistiquées.

En raison de cette connectivité, des défis tels que la sécurité et la confidentialité des données des utilisateurs sont apparus. Les dispositifs IoT doivent communiquer entre eux pour échanger les données qu'ils ont collectées ou reçues d'autres dispositifs et réagir aux informations reçues. Les accès non autorisés aux réseaux IoT, telles que les attaques par usurpation d'identité, sont des problèmes critiques dans ces systèmes. Par conséquent, l'une des principales préoccupations en matière de sécurité dans l'utilisation d'objets connectés dans la vie quotidienne est l'authentification, comme la vérification de l'authenticité d'un dispositif ou l'identification d'un dispositif pour confirmer que les informations sont reçues d'un utilisateur authentifié et envoyées à celui-ci.

Ces protocoles peuvent être confrontés à des menaces provenant d'un serveur honnête mais curieux ou d'un adversaire malveillant. Un adversaire honnête mais curieux essaie d'apprendre les données privées de l'autre partie. Un adversaire malveillant manipule le protocole pour connaître les données privées de l'autre partie ou générer de faux résultats. Par conséquent, pour des raisons de sécurité, l'authentification doit se dérouler de manière à ce que les parties non fiables ne puissent déduire aucune informations auxquelles elles n'ont pas le droit d'accéder. Pour des raisons de confidentialité, le cloud doit authentifier les dispositifs IoT sans révéler leur identité exacte.

Les algorithmes de sécurité traditionnels pour l'authentification sont basés sur le stockage d'informations secrètes (par exemple, une clé cryptographique) dans des mémoires

non volatiles, qui risquent d'être copiées, volées ou détruites. Une fonction non clonable physiquement (PUF) est une primitive de sécurité qui utilise les propriétés physiques d'un dispositif pour générer une empreinte numérique. Par conséquent, une PUF est unique à son dispositif, ce qui les rend idéalement adaptées aux réseaux IoT. Elles permettent l'authentification et l'identification d'objets physiques sans nécessiter de stockage d'informations secrètes.

Dans l'architecture basée sur les PUF pour l'authentification des dispositifs IoT, pendant l'inscription, le serveur envoie d'abord au client une requête appelée défi, puis enregistre la réponse du dispositif. Ces paires défi-réponse seront stockées dans une mémoire sécurisée pour permettre l'autorisation du dispositif (par exemple, dans les configurations PUF optiques, un défi consiste en un point spécifique et les propriétés du faisceau laser entrant. La réponse correspondante de la PUF est le motif formé par les tavelures sur l'illumination brute capturée par la caméra CCD). Au moment de l'authentification, le serveur envoie la même requête à la PUF. Sa réponse au défi est envoyée au serveur, qui la compare au modèle de réponse stocké dans la mémoire sécurisée, et l'authentification est accordée si le taux de correspondance est supérieur à un certain seuil.

Les solutions de pointe en matière de sécurité et de confidentialité qui existent pour la biométrie ne peuvent pas être adaptées à l'IoT. Par exemple, le "Traitement du signal dans le domaine crypté" utilise une cryptographie avancée comme le cryptage entièrement homomorphique. Le traitement des données cryptées est beaucoup plus lent et nécessite une utilisation plus importante de la bande passante par rapport au traitement des données en clair. Compte tenu des caractéristiques cruciales des réseaux de l'IoT, telles que le nombre massif de nœuds et les ressources de communication et de calcul limitées des nœuds, ces techniques de sécurité actuelles ne sont pas efficaces pour l'IoT. Par conséquent, le principal défi consiste à développer des protocoles préservant la vie privée et présentant un bon comportement à l'échelle.

Le traitement des signaux non cryptographiques peut répondre aux exigences de vitesse. Par exemple, les algorithmes de recherche ANN (Approximate Nearest Neighbors) peuvent trouver l'élément le plus similaire à une requête dans de grandes bases de données avec une capacité à passer à l'échelle remarquable. Les performances des ANN seraient bien adaptées à une utilisation dans l'IoT. Cependant, ces algorithmes ne

disposent pas des fonctionnalités de sécurité et de confidentialité requises et, à notre connaissance, ils n'ont pas encore été appliqués aux données PUF.

Cette thèse introduit une nouvelle fonctionnalité liée à l'authentification et l'identification : la vérification de l'appartenance à un groupe. L'objectif est de vérifier si un dispositif IoT donné est membre d'un groupe prédéfini de dispositifs connus. Un groupe peut être défini comme regroupant des dispositifs d'un type particulier comme les équipements médicaux dans un environnement hospitalier, notamment les scanners CT et IRM, les équipements à ultrasons et les équipements à rayons X. En effet, cette fonctionnalité renforce la confidentialité, de sorte que la vérification d'un appareil dans un groupe d'appareils se déroule sans que son identité soit divulguée au cloud.

La vérification de l'appartenance à un groupe est différente des concepts de signature de groupe ou de signature en anneau en cryptographie, dans lesquels tout membre du groupe peut signer anonymement des messages sans révéler son identité. Cette capacité équivaut à prouver que l'on est membre du groupe. Cependant, il n'y a pas d'enrôlement de gabarits biométriques ou PUF puisque l'adhésion est équivalente à la détention d'une des clés valides. En outre, ces protocoles ne sont pas compatibles avec l'IoT car la cryptographie lourde n'est pas adaptée à ces applications. Par conséquent, cette thèse aborde les protocoles efficaces d'authentification, d'identification et de vérification de l'appartenance à un groupe qui préservent la vie privée et qui sont appropriés pour les applications biométriques et IoT.

Contexte

La vérification de l'appartenance d'un objet, d'un dispositif ou d'un individu à un groupe est une tâche naturelle qui constitue la base de nombreuses applications accordant ou refusant l'accès à des ressources sensibles (bâtiments, wifi, paiement, ...). L'appartenance à un groupe peut être mise en œuvre par un processus en deux phases : une *identification* est d'abord effectuée, révélant l'identité de l'individu examiné, suivie d'une *vérification* où l'on vérifie si l'individu identifié est bien membre du groupe revendiqué. Cette mise en œuvre porte atteinte à la vie privée : il n'y a aucune raison *fondamentale* d'identifier l'individu avant d'exécuter l'étape de vérification. C'est plus facile mais pas vraiment nécessaire. Il est fondamental de distinguer les membres du groupe des non-membres,

mais il n'est pas nécessaire de distinguer les membres les uns des autres.

Les mêmes remarques valent pour l'identification de groupes. Un tel système gère plusieurs groupes, par exemple en séparant les individus selon l'équipe dans laquelle ils travaillent. Le but est alors d'identifier le groupe précis auquel appartient un membre, sans procéder d'abord à l'identification de l'individu.

Sans perte de généralité, la vérification de l'appartenance à un groupe nécessite d'abord d'acquérir des motifs d'objets (PUF) ou d'individus (trait biométrique) et de les inscrire dans une structure de données stockée dans un serveur. Ensuite, au moment de la vérification, cette structure de données est interrogée par un client avec un nouveau motif, et l'accès est accordé ou refusé. La sécurité évalue que la structure de données est suffisamment protégée pour qu'un serveur honnête mais curieux ne puisse pas reconstruire les motifs. La protection de la vie privée exige que la vérification se fasse sans révéler l'identité.

La nature des motifs peut varier d'une application à l'autre. Par exemple, les motifs codent des informations relatives aux empreintes digitales, à l'iris ou aux visages des individus, ou des PUF comme les tavelures capturées à partir d'un plastique transparent éclairé au laser (voir le chapitre 4).

Il convient de souligner deux propriétés fondamentales des motifs. Le modèle utilisé au moment de la vérification est une version bruyante de celui acquis au moment de l'inscription. Les conditions d'éclairage, la pression sanguine, le vieillissement, l'usure, les conditions physiques transitoires sont des facteurs possibles qui peuvent causer des variations au moment de l'acquisition. Le protocole de vérification doit absorber ces variations et faire face à la nature continue des motifs. Cependant, il est très peu probable qu'une version bruyante du gabarit correspondant à un membre du groupe soit suffisamment similaire au gabarit enregistré d'un autre membre du groupe. La première propriété est donc en rapport avec la nature continue et distinguable des motifs. La deuxième propriété concerne l'indépendance statistique des gabarits enregistrés.

Le scénario opérationnel traditionnel considère un serveur qui exécute la vérification de l'appartenance à un groupe. Le serveur reçoit les requêtes des clients. Un client

acquiert un nouveau modèle et interroge ensuite le serveur. Les clients sont de confiance. Le serveur est honnête mais curieux : il peut essayer de reconstruire les motifs inscrits ou espionner les requêtes. La conception vise à empêcher le serveur de reconstruire le modèle privé du système tout en déterminant correctement si un utilisateur est ou non membre du groupe revendiqué (vérification du groupe) ou en identifiant le groupe d'appartenance (identification du groupe).

Contributions

Dans cette section, nous énumérons les principales contributions apportées dans cette thèse.

1- Le chapitre 2 propose un protocole de vérification de l'appartenance à un groupe grâce à l'utilisation conjointe de deux mécanismes. Le premier consiste à quantifier les motifs au travers de plongements discrets, ce qui limite la capacité du serveur à reconstruire les motifs. L'autre consiste à regrouper les motifs dans une représentation de groupe, ce qui empêche un serveur de déduire une signature spécifique à partir de cette valeur. Des informations suffisantes doivent être conservées via le processus d'agrégation pour que le serveur puisse affirmer si une signature de requête est ou non membre du groupe.

Tout d'abord, nous considérons deux blocs de construction indépendants, l'un pour les plongements, l'autre pour l'agrégation. Ces deux blocs peuvent être assemblés selon deux configurations: bloc #1 avant bloc #2, le système acquiert puis hache les signatures avant de les agréger. La configuration opposée est celle où les signatures acquises sont agrégées avant de hacher le résultat de cette agrégation. L'assemblage des blocs et les stratégies d'agrégation créent globalement quatre variantes.

Nous considérons un modèle statistique simple pour des motifs de dimension d et aussi un modèle du motif de requête (par exemple, la corrélation entre les vecteurs pré-enregistrés et les vecteurs de requête est supérieure à $c > 0$). En supposant que nous regroupions les descripteurs n dans un seul groupe, nous analysons les performances, la sécurité et la confidentialité en fonction des paramètres d, c et n.

Après cela, nous remplaçons ces fonctions passives par des fonctions produisant les

mêmes types de sortie, mais leurs paramètres sont appris grâce à l'optimisation. Pour les deux constructions, c'est-à-dire les deux manières d'assembler les blocs, ceci est réalisé en minimisant une fonction objective additionnant un coût pour la phase de plongements et un coût pour l'agrégation.

Enfin, plutôt que de considérer les affectations de groupe qui sont prédéterminées, les affectations de groupe sont également apprises en même temps que les représentations des groupes. L'idée est de minimiser la distance globale entre les membres du groupe tout en maximisant la séparation entre les groupes dans le domaine où se réalisent les plongements. Deux scénarios d'application sont étudiés : la vérification de groupe et l'identification de groupe. Nous montrons les améliorations à travers une vaste série d'expériences ciblant la reconnaissance faciale.

2- Dans le chapitre 3, nous analysons un modèle mathématique pour la vérification de l'appartenance à un groupe. Ce schéma étudie l'impact du degré de parcimonie des caractéristiques en haute dimension représentant les membres du groupe sur la qualité des correspondances (véritablement positives) et leur robustesse au bruit.

On suppose que les séquences suivent un modèle statistique donnant un rôle central au symbole 0. Nous avons une probabilité différente pour le symbole 0, tandis que les autres ont une probabilité égale. Ensuite, nous considérons deux configurations : "parcimonieux" et "dense", qui font référence au nombre d'éléments nuls et non nuls dans une séquence. Lorsqu'une séquence est très parcimonieuse, elle contient principalement des zéros et quelques éléments non nuls, tandis que les séquences dans des environnements denses contiennent principalement des éléments non nuls.

Pour calculer la représentation de groupe, on calcule d'abord le type (c'est-à-dire l'histogramme) des symboles n. Étant donné que la cardinalité de l'ensemble des valeurs de type possibles peut être trop grande, on applique une fonction surjective sur les valeurs de type. Nous modélisons également la source de bruit par un canal de communication discret.

Nous présentons trois grandeurs en lien avec la théorie de l'information pour mesurer les performances du schéma. La compacité et la sécurité dépendent du modèle statistique

des séquences et du mécanisme d'agrégation. Les performances de vérification dépendent en outre du canal. Ensuite, nous modélisons le compromis entre sécurité, compacité et performances de vérification avec ces outils théoriques.

Nous appliquons ce point de vue aux séquences aléatoires binaires et ternaires et montrons que le bruit sur la requête a un impact en lien avec la rareté des séquences. En termes de performances de vérification, la configuration parcimonieuse peut être optimale lorsqu'il n'y a pas de bruit, mais l'erreur ne sera jamais égale à zéro en pratique. Cela signifie qu'il n'est pas facile d'avoir une solution réellement parcimonieuse. De plus, lorsque les requêtes positives sont moins corrélées avec les motifs préenregistrés, la configuration dense est plus intéressante en termes de performances de vérification et de niveau de sécurité.

3- Dans le chapitre 4, nous étudions les données expérimentales des fonctions optiques physiques non clonables fournies par notre partenaire de projet de l'Université d'Eindhoven pour concevoir des schémas qui permettront l'authentification et la vérification de l'appartenance au groupe de données de type PUF.

En considérant les valeurs de pixel comme des vecteurs de caractéristiques, nous explorons certaines propriétés des données PUF en utilisant les caractéristiques intra-, inter-PUF et inter-Challenge-distance. Étant donné une image tavelée, il n'est pas possible de détecter de quel PUF l'image provient. De plus, pour un PUF fixe, les corrélations entre toutes les réponses ne sont pas significatives. Par conséquent, il en bon d'utiliser des PUF optiques de manière passive et d'avoir une étape de description extrayant des caractéristiques plus distinctes.

Comme les systèmes d'authentification biométrique, nous considérons une paire de PUF et un défi en tant qu'individu. Nous avons également repensé un réseau siamois pour produire de courts descripteurs pour les données de mesure PUF. L'objectif est de s'assurer que deux images avec la même étiquette ont leur plongements rapprochés dans l'espace de représentation tandis que deux entrées avec des étiquettes différentes sont loin. Nous apprenons des descripteurs tels que l'inter-PUF et l'inter-Challenge-distance seront grands, et l'intra-distance sera petit, ce qui permet une authentification fiable des individus.

Ensuite, il est démontré qu'après l'apprentissage des descripteurs, le chevauchement entre les distributions intra et inter-distance a été réduit de manière significative. Nous montrons également l'efficacité des descripteurs appris pour les scénarios d'authentification PUF passive. Nous développons également un schéma de vérification de l'appartenance à un groupe qui vérifie si la requête PUF correspond à l'un des PUF précédemment inscrits d'un groupe donné. Le principal défi consiste à trouver un compromis entre les performances, la sécurité et la confidentialité. Les propriétés statistiques et le contenu informatif des descripteurs conçus auront un impact significatif sur ce compromis.

Publications

Cette thèse s'appuie sur les résultats précédemment publiés dans les ouvrages suivants.

- Marzieh Gheisari, Teddy Furon, Laurent Amsaleg, Behrooz Razeghi, and Slava Voloshynovskiy. "**Aggregation and embedding for group membership verification**." In IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2019.
- Marzieh Gheisari, Teddy Furon, and Laurent Amsaleg. "**Privacy preserving group membership verification and identification**." In IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2019.
- Marzieh Gheisari, Teddy Furon, and Laurent Amsaleg. "**Group membership verification with privacy: Sparse or dense?**." In IEEE International Workshop on Information Forensics and Security (WIFS), 2019.
- Marzieh Gheisari, Teddy Furon, and Laurent Amsaleg. "**Joint Learning of Assignment and Representation for Biometric Group Membership**." In IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2020.

Table of Contents

INTRODUCTION

Motivations

The Internet of Things (IoT) is one of the most leading technology trends that have emerged in recent years. The Internet of Things refers to the network of billions of physical devices that are embedded with sensors and software. These devices range from common household objects which have very low or nonexistent processing and communication resources to industrial tools with sophisticated resources.

Due to this connectivity, challenges like security and privacy of user data have arisen. The IoT devices need to communicate with each other to exchange the data they collected or received from other devices and react to the received information. Unauthorized access to the IoT networks, such as impersonating attacks, are critical concerns in these systems. Therefore, one of the most security concerns in the use of connecting objects in everyday people's life is authentication, such as verifying if a device is authentic or identifying a device to confirm that the information is received from and sent to an authenticated user.

These protocols might face threats coming from an honest-but-curious server or a malicious adversary. An honest-but-curious adversary tries to learn the private data of the other party. A malicious adversary manipulates the protocol to learn the other party's private data or generate false output. Therefore, for security reasons, the authentication should proceed so that untrusted parties cannot infer any information they are not allowed to. For privacy reasons, the Cloud needs to authenticate IoT devices without revealing their exact identity.

Traditional security algorithms for authentication are based on storing secret information (e.g., cryptographic key) in Non-Volatile Memories, which is at risk of copying, stealing, and destruction. A physically unclonable function (PUF) is a security primitive that utilizes the physical properties of a device to generate a fingerprint. Therefore, a PUF is unique to its device, making them ideally suited for IoT networks. They enable

authentication and identification of physical objects without requiring any storage of secret information.

In PUF-based architecture for authentication of IoT devices, during the enrollment, the server first sends the client a query called a challenge and then records the response of the device. These Challenge-Response pairs will be stored in a secured memory to allow authorization of the device (e.g., in optical PUF configurations, a challenge consists of a specific point and the incoming laser beam's properties. The corresponding response of the PUF is the raw speckle pattern captured by the CCD camera.). At authentication time, the server sends the same query to the PUF. Its response to the challenge is sent to the server, which is compared against the response template stored in the secure memory, and authentication is granted if the rate of matching is above a certain threshold.

The state-of-the-art security and privacy solutions existing for biometrics cannot be adapted to IoT. For example, "Signal Processing in the Encrypted Domain" uses advanced cryptography like fully Homomorphic encryption. Processing encrypted data is much slower and requires higher bandwidth usage compared to processing data in the clear. While regarding the crucial characteristics of IoT networks, such as the massive number of nodes and the limited communication, and computation resources of the nodes, these current security techniques are not efficient for IoT. Therefore the key challenge is developing privacy-preserving protocols with good scaling behavior.

Non-cryptographic signal processing can handle the speed requirements. For instance, Approximate Nearest Neighbors (ANN) search algorithms can find the most similar item to a query in large databases with remarkable scalability. The ANN's performance would be well suited for use in the IoT. However, they do not have the required security and privacy functionality, and to the best of our knowledge, they have not yet been applied to PUF data.

This thesis introduces new functionality related to, but different from, authentication and identification: group membership verification. The goal is to verify if a given IoT device is a member of a predefined group of enrolled devices. A group can be defined as devices with a particular type like medical equipment in a hospital environment, including CT and MRI scanners, ultrasound equipment, and X-ray equipment. Indeed, this func-

tionality strengthens privacy such that verification of one in a group of devices proceeds without disclosing its identity to the Cloud.

Group membership verification is different from group signature or ring signature concepts in cryptography, in which any member of the group can anonymously sign messages without revealing its identity. This ability is equivalent to proving that one is a member of the group. However, there is no enrollment of biometric or PUF templates since membership is equivalent to holding one of the valid keys. Besides, such protocols are not compatible with the IoT as heavy cryptography is not suitable for these applications. Therefore, this thesis addresses efficient privacy-preserving authentication, identification and group membership verification protocols that are appropriate for both biometric and IoT applications.

Context

The verification that an item, a device, or an individual is a member of a group is a natural task which forms the basis of many applications granting or refusing access to sensitive resources (buildings, wifi, payment, ...). Group membership can be implemented through a two-phase process where *identification* is first performed, revealing the identity of the individual under scrutiny, followed by a *verification* phase where it is checked whether or not the identified individual is indeed a member of the claimed group. That implementation breaks privacy: there is no *fundamental* reason to identify the individual before running the verification step. It is easier but not truly needed. It is fundamental to distinguish the members of the group from the non-members, but it does not require to distinguish members from one another.

The same comments hold for group *identification*. Such a system manages multiple groups, for example, separating individuals according to the team they work in. The goal is then to identify the precise group a member belongs to, without proceeding first to the identification of the individual.

Without any loss of generality, group membership verification needs first to acquire *templates* of items (PUF) or individuals (biometric trait) and to enroll them into a data structure stored in a server. Then, at verification time, that data structure is queried by

a client with a new template, and the access is granted or refused. Security assesses that the data structure is adequately protected so that an honest but curious server cannot reconstruct the templates. Privacy requires that verification should proceed without disclosing the identity.

The nature of templates can vary from one application to the other. For example, the templates encode information related to fingerprints, iris, or faces of individuals, or PUFs like speckle patterns captured from laser-illuminated transparent plastic (see Chapter 4).

It is worth highlighting two fundamental properties of the templates. The template used at verification time is a noisy version of the one acquired at enrollment time. Lighting conditions, blood pressure, aging, worn-outs, transient physical conditions are possible factors that might cause variations at acquisition time. The verification protocol must absorb such variations and cope with the continuous nature of the templates. However, it is very unlikely that a noisy version of the template corresponding to one group member gets similar enough to the enrolled template of any other group member. The first property is, therefore, in relation to the continuous and distinguishable nature of the templates. The second property is about the statistical independence of the enrolled templates.

The traditional operational scenario considers a server that runs the group membership verification. The server receives queries from clients. A client acquires a new template and then queries the server. Clients are trusted. The server is honest but curious: It might try to reconstruct the enrolled templates or spy on the queries. The design intends to prevent the server from reconstructing the private template from the system while correctly determining whether or not a user is a member of the claimed group (group verification) or identifying the group of membership (group identification).

Contribution

In this section, we list the main contributions made in this thesis.

1- Chapter 2 proposes a group membership verification protocol through the joint use of two mechanisms: quantizing templates into discrete embeddings and aggregating several templates into one group representation. First, we consider two independent procedures,

one for embedding, the other for aggregating and analyze the performance, security, and privacy by considering a simple statistical model of the enrolled templates together with a model of the query template. Thereafter, we replace those passive functions with functions producing the same types of output, but their parameters are learned through optimization. Finally, rather than considering group assignments that are predetermined, group assignments are also learned together with representations of the groups. We show the improvements through an extensive series of experiments targeting face recognition.

2- In chapter 3, we analyze a mathematical model for group membership verification. This scheme investigates the impact of the sparsity level of the high dimensional features representing group members on both security, compactness, and verification performances. It shows it is possible to trade compactness and sparsity for better security or better verification performance.

3- In chapter 4, we redesign a Siamese network to produce short descriptors for the PUF measurement data. We show the efficiency of learned descriptors for passive PUF authentication scenario (which has a similar structure as biometric). Then we design a practical group membership verification scheme suitable for learned descriptors. The main challenge is obtaining a trade-off between performance, security, and privacy. The statistical properties and information content of designed descriptors will have a significant impact on this trade-off.

Publications

This thesis build on the results previously published in the following publications.

- Marzieh Gheisari, Teddy Furon, Laurent Amsaleg, Behrooz Razeghi, and Slava Voloshynovskiy. "**Aggregation and embedding for group membership verification**." In IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2019.

- Marzieh Gheisari, Teddy Furon, and Laurent Amsaleg. "**Privacy preserving group membership verification and identification**." In IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2019.

- Marzieh Gheisari, Teddy Furon, and Laurent Amsaleg. "**Group membership verification with privacy: Sparse or dense?**." In IEEE International Workshop on Information Forensics and Security (WIFS), 2019.
- Marzieh Gheisari, Teddy Furon, and Laurent Amsaleg. "**Joint Learning of Assignment and Representation for Biometric Group Membership**." In IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2020.

Chapter 1

Related Works

In this thesis, we address privacy-preserving protocols in the context of both biometric and PUF-based applications. Since a PUF uses the physical characteristics of a device to produce a fingerprint, a PUF is unique to that device and thus it works like a biometric for humans. Indeed, IoT networks using PUF can face the same types of problems as biometric systems. Accordingly, this chapter provides an overview of different schemes to protect generic biometric template data.

Since the main idea behind our template protection scheme is to aggregate multiple templates into a unique template, we place a major focus on aggregation. Considering that this thesis deals with the intersection of security, machine learning, and computer vision, this chapter describes the classical methods used for data aggregation in the respective domains.

The rest of this chapter is organized as follows. Section 1.1 describes various biometric template protection methods. Different aggregation mechanisms are discussed in Section 1.2. Finally, Section 1.3 concludes the chapter with a brief summary and discussion.

1.1 Security of data

Many methods have been proposed and investigated in hopes of robustly securing biometric templates. Template protection schemes can be categorized as biometric cryptosystems, cancelable biometrics and approaches based on signal processing in the encrypted domain.

1.1.1 Biometric Cryptosystems

In biometric encryption schemes, biometric templates are associated with a user-specific key to produce the encrypted biometric template; then, the key can be released only if a genuine user's query is given at verification time. Biometric cryptosystem approaches either bind information with biometric templates or use biometric templates directly to derive keys. Both strategies produce biometric dependant public data called *helper data*. Based on how this helper data is used, there are two subclasses of schemes: key generation and key binding methods.

In key binding schemes, the users must provide secret information combined with their biometric templates to generate helper data. At verification time, by applying an appropriate key retrieval algorithm to the query sample, the key is recovered from the resulting helper data. Fuzzy commitment [JW99] and fuzzy vault [JS06] schemes are examples of key binding methods. In key generation schemes, both the helper data and the key are generated directly from the biometric template. Fuzzy extractor [DRS04] and secure sketch schemes [LSM06] are examples of key generating methods.

Most cryptobiometric systems rely on the fuzzy commitment and fuzzy vault schemes. In fuzzy commitment schemes, a biometric template is committed to a codeword of an error-correcting code. The difference between them and a hash value of the codeword is stored as helper data. At verification time, the difference vector is used to reconstruct the codeword from the query sample. Its corresponding hash is compared to the hash value stored as part of the helper data [RU10, HAD06, IW09].

There also exist cryptographic protocols setting a key management system to provide the members of a group with anonymous authentication [SPH99]. There is no enrollment of biometric or PUF templates since membership is equivalent to holding one of the valid keys. As for biometric or PUF applications, our scenario is different from authentication, identification, and secret binding. These applications secure the templates at the server and/or the client sides but ultimately reveal the user/object's identity.

1.1.2 Cancelable Biometrics

Cancelable biometric refers to approaches applying intentional, repeatable distortions of biometric template based on transformations, which provide a comparison of biometric templates in the protected domain. The transformations can be performed either in the signal domain or in the feature domain.

The templates are transformed using a one-way function, which means it is difficult to recover the original template, even if the transformation and the transformed templates are publicly available: if the transformed template is compromised, the attacker cannot acquire the personal information of the user. In addition, the user can revoke, or cancel, the cancelable biometric template, and a new one can be generated using the same functions with different parameters.

One of the earliest methods for generating cancelable biometric templates was based on non-invertible geometric transforms. [RCB01] applied geometric transformations in the image domain. At enrollment, the transform is applied to biometric inputs choosing application-dependent parameters. At verification time, the query image is transformed, employing the same parameters, and compared to the stored reference.

Random projection [PPCR10, PPCR11] is a non-invertible transformation that is broadly used for generating cancelable biometrics. With these methods, the extracted feature vector $\mathbf{x} \in \mathbb{R}^d$ is projected onto a random subspace $\mathbf{W} \in \mathbb{R}^{l \times d}$ with $l < d$ to generate the transformed vector. For any verification task to be effective, it is important that the relative distance between any two points in the original space be preserved as much as possible in the embedded space.

[TGN06] propose BioHashing method, which is an extension of random projection. In BioHashing using user-specific Tokenized Random Numbers, l orthogonal random vectors $\mathbf{w}_i$ are created. The l bit BioHash is calculated by applying a threshold on the dot product of the feature vector and all the user-specific random vectors. This technique can also be considered as a key binding scheme where the secret key is blended with biometric data to acquire a distorted biometric template, and secret subject-specific tokens (instead of public helper data) are used at verification.

Approaches based on feature mixing have also been proposed. [JLK+06] merges two different feature extraction methods to produce cancelable face biometrics: Principal Component Analysis (PCA) and Independent Component Analysis (ICA) coefficients are extracted, and both feature vectors are randomly shuffled and added to create a transformed template.

Cryptographically secure BioTokens is proposed by [Bou06]. The key idea is to split biometric features into stable and unstable components. For the face, real feature values are simply split into an integer component and a fractional component. Then, the stable component is encrypted in a secure fashion, and the unstable part is obscured by applying non-invertible projections.

Blending biometric data with helper data to derive a distorted version of the biometric template is known as biometric salting. Some of these salting methods use random noise patterns, synthetic patterns, and so on to create the transformed templates [ZRC08]. The main limitation of this method is how to determine what amount of artificial pattern we need to add. The addition of strong noise will reduce the discriminative property of original templates. In contrast, the addition of a weaker pattern will reduce the code's security, making it easier to extract the original template.

Schemes based on one-way functions produce the same code for two identical queries. Then, for example, the server might attempt to cluster queries from the same client so as to find the client interests based on the similarity of queries. Razeghi *et al.,* propose to increase uncertainty at the server-side by adding ambiguation noise to the query [RVKT17].

Approaches based on cancelable biometrics and cryptobiometrics suffer from at least one of the following shortcomings: The first is performance degradation compared to unprotected systems due to information loss, and the second is they require helper data. Attacks on this helper data can disclose sensitive information, which compromises the user's privacy and the security of the system.

1.1.3 Signal Processing in Encrypted Domain

As an alternative to biometric cryptosystems using helper data, signal processing in the encrypted domain based on Homomorphic Encryption schemes allow for computations

to be performed on ciphertexts (encrypted data), with no additional helper data. These approaches use a regular cryptographic key to encrypt the templates instead of bounding or generating them from biometric data, and there is no need to decrypt the protected templates at verification time.

These schemes can be categorized into two classes, Partially homomorphic encryption, and Fully homomorphic encryption systems. Partially homomorphic encryption schemes support only a single arithmetic operation, i.e., either addition or multiplication in the encrypted domain. In contrast, fully homomorphic encryption systems support unlimited additions and multiplication operations in the encrypted domain.

Biometric authentication methods based on homomorphic encryption [LEB12] were mostly based on partial homomorphic encryption schemes. Paillier cryptosystem [Pai99a] is a partially homomorphic encryption scheme which allows a party two types of computation: to obtain the encryption of the addition of two values available to him only in encrypted form and also the multiplication of a known integer value and a value available to him under encryption.

Erkin *et al.,* [EFG+09] proposed a privacy preserving face recognition system for eigenfaces by using the Paillier cryptosystem. After that, a more efficient approach is developed in [SSW10] to perform threshold comparison. Barni *et al.,* presented a fingerprint verification system based on homomorphic encryption on Fingercode templates in a semi-honest model [BBC+10], where the query is encrypted while the database stored in the server is not encrypted, which provides no security to the database. Gomez-Barrero *et al.,* [GBMG+17] have developed multi-biometric template protection schemes based on homomorphic encryption (Multi-biometric systems employ multiple biometrics of the same person in order to improve the recognition rate).

Recently, few works have demonstrated the use of Fully Homomorphic Encryption schemes, which support arbitrary computations on encrypted data. Gentry [GB09] introduced the first fully homomorphic encryption scheme, which is able to process in the encrypted domain both addition and multiplication operations at the same time. Hence, they allow the generation of encrypted inputs for any functionality, generating encryption of the result that can be employed by untrusted parties without revealing sensitive data.

Torres *et al.,* attempts to evaluate the capability of Fully Homomorphic Encryption schemes to protect user privacy in a biometric authentication model. The proposed protocol achieves promising security results, yet it is very computationally intensive [TBS14].

Although, any function with any complexity can be produced from these basic operations, they need a large number of operations, so they cannot be implemented efficiently [ABC+15]. Fully homomorphic encryption schemes also need huge sizes of the keys and the encrypted messages, thus, causes too high complexity for practical applications.

1.2 Aggregation Strategies

Low cost Partially homomorphic schemes can only protect either the query or the enrolled signatures. In order to protect both, we need to use fully homomorphic encryption. However, Implementation of these schemes in practice is difficult due to their high communication overhead and computational complexity. Our solution is to make database templates secured by aggregating different templates into one unique template. Following, we summarize various aggregation mechanisms in computer science, machine learning, and computer vision.

1.2.1 Aggregation in computer science

Group membership is linked to the well-known Bloom filter used to test whether an element is a member of a set. A Bloom filter hashes and blends n elements into one array of bits $\mathbf{R} \in \{0,1\}^m$ thanks to k hash functions.

At enrollment, the data structure is empty. Then, n elements to be enrolled are processed sequentially. Thanks to k independent hash functions, one object is mapped into k indices in $\{1,\ldots,m\}$. The bits of $\mathbf{R}$ associated with these indices are set to '1' (whatever their previous value). A query object is mapped into k indices at query time by the very same hash functions. The query is then verified as one member of the set if all the corresponding bits of $\mathbf{R}$ equal '1'.

A Bloom filter can not cause any false negative, *i.e.* probability of false negative

is exactly 0. Whereas it makes false positives with reduced probability at the cost of a larger array of bits. The number of hash functions minimizing this probability is $k = \lfloor \log(2)m/n \rfloor$. Then, the necessary length to meet a required false positive level ϵ is $m \geq -n \log(\epsilon)/(\log(2))^2$.

Gomez-Barrero *et al.,* [GBRL+18] [RGBB+15] proposed several template protection schemes based on Bloom filter belonging to Cancelable Biometrics. Basically, the Bloom filter template protection is used to obscure the data, through a non-invertible transformation. When Bloom filter is used in the context of privacy, it is demonstrated that a server cannot infer any information on one specific entry [BBL12]. However, a Bloom filter can not be used as is in our application for two reasons. First, a Bloom Filter deals with discrete objects, whereas we consider continuous high dimensional vectors as the templates. Using Bloom filters in our context would require turning templates into discrete objects. Designing that quantizer is challenging; it must absorb the noise, *i.e.* the difference between the enrolled and the fresh template. Second, at verification time, the hash of the query cannot be sent in the clear for privacy reasons [BKOS07]. For instance, Beck and Kerschbaum protect the query with partially homomorphic encryption since there is no need to protect the filter at the server side [BK13].

Bloom filter is an excellent tool for aggregation when there is absolutely no noise. In contrast, some contributions in the computer vision domain have demonstrated that it is possible to aggregate vectors while simultaneously dealing with their continuous nature and the presence of noise. Our contributions are in part inspired by these techniques. We, therefore, describe now some of the seminal works in computer vision that deal with vector aggregation.

1.2.2 Aggregation in computer vision

Aggregating vectors into one representation is a very common mechanism in computer vision. Local feature descriptors are utilized to characterize image patches and represent them by vectors. For example, in image classification and image retrieval tasks, aggregation techniques have been introduced to summarize the information contained in all the local features extracted from an image into a single descriptor. This section reviews some of the popular aggregation methods that aggregate local descriptors to learn compact image representations.

Bag of Words

For each image in the dataset, regions of interest are detected [Low04, BETVG08] and characterized by an invariant descriptor, which is d-dimensional. Next, the descriptors extracted from the training set are clustered into k clusters using the K-means clustering algorithm, which creates a visual vocabulary. An image is then represented by counting the occurrence of each visual word or computing the statistics on the deviation from the cluster centers.

The bag-of-words model [SZ03] can be seen as the first approach that provides such aggregation to produces a single vector for each image. Each feature $\mathbf{x}$ is assigned to the closest cluster center, and the image is represented as a histogram of occurrences of visual words. Given a set of local descriptors $\{\mathbf{x}_1, \ldots, \mathbf{x}_N\} \subset \mathbb{R}^d$ and k cluster centers $\{\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_k\}$, the image representation $\mathbf{v}$ obtained as:

$$v_j = \frac{1}{N} \sum_{\mathbf{x}_i \; s.t. \; NN(\mathbf{x}_i)=\boldsymbol{\mu}_j} 1 \tag{1.1}$$

where $NN(\mathbf{x})$ assigns descriptor $\mathbf{x}_i$ to its closest visual word from the precomputed visual vocabulary.

Methods employing significantly less visual words than a standard BoW are also proposed. Some methods such as VLAD [JPD+12] and Fisher vectors [PD07] obtain compact image representation by compressing BoW representation of an image. In the following, we will introduce the VLAD and Fisher Vector encodings in detail.

Fisher Vectors

Perronnin and Dance applied the Fisher Vector (FV), which is related to the approximation of the Fisher kernel, to compute the similarity for image classification [PD07].

The basic idea is to characterize how a sample of descriptors deviates from an average distribution modeled by a parametric generative model. The Gaussian Mixture Model (GMM) parameterizes the descriptor space using mean $\boldsymbol{\mu}_j$, covariance $\boldsymbol{\Sigma}_j$ and priors w_j for each visual word $j = 1, \ldots, k$. FV representation first computes the posterior probability

between the j-th visual word and the i-th descriptor with:

$$\theta_j(\mathbf{x}_i) = \frac{exp\{-\frac{1}{2}(\mathbf{x}_i - \boldsymbol{\mu}_j)^T \boldsymbol{\Sigma}_j^{-1}(\mathbf{x}_i - \boldsymbol{\mu}_j)\}}{\sum_{t=1}^{k} exp\{-\frac{1}{2}(\mathbf{x}_i - \boldsymbol{\mu}_t)^T \boldsymbol{\Sigma}_j^{-1}(\mathbf{x}_i - \boldsymbol{\mu}_t)\}} \tag{1.2}$$

The derivative of local descriptors with respect to the means and the covariances of the GMM is computed with:

$$\boldsymbol{\alpha}_j = \frac{1}{N\sqrt{w_j}} \sum_{i=1}^{N} \theta_j(\mathbf{x}_i) \boldsymbol{\Sigma}_j^{-\frac{1}{2}} (\mathbf{x}_i - \boldsymbol{\mu}_j) \tag{1.3}$$

$$\boldsymbol{\beta}_j = \frac{1}{N\sqrt{2w_j}} \sum_{i=1}^{N} \theta_j(\mathbf{x}_i) \boldsymbol{\Sigma}_j^{-1} (\mathbf{x}_i - \boldsymbol{\mu}_j) \odot (\mathbf{x}_i - \boldsymbol{\mu}_j) - 1 \tag{1.4}$$

Finally, the aggregated fisher vector is constructed by concatenating the derivatives (1.3) and (1.4) for each visual word.

Contrary to the BoW, which counts the occurrences of visual words, *i.e.* only taking into account 0-order statistics, the FV provides a comprehensive representation by employing higher-order statistics related to the descriptors distribution. Furthermore, because fewer visual words are required to achieve the same level of performance, FV results in a more efficient representation.

VLAD

The Vector of Locally Aggregated Descriptors (VLAD) is a global image descriptor aggregated from local descriptors, proposed by [JPD+12].

VLAD can be regarded as a non-probabilistic variant of the FV, which also keeps statistics about the relationship between visual words and local descriptors in addition to computing frequencies of each visual word, while VLAD applies a more straightforward aggregation technique.

Like BoW, the VLAD method starts with quantizing the local descriptors of an image by using a visual vocabulary learned by k-means. VLAD takes into account the aggregated difference between the visual words and the corresponding descriptors, rather than just counting the number of descriptors assigned to each visual word.
For each visual word $\boldsymbol{\mu}_j$, VLAD representation accumulates the residual $\mathbf{x}_i - \boldsymbol{\mu}_j$ for all

$\mathbf{x}_i$ such that $NN(\mathbf{x}_i) = \boldsymbol{\mu}_j$. Therefore, the component $\mathbf{v}_j$ associated with the visual word $\boldsymbol{\mu}_j$ is computed as:

$$\mathbf{v}_j = \sum_{\mathbf{x}_i \; s.t. \; NN(\mathbf{x}_i)=\boldsymbol{\mu}_j} \mathbf{x}_i - \boldsymbol{\mu}_j \tag{1.5}$$

Finally, all vectors $\mathbf{v}_j$ are concatenated into a dk-dimensional vector $\mathbf{x}$ (similar to the Fisher Vector).

Aggregating CNN activations

Convolutional Neural Networks (CNNs) can be regarded as the most powerful feature extractor used for many computer vision tasks like image retrieval and image classification. Previously, we reviewed the traditional aggregation mechanism used in computer vision involving handcrafted local features. In this section, we discuss the more recent research approaches that aggregate the feature maps from a CNN, to obtain the final image representation.

Indeed, the fully connected network can be conceptually understood as an aggregation mechanism that accumulates the convolutional features. Early application of CNNs included methods that use the fully-connected layer activations as the global image representation [SRASC14, GWGL14, BSCL14]. The work by Razavian *et al.,* replaced the two fully-connected layer with global pooling to aggregate the convolutional features [RSCM16]. A compact image representation is constructed in this fashion with dimensionality equivalent to the number of feature maps of the corresponding convolutional layer.

In particular, the authors of [RSCM16] applied global max-pooling on the feature maps to obtain the image representation. Similarly, [BL15] showed that sum-pooling over the feature maps of the last convolutional layer achieves the best performance in image retrieval. A hybrid scheme is the R-MAC method [TSJ15], which performs max-pooling over regions and ultimately applies sum pooling on the regional descriptors.

Popular encodings such as BoW, VLAD, and Fisher Vectors are redesigned on top of CNN activations. [MMO$^+$16] proposed utilizing the BoW method to encode the convolutional features of CNNs. Deep filter banks [CMKV16] formed Fisher Vector representation within the context of CNN activations. NetVLAD [AGT$^+$16] is also a CNN version of

the VLAD. To enable the NetVLAD layer to be differentiable, they replaced the hard assignment of descriptors to clusters (as in the original VLAD) with a soft assignment. It is shown that NetVLAD outperforms sum and max-pooling of CNN activations for the same dimensionality of image descriptors.

However, all the aggregation approaches that exist in the computer vision domain are designed to facilitate the identification of similar elements in images and have no security or privacy capabilities. Another recent aggregation method better fits with the security and privacy requirements that we need. In [IFG+17], Iscen *et al.,* design a strategy for packing a random set of image descriptors into a unique high-dimensional vector. One salient property of that strategy is that the similarity between images can be determined by solely comparing these (few) aggregated vectors to the description of the query, without the need of the original (and numerous) raw image descriptors. This saves space (memory footprint of the database) and time (complexity at query time). These gains are the main motivation of [IFG+17].

1.2.3 Aggregation in machine learning

This section explores the aggregation mechanisms used in machine learning, where the concept of aggregation is used in this domain with the purpose of keeping memory.

Associative memory is a data structure that maps the input pattern $\mathbf{x} \in \mathbb{R}^m$ to an output pattern $\mathbf{y} \in \mathbb{R}^n$. Associative memories store paired patterns $(\mathbf{x}_k, \mathbf{y}_k)$. Neural networks have been used as associative memories. Associative memory is represented by a matrix $\mathbf{W}$ whose components $\mathbf{w}_{ij}$ can be seen as the synaptic weights. Then, given an input pattern, the associative memory produces the paired output pattern.

Associative memories have learning and recall phases: storing and retrieving. In the learning (storing) phase, pairs of patterns are given then, the connections between neurons are modified, such that an "aggregated representation" of the stored patterns is constructed. During the recall (retrieval) phase, given a noisy version of a memorized pattern, the memory should retrieve the most relevant pattern that was stored.

Linear associator [Koh72, And72], the Hopfield neural networks [Hop82] and bidirectional associative memory models [Kos88] are some of the most popular artificial neural networks used to design associative memories. Many studies have also been performed to increase the maximum number of patterns that can be stored and then correctly retrieved by such models.

The linear associator is the simplest associative memory model, a feed-forward network with an input layer and an output layer. All input units are connected to all the output units via the connection weight matrix $W = [w_{ij}]_{m \times n}$, which stores the K different associated pattern pairs. During storing, the weights are modified according to Hebbian learning rule [Heb49] as $\mathbf{W} = \sum_{k=1}^{K} \mathbf{x}_k \times \mathbf{y}_k$, where $\times$ denotes outer product. After memorization, given an input pattern, the stored pattern is retrieved by one step of feed-forward computation.

Since the paired pattern is computed by a linear combination of the input patterns, a perfect retrieval can only happen if all the input patterns are pairwise orthogonal. Thus the number of patterns that the network can store is limited by the correlation among the input patterns.

Generally, any pattern recognition task can be considered to be a model of associative memory. For instance, in the image classification problem, an image is given to the network, and the task is to label the image. In connection with associative memory, the network stores a set of memory vectors. Then at query time, an incomplete pattern similar to one of the stored memories is fed to the network, and the task is to recover the full memory (recall the label) [VS08].

1.3 Conclusion

This thesis focuses on privacy preserving group membership verification procedures, checking whether an item or an individual is a member of a group. Our main contribution relies on the aggregation and the embedding of several distinctive templates into a unique and compact vector representing the members of a group, which allows a good assessment of the membership property at test time and also provides privacy and security. For security reason, the group representations must be adequately protected so that a honest but

curious server cannot reconstruct the signatures. For privacy reasons, verification should performed anonymously without disclosing identities.

Existing aggregation approaches in computer vision tasks attempt to aggregate multiple descriptors into one unique representation in order to facilitate the identification of similar elements in images. In face recognition tasks, multiple faces captured from the same person are also aggregated in order to enhance the robustness of features against changes in poses and expressions. However, these works are not concerned with privacy and security issues; we draw inspiration from them for our work.

Signal processing in the encrypted domain can provide a solution to group membership verification. At enrollment time, each template is quantized and protected with homomorphic encryption. The query is protected in the same manner at the verification stage. This allows to compute distances between the query and the templates in the encrypted domain and also compare the encrypted results to a threshold. These encrypted comparisons are sent back to the clients which decrypt and check whether there is at least one positive. Security and privacy are as high as the security of the cryptographic primitives. Homomorphic encryption, however, practical implementations of such schemes is a big challenge, suffering from large communication overhead and computational complexity.

Ultimately, low cost partially homomorphic encryption schemes which protect either the query or the enrolled signatures can be leveraged on top of our work. This means that the query will be protected by a partially homomorphic scheme while the security of database templates is provided by aggregation. Hence, more secure protocols will be built by combining such an encryption approach with our group membership verification scheme, where unauthorized parties cannot infer any information that they are not allowed to.

It should be noted that, the state-of-the-art is mature in the field of partilaly homomorphic encryption schemes so any further analysis falls outside the scope of this thesis. Although, this protocol is not implemented, our scheme is developed in compliance with this protocol. As all aggregated vectors are quantized to ensure adoption by cryptographic algorithms.

Chapter 2

Aggregation and Embedding for Group Membership

Introduction

Group membership verification protocols first enroll eligible *signatures* into a data structure stored at a server. Then, at verification time, the structure is queried by a client signature and the access is granted or not. For security, the data structure must be adequately protected so that a honest but curious server cannot reconstruct the signatures. For privacy, verification should proceed anonymously, not disclosing identities.

A client signature is a noisy version of the enrolled one, e.g. due to changes in lighting conditions. The verification must absorb such variations and cope with the continuous nature of signatures. They must be such that it is unlikely that a noisy version for one user gets similar enough to the enrolled signature of any other user. Continuity, discriminability and statistical independence are inherent properties of signatures.

The rest of the chapter is organized as follows. In Section 2.1, the group assignment is fixed, and group representation is computed using fixed embedding and aggregation functions. Then, in order to improve the performance of our group membership verification protocol, in Section 2.2, aggregation and embedding are learned jointly based on predefined assignments. Finally, in Section 2.3, group assignment, aggregation and embedding functions are learned all together.

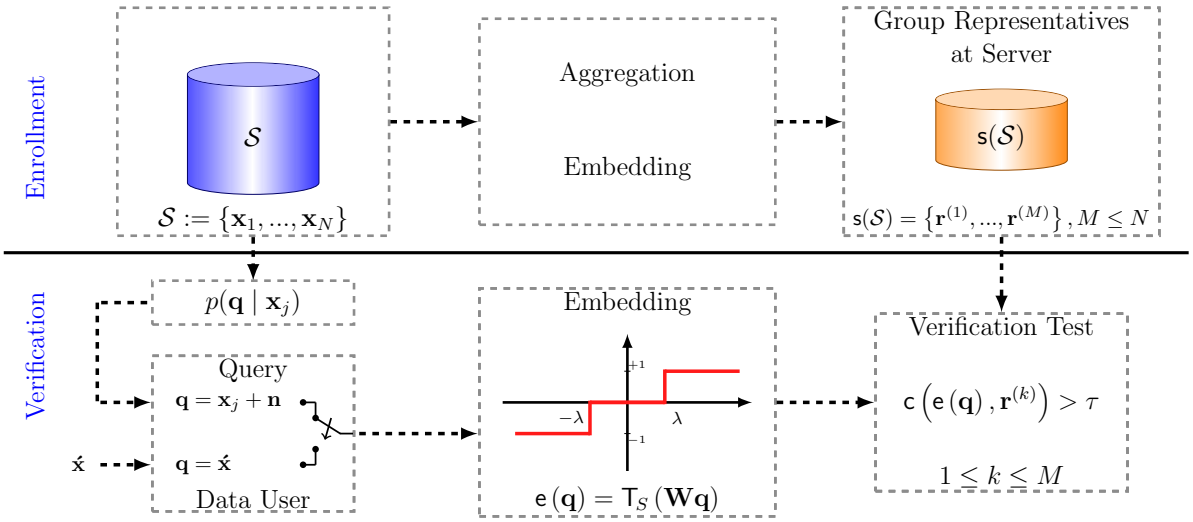


Figure 2.1 – Block diagram of the proposed model.

2.1 Aggregation and embedding for group membership verification

We propose a group membership verification protocol preventing a curious but honest server from reconstructing the enrolled signatures and inferring the identity of querying (trusted) clients. It combines two building blocks:

Block #1: One building block hashes continuous vectors into discrete *embeddings*. This lossy process limits the ability of the server to reconstruct signatures from the embeddings.

Block #2: The other building block *aggregates* multiple vectors into a unique representative value which will be enrolled at the server. The server can therefore not infer any specific signature from this value. Sufficient information must be preserved through the aggregation process for the server to assert whether or not a querying signature is a member of the group.

These two blocks can be assembled according to two configurations: block #1 before block #2, the system acquires and then hashes the signatures before aggregating them. The opposite configuration is where acquired signatures are aggregated before hashing the result of this aggregation.

At query time, the newly acquired signature is always hashed before being sent to the server. Weaknesses and strengths of these two configurations are explored in the following.

2.1.1 Notations and definitions

Signatures are vectors in $\mathbb{R}^d$. If N users/items belong to the group, then the protocol considers N signatures, $\mathcal{S} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\} \subset \mathbb{R}^d$. The signature to verify is a query vector $\mathbf{q} \in \mathbb{R}^d$. Group membership verification considers two hypotheses linked to the continuous nature of the signatures:

$\mathcal{H}_1$: The query is related to one of the N vectors. For instance, it is a noisy version of vector j, $\mathbf{q} = \mathbf{x}_j + \mathbf{n}$, with $\mathbf{n}$ to be a noise vector.

$\mathcal{H}_0$: The query is not related to any vector in the group.

We first design a group aggregation technique s which computes a single representation from all N vectors $\mathbf{r} := \mathsf{s}(\mathcal{S})$. This is done at the enrollment phase. Variable ℓ denotes the size in bits of this representation.

At the verification phase, the query $\mathbf{q}$ is hashed by a function e of size ℓ in bits. This function might be probabilistic to ensure privacy. The group membership test decides which hypothesis is deemed true by comparing $\mathsf{e}(\mathbf{q})$ and $\mathbf{r}$. This is done by first computing a score function c and thresholding its results: $t := [\mathsf{c}(\mathsf{e}(\mathbf{q}), \mathbf{r}) > \tau]$.

Verification Performances

The performance of this test are measured by the probabilities of false negative, $p_{\mathsf{fn}}(\tau) := \mathbb{P}(t = 0|\mathcal{H}_1)$, and false positive, $p_{\mathsf{fp}}(\tau) := \mathbb{P}(t = 1|\mathcal{H}_0)$. As τ varies from $-\infty$ to $+\infty$, these measures are summarized by the AUC (Area Under Curve) of the ROC (Receiver Operating Characteristic) curve. Figure 2.2 shows the AUC graphically. Another figure of merit is $p_{\mathsf{fn}}(\tau)$ for τ s.t. $p_{\mathsf{fp}}(\tau) = \epsilon$, a required false positive level.

Security and Privacy

A curious server can try to reconstruct a signature $\mathbf{x}$ from its embedding (for instance the query): $\hat{\mathbf{x}} = \mathsf{rec}(\mathsf{e}(\mathbf{x}))$. This endangers privacy of the querying user. The mean squared

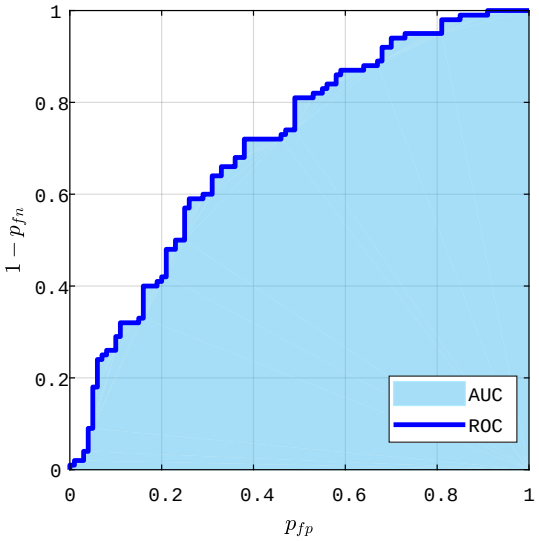


Figure 2.2 – AUC (Area under the ROC Curve).

error assesses how accurate is this reconstruction:

$$\mathsf{MSE}_P = d^{-1}\mathbb{E}(\|\mathbf{X} - \mathsf{rec}(\mathsf{e}(\mathbf{X}))\|^2) \tag{2.1}$$

The best reconstruction is known to be the conditional expectation: $\hat{\mathbf{x}} = \mathbb{E}(\mathbf{X}|\mathsf{e}(\mathbf{x}))$.

Reconstructing an enrolled signature from the group representation is even more challenging. For the security of the enrolled templates, a curious server can only reconstruct a single vector $\hat{\mathbf{x}}$ from the aggregated representation, and this vector serves as an estimation of any template in the group:

$$\mathsf{MSE}_S = (dN)^{-1}\sum_{j=1}^{N}\mathbb{E}(\|\mathbf{X}_j - \hat{\mathbf{X}}\|^2) \tag{2.2}$$

2.1.2 Verification for a few group members

This section discusses the verification protocol when N is small. We study the two different configurations for assembling block #1 and block #2.

Block #1: Embedding. An embedding $\mathsf{e} : \mathbb{R}^d \to \mathcal{A}^\ell$ maps a vector to a sequence of ℓ discrete symbols. This quantization shall preserve enough information to tell whether two embeddings are related, but not enough to reconstruct a vector. We intentionally

choose the sparsifying transform coding described in [RVKT17, RV18] for its security and privacy good properties. It projects $\mathbf{x} \in \mathbb{R}^d$ on the column vectors of $\mathbf{W} \in \mathbb{R}^{d\times\ell}$. The output alphabet $\mathcal{A} = \{-1, 0, +1\}$ is imposed by quantizing the components of $\mathbf{Wx}$. The $\ell - S$ components having the lowest amplitude are set to 0. The S remaining ones are quantized to +1 or -1 according to their sign.

$$\begin{aligned} \mathsf{e} : \mathbb{R}^d &\rightarrow \mathcal{A}^\ell \\ \mathbf{x} &\mapsto \mathsf{e}(\mathbf{x}) = \mathsf{T}_S(\mathbf{W}^T\mathbf{x}). \end{aligned} \tag{2.3}$$

Block #2: Aggregation. Aggregation a processes a set of input vectors to produce a unique output vector.

Whereas the embedding function e is fixed, we can have two constructions: The aggregation of embeddings (AoE) or the embedding of the aggregation (EoA).

AoE When block #1 is used before aggregation, that is, when considering $\mathsf{s} = \mathsf{a} \circ \mathsf{e}$, then $\mathsf{a} : \mathcal{A}^{\ell\times N} \rightarrow \mathcal{A}^\ell$. The group representative vector $\mathbf{r}$ is computed as:

$$\mathbf{r} = \mathsf{a}_{\text{AoE}}\left(\{\mathsf{e}(\mathbf{x}_i)\}_{i\in\mathcal{S}}\right). \tag{2.4}$$

EoA When block #2 is used before block #1, that is when considering $\mathsf{s} = \mathsf{e} \circ \mathsf{a}$, then $\mathsf{a} : \mathbb{R}^{d\times N} \rightarrow \mathbb{R}^d$:

$$\mathbf{r} = \mathsf{e}(\mathsf{a}_{\text{EoA}}(\mathbf{X})), \tag{2.5}$$

where $\mathbf{X}$ is the $d \times |\mathcal{S}|$ matrix storing the templates of the group.

Aggregation strategies

The nature of a highly depends on the type of vector the aggregation function receives. When considering $\mathsf{s} = \mathsf{e} \circ \mathsf{a}$, then a gets continuous signatures. In this case it is possible to design two aggregations schemes that are:

$$\mathsf{a}(\mathcal{S}) = \sum_{\mathbf{x}\in\mathcal{S}} \mathbf{x} = \mathbf{X}\mathbf{1}_N \quad \text{or} \tag{2.6}$$

$$\mathsf{a}(\mathcal{S}) = (\mathbf{X}^\dagger)^\top \mathbf{1}_N. \tag{2.7}$$

where $\mathbf{X} := [\mathbf{x}_1, \ldots, \mathbf{x}_N]$ is the $d \times N$ matrix, $\mathbf{1}_N := (1, \ldots, 1)^\top \in \mathbb{R}^N$, and $\mathbf{X}^\dagger$ is the pseudo-inverse of $\mathbf{X}$. Equation (2.6) is called the sum and (2.7) the pinv schemes in [IFG+17].

When considering $\mathsf{s} = \mathsf{a} \circ \mathsf{e}$, then a gets the embeddings of the signatures. Two additional aggregation strategies are the sum and sign pooling (2.8) and the majority vote (2.9):

$$\mathbf{r} = \mathsf{sign}(\sum_{\mathbf{x} \in \mathcal{S}} \mathsf{e}(\mathbf{x})) \quad \text{or} \tag{2.8}$$

$$r_i = \arg \max_{s \in \{-1,0,1\}} |\{\mathbf{x} \in \mathcal{S} | \mathsf{e}(\mathbf{x})_i = s\}| \tag{2.9}$$

Four resulting schemes

The assemblage of the blocks and the aggregation strategies overall create four variants. We name them:

- **EoA-2.6:** this scheme sums the raw signatures into a unique vector before embedding it in order to obtain $\mathbf{r}$. It therefore corresponds to the case where $\mathsf{s} = \mathsf{e} \circ \mathsf{a}$, the aggregation a being defined by (2.6).
- **EoA-2.7:** here also, aggregation precedes embedding, $\mathsf{s} = \mathsf{e} \circ \mathsf{a}$, and a is defined by (2.7).
- **AoE-2.8:** this scheme embeds each signature before aggregating with sum and sign pooling as defined by (2.8).
- **AoE-2.9:** here also, embedding precedes aggregation, but the majority vote is used as defined by (2.9).

The score function c comparing the hashed query with the group representation is always $\mathsf{c}(\mathsf{e}(\mathbf{q}), \mathbf{r}) = -\|\mathsf{e}(\mathbf{q}) - \mathbf{r}\|$.

2.1.3 Reconstruction and Verification

This section makes the following assumptions: i) Enrolled signatures are modelled by $\mathbf{X} \sim \mathcal{N}(\mathbf{0}_d, \sigma_x^2 \mathbf{I}_d)$, ii) Square orthogonal matrix $\mathbf{W}$ known by the attacker. It should be noted that this advantage is granted and may not reflect reality.

Ability to reconstruct from the embedding

Now that $\mathbf{W}$ preserves the norm, the MSE_P on $\mathbf{X}$ is the same as the mean square reconstruction error on $\mathbf{Z} = \mathbf{WX}$, which is also white Gaussian distributed. Thanks to the independance of the components of $\mathbf{Z}$, the conditional expectation can be computed component-wise. We introduce the density function conditioned on the interval $\mathcal{R}_s \subset \mathbb{R}$:

$$f(z|\mathcal{R}_s) := \phi_{\sigma_x}(z).\mathbb{1}_{\mathcal{R}_s}(z)/\mathbb{P}(Z \in \mathcal{R}_s), \tag{2.10}$$

with intervals $\mathcal{R}_0 = [-\lambda, \lambda]$, $\mathcal{R}_1 = (\lambda, +\infty)$, and $\mathcal{R}_{-1} = (-\infty, -\lambda)$. Function ϕ_{σ_x} is the p.d.f. of $Z \sim \mathcal{N}(0; \sigma_x^2)$ and $\mathbb{1}_{\mathcal{R}_s}$ is the indicator function of interval $\mathcal{R}_s$.

Observing the i-th symbol of $\mathsf{e}(\mathbf{x})$ equals s reveals that $z_i \in \mathcal{R}_s$. This component is reconstructed as $\hat{z}_i(s) := \mathbb{E}(Z|\mathcal{R}_s)$. Note that $\hat{z}_i(0) = 0$ because $f(z|\mathcal{R}_0)$ is symmetric around 0. For $s = 1$, the reconstruction value equals $\hat{z}_i(1) = \int_{-\infty}^{+\infty} z.f(z|\mathcal{R}_1)dz = \frac{\sigma_y}{p_1\sqrt{2\pi}} e^{-\frac{\lambda^2}{2\sigma_x^2}}$, where $p_1 := \mathbb{P}(Z \in \mathcal{R}_1) = \Phi(-\lambda/\sigma_x)$. By symmetry, $\hat{z}_i(-1) = -\hat{z}_i(1)$, and MSE_P admits the following close form:

$$\mathsf{MSE}_P = \sigma_x^2.\mathsf{MSE}(\lambda) \tag{2.11}$$

$$\mathsf{MSE}(\lambda) := 1 - \frac{1}{\pi\Phi(-\lambda/\sigma_x)} e^{-\frac{\lambda^2}{\sigma_x^2}}. \tag{2.12}$$

This quantity starts at $1 - 2\pi^{-1}$ when $\lambda = 0$. The embeddings are then full binary words ($p_1 = 1/2$). All components are reconstructed by $\pm\hat{z}_i$ but with a large variance. As λ increases, this variance decreases but less non-null components are reconstructed. $\mathsf{MSE}(\lambda)$ achieves a minimum of ≈ 0.19 for $\lambda \approx 0.60$, where 55% of the symbols of an embedding are non null. Then, $\mathsf{MSE}(\lambda)$ increases up to 1 for a large λ: the embeddings becomes sparser and sparser. When fully zero, each component is reconstructed by 0, and MSE_P equals σ_x^2.

Ability to reconstruct the signatures

The curious server tries to reconstruct a unique vector $\hat{\mathbf{x}}$ from $\mathbf{r}$ which represents the N enrolled signatures. Note that $\mathbf{r}$ is scale invariant: scaling the signatures by any positive factor does not change $\mathbf{r}$. Suppose that the curious server reconstructs $\hat{\mathbf{x}} = \kappa\mathbf{u}$. The best scaling minimizing MSE_S (2.2) is: $\kappa^\star = \|\mathbf{u}\|^{-2}\mathbf{u}^\top\mathbf{m}$, with $\mathbf{m} := N^{-1}\sum_{j=1}^{N}\mathbf{x}_j$. The curious

server can not compute $\kappa^\star$ giving birth to a larger distortion:

$$\mathsf{MSE}_S \geq \sum_{j=1}^{N} \|\mathbf{x}_j\|^2 - N\frac{(\mathbf{u}^\top \mathbf{m})^2}{\|\mathbf{u}\|^2}. \tag{2.13}$$

This lower bound is further minimized by choosing $\mathbf{u} \propto \mathbf{m}$.

Therefore, aggregation (2.6) is less secure as the other schemes do not allow the reconstruction of $\mathbf{m}$. In the worst case (2.6), the curious server estimates $\mathbf{m}$ by $N^{-1}\mathsf{rec}(\mathsf{e}(\mathsf{a}(\mathcal{S})))$:

$$\begin{aligned} d.\mathsf{MSE}_S &= \mathbb{E}\|\mathbf{X}_j - N^{-1}\mathsf{rec}(\mathsf{a}(\mathcal{S}))\|^2 \\ &= \mathbb{E}\|\mathbf{X}_j - \frac{\mathsf{a}(\mathcal{S})}{N}\|^2 + \frac{\mathbb{E}\|\mathsf{a}(\mathcal{S}) - \mathsf{rec}(\mathsf{a}(\mathcal{S}))\|^2}{N^2}. \end{aligned} \tag{2.14}$$

The first term is the squared distance between $\mathbf{X}_j$ and $\mathbf{m}$, whereas the second term corresponds to the error reconstruction for inverting the embedding. In the end:

$$\mathsf{MSE}_S = \sigma_x^2\left(1 - \frac{1}{N}(1 - \mathsf{MSE}(\lambda))\right). \tag{2.15}$$

This figure of merit increases with N because $\mathsf{MSE}(\lambda) \leq 1$, $\forall\lambda \geq 0$: Packing more signatures increases security.

Verification performance

We compare to a baseline defined as a Bloom filter optimally tuned for given N and p_{fp} having length $\ell_B = \lceil N|\log p_{\mathsf{fp}}|\log(2)^{-2}\rceil$ [FCAB00]. An embedding e is mandatory to first turn the real signatures into discrete objects. This means that, under $\mathcal{H}_1$, a false negative happens whenever $\mathsf{e}(\mathbf{x}_j + \mathbf{n}) \neq \mathsf{e}(\mathbf{x}_j)$.

Figure 2.3 shows the AUC vs. MSE (2.11) for the schemes of Section 2.1.2 for different sparsity S/d. Two schemes performs better. For low privacy (small MSE_P), EoA-2.7 achieves the largest AUC (with $0.5 \leq S/d \leq 0.6$) ; for high privacy, AoE-2.8 is recommended (with $S/d \geq 0.85$). In these regimes, the performances are better than the Bloom filter.

Figure 2.4 shows how the verification performances decrease as the number N of enrolled signatures increases. As mentioned in [IFG+17], the behavior of the aggregation

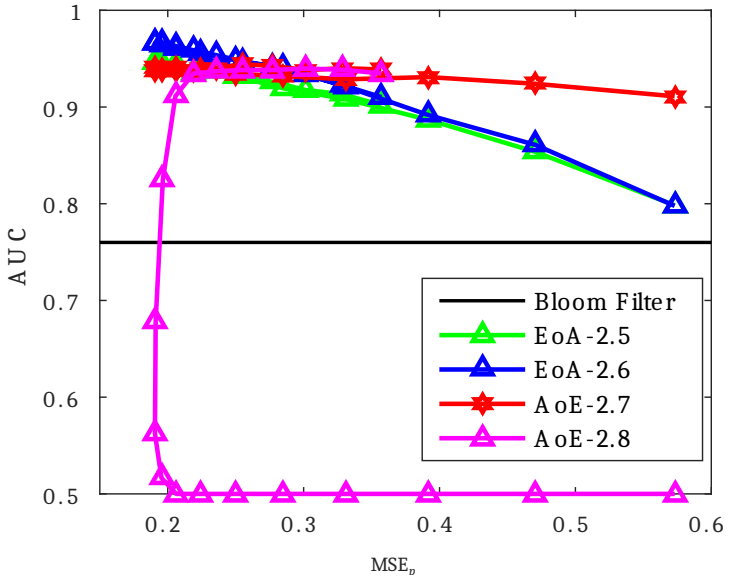


Figure 2.3 – Unique group: AUC vs. $\mathsf{MSE}_P/\sigma_y^2$. $N = 128$, $d = 1024$, $\sigma_n^2 = 0.01$ for varying $S \in (0.1 \times d, 0.9 \times d)$.

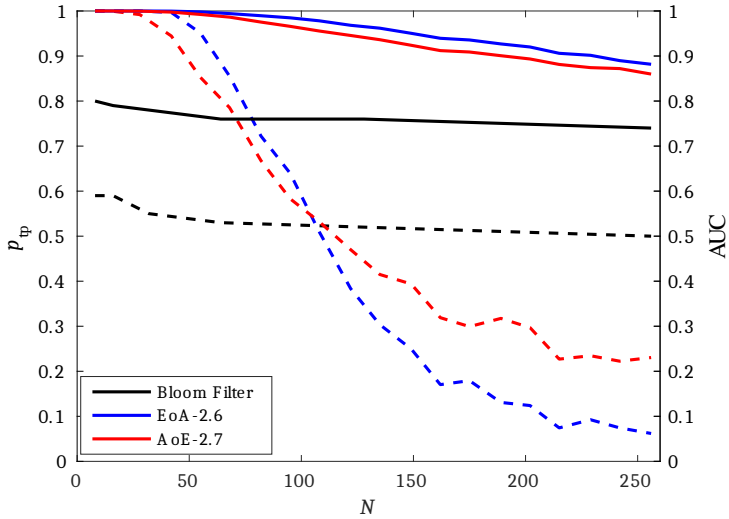


Figure 2.4 – Unique group: AUC and p_{tp} vs. N. Solid and dashed lines correspond to AUC and $p_{\mathsf{tp}}@p_{\mathsf{fp}} = 10^{-2}$.

scheme depends on the ratio N/d. The longer the signatures, the more of them can be packed into one representation.

2.1.4 Verification for multiple groups

When N is large, aggregating all the signatures into a unique $\mathbf{r}$ performs poorly. Rather, for large N, we propose to partition the enrolled signature into $M > 1$ groups, and to compute M different representatives, one per partition.

Random assignment: The signatures are randomly assigned into M groups of size $n = N/M$.

Clustering: Similar signatures are assigned to the same group. We used k-means algorithm to do so. Yet, the size of the groups is no longer constant.

Verification performance

Denote by $(p_{\mathsf{fp}}^{(k)}, p_{\mathsf{tp}}^{(k)})$ the operating point of group number k, $1 \leq k \leq M$. The overall system outputs a positive answer when at least one group test is positive. Denote by $(P_{\mathsf{fp}}(M), P_{\mathsf{tp}}(M))$ the performance of the global system. Under $\mathcal{H}_0$, the query is not related to any vector. Therefore,

$$P_{\mathsf{fp}}(M) = 1 - \prod_{k=1}^{M} (1 - p_{\mathsf{fp}}^{(k)}), \tag{2.16}$$

Under $\mathcal{H}_1$, the query is related to only one vector belonging to one group. A false negative occurs, if this test produces a false negative and the other tests a true negative each:

$$P_{\mathsf{fn}}(M) = \sum_{k=1}^{M} \frac{n_k}{N} p_{\mathsf{fn}}^{(k)} \prod_{l \neq k} (1 - p_{\mathsf{fp}}^{(l)}). \tag{2.17}$$

The operating point of a group test is mainly due to the size of the group. The random assignment creates even groups (if M divides N), so these share the operating point $(p_{\mathsf{fp}}, p_{\mathsf{tp}})$.

Figure 2.5 shows the experimental AUC and the one predicted by (2.16) and (2.17) when M ranges from 8 to 512. Since clustering makes groups of different sizes, we show the performances versus $n_{min} = \min_{1 \leq k \leq M}(n_k)$, where n_k is the size of k-th group. The

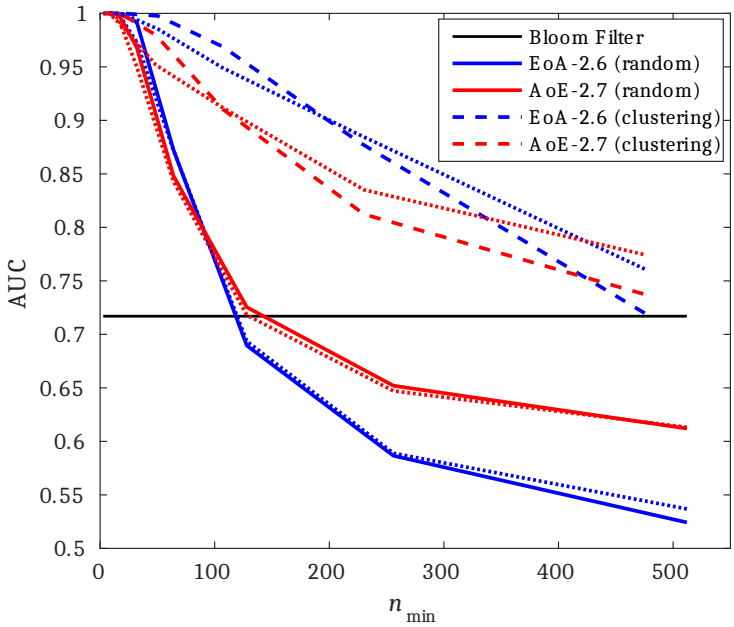


Figure 2.5 – Multiple groups: AUC vs. $n_{\min}$. Dotted lines are theoretical AUC. $N = 4096$, $d = 1024$, $\sigma_n^2 = 10^{-2}$, $S/d = 0.6$ for EoA-2.6, and $S/d = 0.85$ for AoE-2.7.

theoretical formulas are more accurate for random partitioning where the group are even. Estimations of $(p_{\mathsf{fp}}^{(k)}, p_{\mathsf{fn}}^{(k)})$ were less precise with the clustering strategy, and this inaccuracy cumulates in (2.16) and (2.17).

Clustering improves the verification performances a lot especially for EoA-2.7. A similar phenomenon was observed in [IFG+17]. Yet, Figure 2.6 shows that it does not endanger the system: MSE_S is only slightly smaller than for random assignment, and indeed close to 1 for $n_{\min} \geq 100$. This is obtained for $M = 32$ for EoA-2.7 giving $\mathsf{AUC} = 0.97$. The space is so big that the clusters are gigantic and not revealing much about where the signatures are. However, the anonymity is reduced because the server learns which group provided a positive test. This is measured in term of k-anonymity by the size of the smallest group, *i.e.* $n_{\min}$. Figure 2.5 indeed shows the trade-off between k-anonymity and the verification performances.

2.2 Learning aggregation and embedding jointly

In the previous section we proposed a privacy preserving group membership verification protocol quantizing templates into discrete embeddings and aggregating multiple embed-

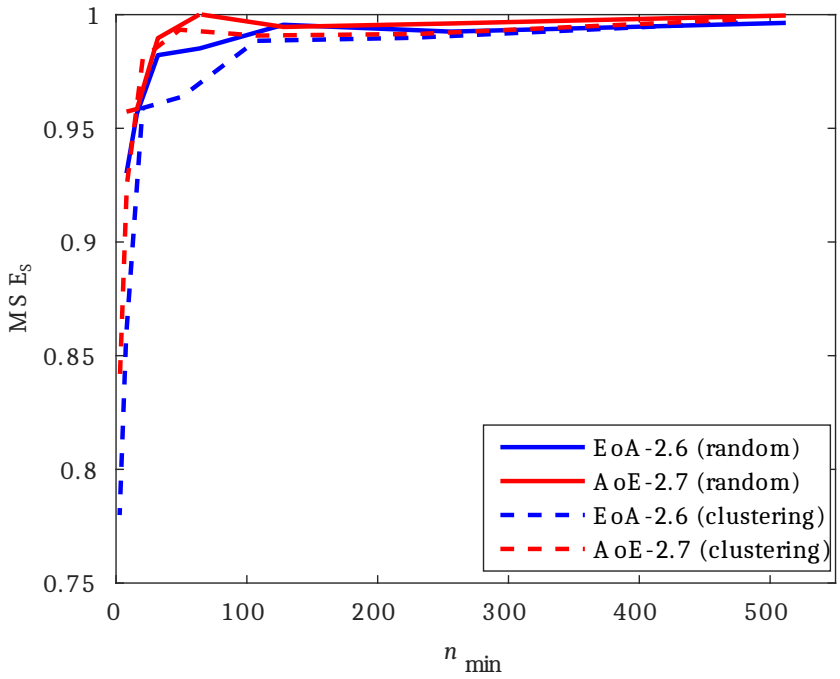


Figure 2.6 – Multiple groups: MSE_S vs. n_{min}. $N = 4096$, $d = 1024$, $\sigma_n^2 = 10^{-2}$, $S/d = 0.6$ (EoA-2.6) or 0.85 (AoE-2.7).

dings into a group representation. That scheme has several desirable properties: It is pure signal processing and linear algebra, hence it is cheap to run; quantization and aggregation fully succeed to make reconstruction difficult and impede identification; it is demonstrated to allow trading-off the strength of its security against group verification error rates.

That work, however, is fully deterministic in the sense that it sticks to a set of hard coded rules that drive the way templates are embedded, how they are grouped and then aggregated into group representations. Although well justified and sound, these rules govern the behavior of two independent procedures, one for embedding, the other for aggregating. In the next section we show that jointly considering the embedding and aggregation stages results in better performances, *i.e.* a better membership verification without damaging security.

2.2.1 Variants of the Protocol

This work aims at learning the aggregated vectors and the embeddings *jointly*. For both construction, this is done by minimizing an objective function summing a cost for embedding C^E and a cost for aggregating C^A.

For AoE (first embed, then aggregate), denote $\mathbf{E} \in \mathcal{A}^{\ell \times N}$ the matrix storing the embeddings of the enrolled templates. Like for $\mathbf{X}$, we write $\mathbf{E} := [\mathbf{E}_1, \ldots, \mathbf{E}_M]$ with $\mathbf{E}_g$ the matrix gathering the embeddings of the templates of group $\mathcal{S}_g$.

For EoA (first aggregate, then embed), denote $\mathbf{A} := [\mathbf{a}_1, \ldots, \mathbf{a}_M] \in \mathbb{R}^{d \times M}$ the matrix gathering the aggregations of the templates enrolled in a group.

Matrices $\mathbf{E}$ and $\mathbf{A}$ will be defined through optimization problems detailed below. For the embedding, function e is still prototyped according to (2.3). Papers [RVKT17, RV18] show that privacy and security stem from the sparsifying transform. Only its matrix $\mathbf{W}$ is learned.

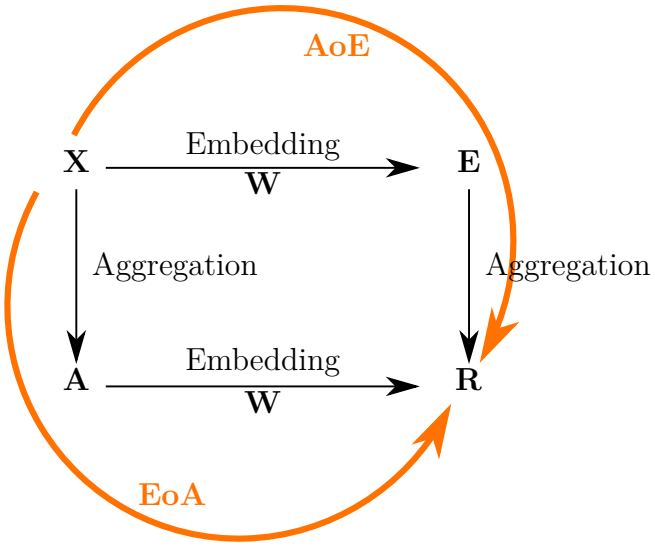


Figure 2.7 – Overview

For EoA, Figure 2.7 shows that it starts from $\mathbf{X}$ to create $\mathbf{A}$ before outputting $\mathbf{R}$ using matrix $\mathbf{W}$. This defines the optimization problem:

$$\min_{\mathbf{A},\mathbf{W},\mathbf{R}} \gamma C^A_{\text{EoA}}(\mathbf{X},\mathbf{A}) + C^E_{\text{EoA}}(\mathbf{A},\mathbf{R},\mathbf{W}), \tag{2.18}$$

where γ is the penalty parameter, $\mathbf{A} \in \mathbb{R}^{d\times M}$, $\mathbf{W} \in \mathbb{R}^{d\times \ell}$, and $\mathbf{R} \in \mathcal{A}^{\ell\times M}$.

On the other hand, Figure 2.7 shows that for AoE, we start from $\mathbf{X}$ to create $\mathbf{E}$ using matrix $\mathbf{W}$ before outputting the group representations $\mathbf{R}$. This defines the optimization problem:

$$\min_{\mathbf{E},\mathbf{W},\mathbf{R}} C^E_{\text{AoE}}(\mathbf{X},\mathbf{W},\mathbf{E}) + \xi C^A_{\text{AoE}}(\mathbf{E},\mathbf{R}), \tag{2.19}$$

with $\mathbf{E} \in \mathcal{A}^{\ell\times M}$, $\mathbf{W} \in \mathbb{R}^{d\times \ell}$, and $\mathbf{R} \in \mathcal{A}^{\ell\times M}$.

Under both constructions, the optimization is joint because the embedding and the aggregating costs share a common variable ($\mathbf{A}$ or $\mathbf{E}$). What follows define the costs and solve the optimization problems.

AoE: Aggregation of Embeddings

This scheme first embeds and then aggregates by solving (2.19). The cost for embedding is defined as:

$$C^E_{\text{AoE}}(\mathbf{X},\mathbf{W},\mathbf{E}) := \sum_{i=1}^{N} \left\| \mathbf{e}_i - \mathbf{W}^\top \mathbf{x}_i \right\|_2^2, \tag{2.20}$$

$$= \|\mathbf{E} - \mathbf{W}^\top \mathbf{X}\|_F^2. \tag{2.21}$$

This term represents the quantization loss of an embedding $\mathbf{e}$ w.r.t. a template $\mathbf{x}$ in the transformed domain.

For each group of embedded templates, the aggregated vector should satisfy some properties as well:

- For each group the overall distance between group members and the aggregated vector is minimized.
- Aggregated vector should be represented as a sparse ternary code.

The cost of aggregation is then defined as

$$C^A_{\text{AoE}}(\mathbf{E},\mathbf{R}) := \sum_{g=1}^{M} \|\mathbf{E}_g - \mathbf{r}_g \mathbf{1}_{|\mathcal{S}_g|}^\top\|^2. \tag{2.22}$$

We add the following constraints:

$$\begin{aligned} \mathbf{W}^\top \mathbf{W} &= \mathbf{I}_\ell, && (2.23)\\ \|\mathbf{e}_i\|_0 &\leq S,\ \forall i \in [N] && (2.24)\\ \|\mathbf{r}_g\|_0 &\leq S\ \forall g \in [M]. && (2.25) \end{aligned}$$

The constraint makes sure that the representative $\mathbf{r}$ is sparse, ternary, and diverse.

We propose to optimize problem (2.19) iteratively by alternating updates of one parameter while fixing the remaining ones. Each step minimizes the total cost function lower bounded by 0, insuring a convergence to a local minimum.

W-Step. We fix $\mathbf{E}$ and $\mathbf{R}$ and update $\mathbf{W}$ by solving:

$$\begin{aligned} \min_{\mathbf{W}} \quad & \left\|\mathbf{E} - \mathbf{W}^\top \mathbf{X}\right\|_F^2 \\ \text{s.t.} \quad & \mathbf{W}^\top \mathbf{W} = \mathbf{I}_\ell \end{aligned} \tag{2.26}$$

This problem is a least square Procruste problem with orthogonality constraint. By setting $\mathbf{S} := \mathbf{X}\mathbf{E}^\top$, [Sch66] shows that $\mathbf{W} = \mathbf{U}\mathbf{V}^\top$, where $\mathbf{U}$ contains the eigenvectors corresponding to the ℓ ($\ell < d$) largest eigenvalues of $\mathbf{S}\mathbf{S}^\top$ and $\mathbf{V}$ contains the eigenvectors of $\mathbf{S}^\top\mathbf{S}$.

E-Step. $\mathbf{W}$ and $\mathbf{R}$ being fixed, we can solve the problem for each $\mathbf{E}_g$ independently: $\forall g \in [M]$,

$$\begin{aligned} \min_{\mathbf{E}_g} \quad & \left\|\mathbf{E}_g - \mathbf{W}^\top \mathbf{X}_g\right\|_F^2 + \xi \left\|\mathbf{E}_g - \mathbf{r}_g \mathbf{1}_{|\mathcal{S}_g|}^\top\right\|_F^2 \\ \text{s.t.} \quad & \mathbf{E}_g \in \mathcal{A}^{\ell \times |\mathcal{S}_g|},\ \|\mathbf{e}_i\|_0 \leq S,\ \forall i \in \mathcal{S}_g. \end{aligned} \tag{2.27}$$

According to [RVKT17], we first find the solution without considering the constraints and then apply ternarization function T_S (2.3) to obtain sparse codes. Therefore $\mathbf{E}_g$ is found as:

$$\mathbf{E}_g = \mathsf{T}_S(\mathbf{W}^\top \mathbf{X}_g + \xi \mathbf{r}_g \mathbf{1}_{|\mathcal{S}_g|}^\top). \tag{2.28}$$

R-Step. Like for the **E**-step, updating each group representation $\mathbf{r}_g$ is done independently, while fixing $\mathbf{W}$ and $\mathbf{E}$:

$$\begin{aligned} \min_{\mathbf{r}_g} \quad & \left\| \mathbf{E}_g - \mathbf{r}_g \mathbf{1}_{|\mathcal{S}_g|}^\top \right\|_F^2 \\ \text{s.t.} \quad & \mathbf{r}_g \in \mathcal{A}^\ell, \ \ \|\mathbf{r}_g\|_0 \leq S. \end{aligned} \tag{2.29}$$

Then the representative of g-th group is obtained as:

$$\mathbf{r}_g = \mathsf{T}_S(\mathbf{E}_g \mathbf{1}_{|\mathcal{S}_g|}). \tag{2.30}$$

EoA: Embedding of Aggregation

We now consider the construction of (2.18) that first aggregates and then embeds. The cost of the aggregation is defined as:

$$C^A_{\text{EoA}}(\mathbf{X}, \mathbf{A}) := \sum_{g=1}^{M} \left\| \mathbf{X}_g^\top \mathbf{a}_g - \mathbf{1}_{|\mathcal{S}_g|} \right\|_2^2 + \eta \left\| \mathbf{a}_g \right\|_2^2 . \tag{2.31}$$

Minimizing this cost amounts to equalize the similarity between each members of the group and the aggregated vector $\mathbf{a}$. The cost for embedding is defined as previously:

$$C^E_{\text{EoA}}(\mathbf{A}, \mathbf{R}, \mathbf{W}) := \|\mathbf{R} - \mathbf{W}^\top \mathbf{A}\|_F^2. \tag{2.32}$$

As for the constraints:

$$\mathbf{W}^\top \mathbf{W} = \mathbf{I}_\ell, \tag{2.33}$$

$$\mathbf{r}_g \in \mathcal{A}^\ell, \|\mathbf{r}_g\|_0 \leq S, \ \forall g \in [M]. \tag{2.34}$$

The optimization problem (2.18) with these costs and constraints is solved by iterating the following steps.

W- Step. Like for (2.26), updating $\mathbf{W}$ while $\mathbf{R}$, $\mathbf{A}$ are fixed is a Procruste problem under orthogonality constraint:

$$\begin{aligned} \min_{\mathbf{W}} \quad & \left\| \mathbf{R} - \mathbf{W}^\top \mathbf{A} \right\|_F^2 , \\ \text{s.t.} \quad & \mathbf{W}^\top \mathbf{W} = \mathbf{I}_\ell. \end{aligned} \tag{2.35}$$

Similar to (2.26), we define $\mathbf{S} := \mathbf{A}\mathbf{R}^\top$. The solution is found as $\mathbf{W} = \mathbf{U}\mathbf{V}^\top$, where $\mathbf{U}$ contains the eigenvectors corresponding to the ℓ largest eigenvalues of $\mathbf{S}\mathbf{S}^\top$ and $\mathbf{V}$ the eigenvectors of $\mathbf{S}^\top\mathbf{S}$.

A- Step. When fixing $\mathbf{W}$ and $\mathbf{R}$, the aggregated vector for each group $g \in [M]$ is found independently by minimizing:

$$\min_{\mathbf{a}_g} \left\|\mathbf{r}_g - \mathbf{W}^\top\mathbf{a}_g\right\|_2^2 + \gamma(\left\|\mathbf{X}_g^\top\mathbf{a}_g - \mathbf{1}_{|\mathcal{S}_g|}\right\|_2^2 + \eta\left\|\mathbf{a}_g\right\|_2^2), \tag{2.36}$$

whose solution is

$$\mathbf{a}_g = (\mathbf{W}\mathbf{W}^\top + \gamma(\mathbf{X}_g\mathbf{X}_g^\top + \eta\mathbf{I}_d))^{-1}(\mathbf{W}\mathbf{r}_g + \gamma\mathbf{X}_g\mathbf{1}_{|\mathcal{S}_g|}).$$

R- Step. Projection matrix $\mathbf{W}$ and the group aggregations $\mathbf{A}$ are fixed. The group representatives are obtained by applying sparse ternarization function on the projected aggregated vectors: $\mathbf{R} = \mathsf{T}_S(\mathbf{W}^\top\mathbf{A})$.

2.2.2 Experiments

We implemented the group membership protocol that is described in Section 2.1. Experimenting with this implementation gives the baseline performances. Note that the experimental part of Section 2.1 only deals with synthetic data. However, the sequel presents comparisons on real data.

Experimental Setup

We evaluate the performances of the above scheme with face recognition. Face images are coming from LFW [HMBLM08], CFP [SCC+16] and FEI [TG10] databases. Face descriptors are obtained from a pre-trained network based on VGG-Face architecture [PVZ+15]. The output vector of the penultimate layer (*i.e.* before the final classifier layer) is PCA reduced to a lower dimension ($d = 1,024$ for LFW and CFP, $d = 256$ for FEI database), and then L_2-normalized. The result is the template $\mathbf{x} \in \mathbb{R}^d$. The values of ℓ, S, ξ, γ and η are set empirically as $0.9d$, $0.7l$, 1, 10^4 and 1 respectively. Also, not all individuals from these databases are enrolled.

LFW. Labeled Faces in the Wild contains 13,233 images of faces collected from the web. We used cropped LFW images. The enrollment set consists of $N = 1,680$ individuals with at least two images in the LFW database. One random template of each individual is enrolled in the system, playing the role of $\mathbf{x}_i$. The other templates are used for queries. These are partitioned into two subsets: templates that are correlated with $\mathbf{x}_i$ with a similarity bigger than 0.95 form the "easy queries" set; the remaining templates with a similarity bigger than 0.9 form the "hard queries" set. The remaining individuals not enrolled in the system $(5,749 - N)$ play the role of impostors (hypothesis $\mathcal{H}_0$).

CFP. The Celebrities in Frontal-Profile (CFP) database is composed of 500 subjects with 10 frontal and 4 profile images for each subject in a wild setting. We only use the frontal images. The impostor set is a random selection of 100 individuals. One random template of the remaining individuals is enrolled in the system. Like the setting described for LFW, we have two subsets of queries.

FEI. We use frontal and pre-aligned images of the Brazilian FEI database. There are 200 subjects with two frontal images (one with a neutral expression and the other with a smiling facial expression). The database is created by random sampling 150 individuals. For each identity, one random image is enrolled while the other is used as query. The remaining individuals are considered as impostors.

At the enrollment phase, all groups have exactly the same number of members: $|\mathcal{S}_g| = m, \forall g \in [M]$. Individuals are randomly assigned to a group. The performances of the system are gauged with error probabilities evaluated by Monte Carlo estimator over the testing set. Since N is not so large, the confidence interval at 95% is $\approx \pm N^{-1/2} = \pm 2.5\%$, which prevents us from estimating small probabilities. Therefore, *we put our system under stress by selecting a hard setup.*

First, note that LFW and CFP are difficult datasets due to the 'in the wild' variations (poses, illuminations, expressions and occlusions). They do not reflect the application of accessing a building (as mentioned in the Introduction) where the capture environment is more under control and the individuals collaborate. Second, not only the dimension of the templates have been reduced but also the length of the embeddings and the group representation ($\ell = 0.9d$) with a sparsity of $S = 0.7l$ (unless stated otherwise). Probabil-

ities of errors are then big but measurable with accuracy. We believe that this protocol makes sense to benchmark approaches.

Two applications scenarios are investigated: group verification and group identification.

Group verification. A user claims she/he belongs to group g. This claim is true under hypothesis $\mathcal{H}_1$ and false under hypothesis $\mathcal{H}_0$ (*i.e.* the user is an impostor). Her/his template $\mathbf{q}$ is embedded, and $(\mathsf{e}(\mathbf{q}), g)$ is sent to the system, which compares $\mathsf{e}(\mathbf{q})$ to the group representation $\mathbf{r}_g$. The system accepts ($t = 1$) or rejects ($t = 0$) the claim. This is a two hypothesis test with two probabilities of errors: $P_{\mathsf{fp}} := \mathbb{P}(t = 1|\mathcal{H}_0)$ is the false positive rate and $P_{\mathsf{fn}} := \mathbb{P}(t = 0|\mathcal{H}_1)$ is the false negative rate. The figure of merit is P_{fn} when $P_{\mathsf{fp}} = 0.05$.

Group identification. The scenario is an open set identification where the querying user is either enrolled or an impostor. The system has two steps. First, it decides whether or not this user is enrolled. This is verification as above, except that the group is unknow: The system computes $\delta_j = \|\mathsf{e}(\mathbf{q}) - \mathbf{r}_j\|$, $\forall j \in [M]$. The system accepts ($t = 1$) if the minimum of these M distances is below a given threshold τ. The figure of merit is P_{fn} when $P_{\mathsf{fp}} = 0.05$.

When $t = 1$, the system proceeds to the second step. The estimated group is given by $\hat{g} = \arg\min_{j\in[M]} \delta_j$. The figure of merit for this second step is $P_\epsilon := \mathbb{P}(\hat{g} \neq g)$ or the Detection and Identification Rate $DIR := (1 - P_\epsilon)(1 - P_{\mathsf{fn}})$.

Exp. #1: Comparison to the baseline

Figure 2.8 shows that our method brings improvement compared to the baseline, since the AoE and the EoA plots are way below the ones corresponding to the baseline. The high probabilities of false negatives for the baseline are caused by the great losses in information: AoE (Baseline) looses information from each template it embeds before the aggregation—the accumulated losses are therefore great; EoA (Baseline) has better performances since plain templates are first aggregated before running the embedding step which causes less information loss.

Our method does not suffer that much from this information loss: EoA and AoE have roughly similar performances, with much more acceptable P_{fn} values.

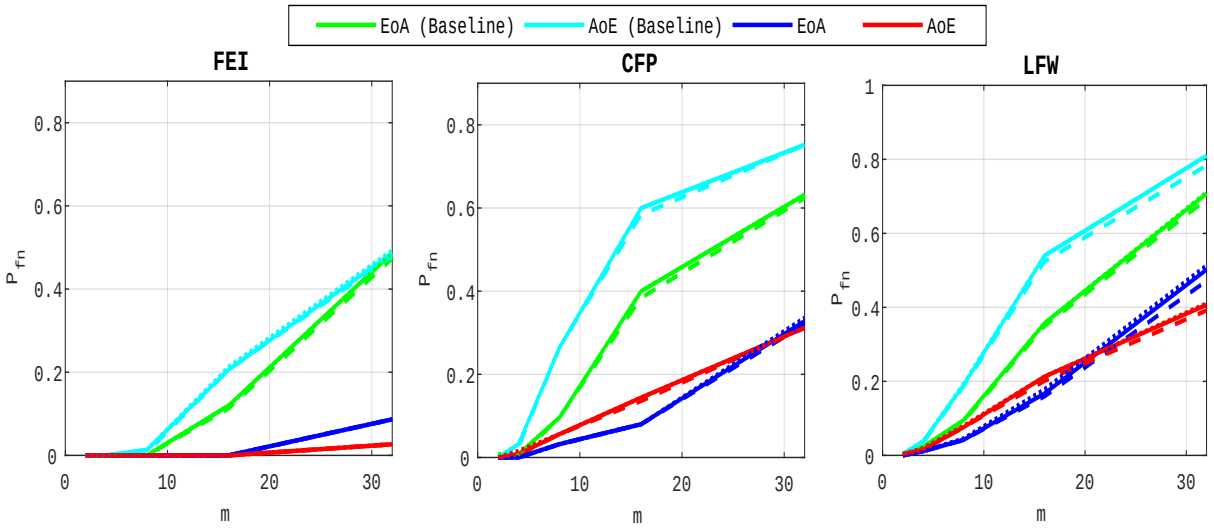


Figure 2.8 – Performances comparison with baselines for varying group size m. P_{fn} at $P_{\text{fp}} = 0.05$ for group verification (solid), the first step of group identification (dashed), and P_ϵ for the second step of group identification (dotted).

Exp. #2: Detection and Identification Rate

Figure 2.9 compares the DIR performances for group identification with $m = 16$. Our schemes have results close to perfection on the FEI dataset. Easy queries are correctly handled on CFP but not on the LFW dataset at this size of group. Hard queries are more difficult to cope with. This is explained by the poor correlations they have with their corresponding $\mathbf{x}_i$. That poor correlation, already existing on the original templates, before any embedding or aggregation, can only lower the performances of any membership identification scheme.

Figure 2.10 shows the impact of the size of group on DIR. Packing more templates into one group representation is detrimental even if the queries are well correlated with their corresponding enrolled template. This suggests to split large groups into subgroups of size lower or equal to $m = 32$. This restricts privacy to m-anonymity as the server is now able to identify the subgroup a query belongs to.

Exp. #3: Easy vs. Hard Queries

Figure 2.11 gives an additional perspective on the phenomenon highlighted above, that is, the genuine similarity between the query and the enrolled template is a key factor. Easy

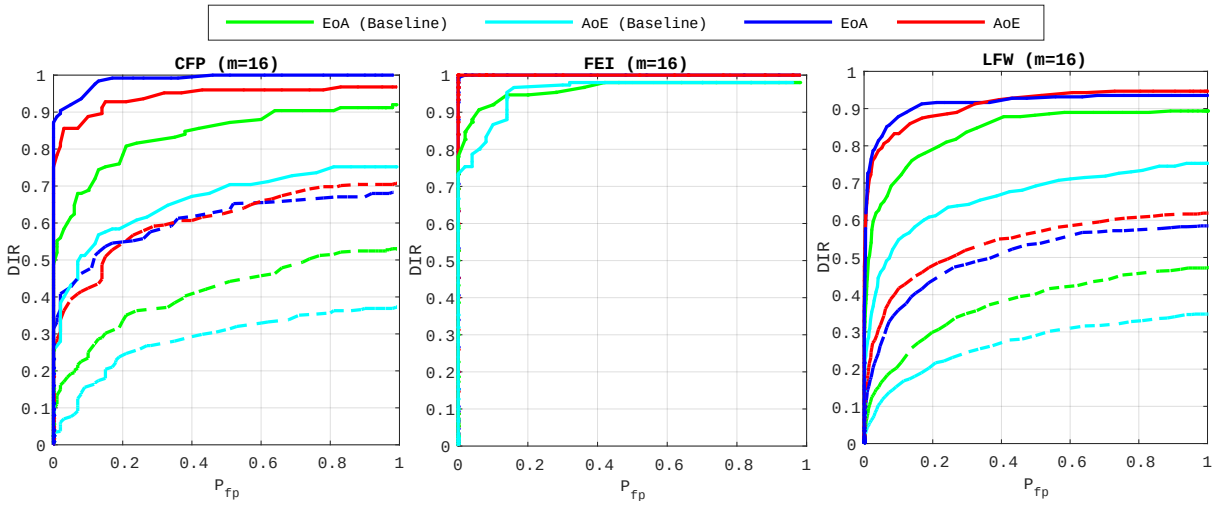


Figure 2.9 – The Detection and Identification Rate (DIR) vs. P_{fp} for group identification. Performances for hard queries are plotted in dashed lines.

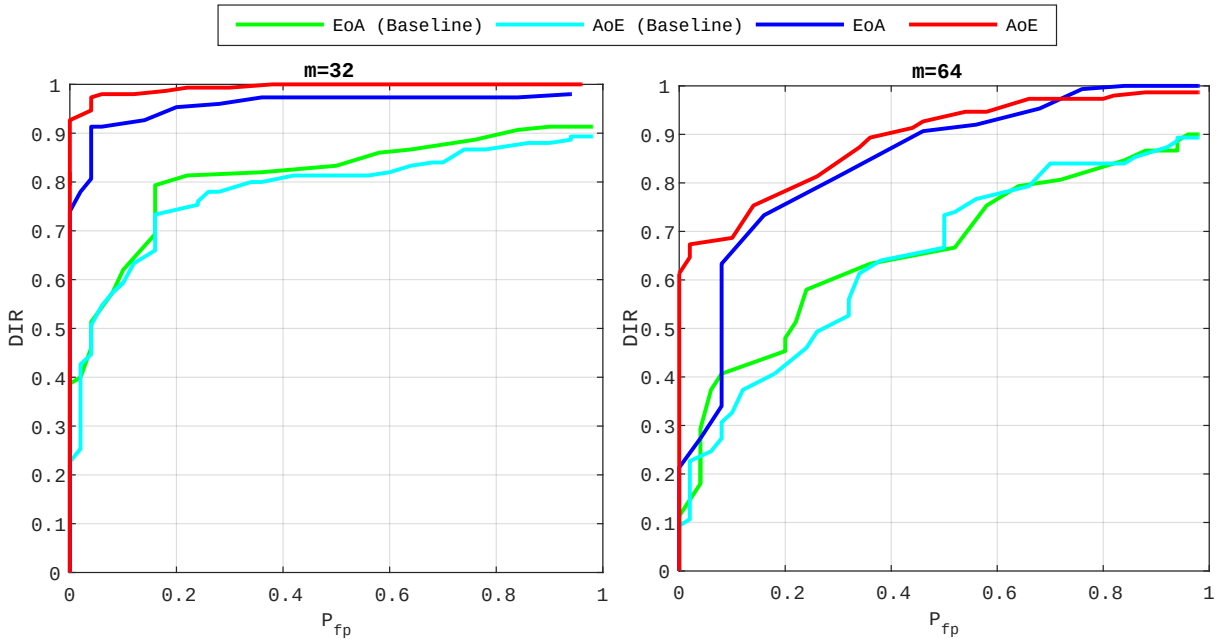


Figure 2.10 – The Detection and Identification Rate (DIR) vs. P_{fp} for group identification on FEI.

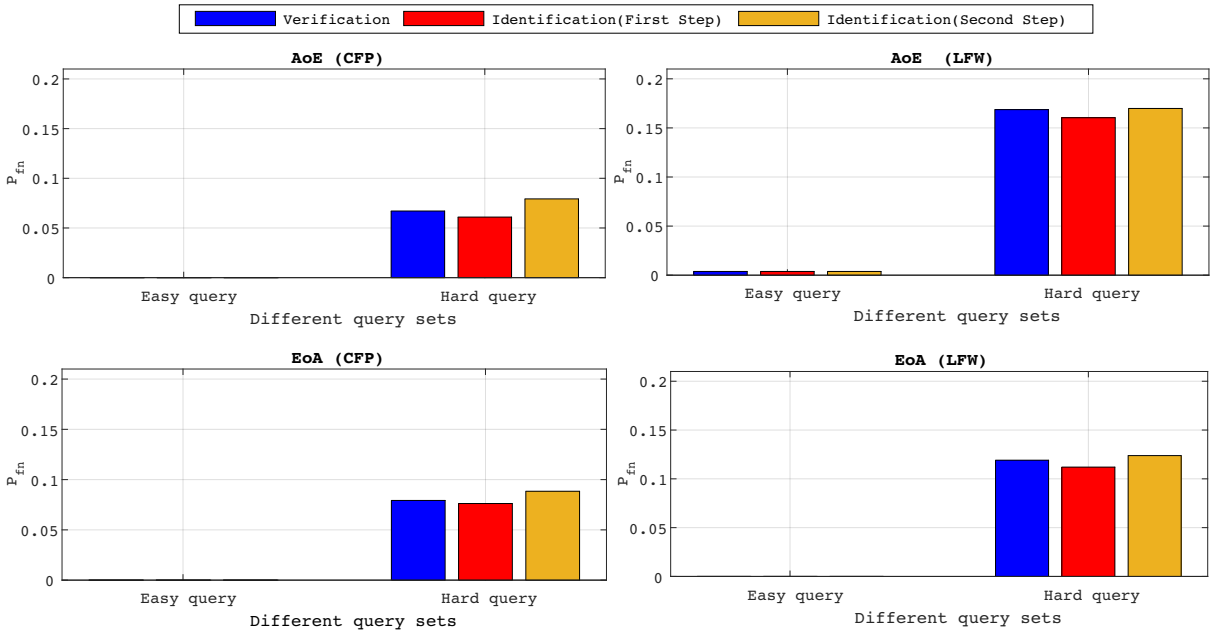


Figure 2.11 – The impact of the similarity of the query with the enrolled template on group verification and identification.

queries are very well handled whereas hard queries are more problematic. Put differently, the proposed method do not severely degrade the recognition power of the descriptors obtained through the VGG16 network.

Descriptors poorly correlated already at the image level can only cause poor performance once embedded and aggregated. This is also shown in Figure 2.12 which displays some enrolled and querying faces of the 'in the wild' datasets LFW and CFP. All the failed identification examples show a change of lighting, pose or expression, and / or occlusion. Yet, such changes do not automatically give a failure.

Exp. #4: Security and Privacy

As for the security and privacy, the quantities (2.1) and (2.2) were measured as empirical average over the dataset. Knowing that the query has unit norm, the reconstruction mechanism yields a unit vector as follows: $\hat{\mathbf{q}} = \mathbf{W}\mathsf{e}(\mathbf{q})/\|\mathbf{W}\mathsf{e}(\mathbf{q})\|$. The quality of the reconstruction mainly depends on the sparsity factor S. When S is small, the template is reconstructed with few columns of $\mathbf{W}$. When S is big, more columns are used but the amplitude modulating each column is coarsely reconstructed.

Figure 2.12 – Examples of group identification on CFP(left) and LFW(right). Blue frames indicate enrolled samples, green / red frames successful / failed queries, respectively.

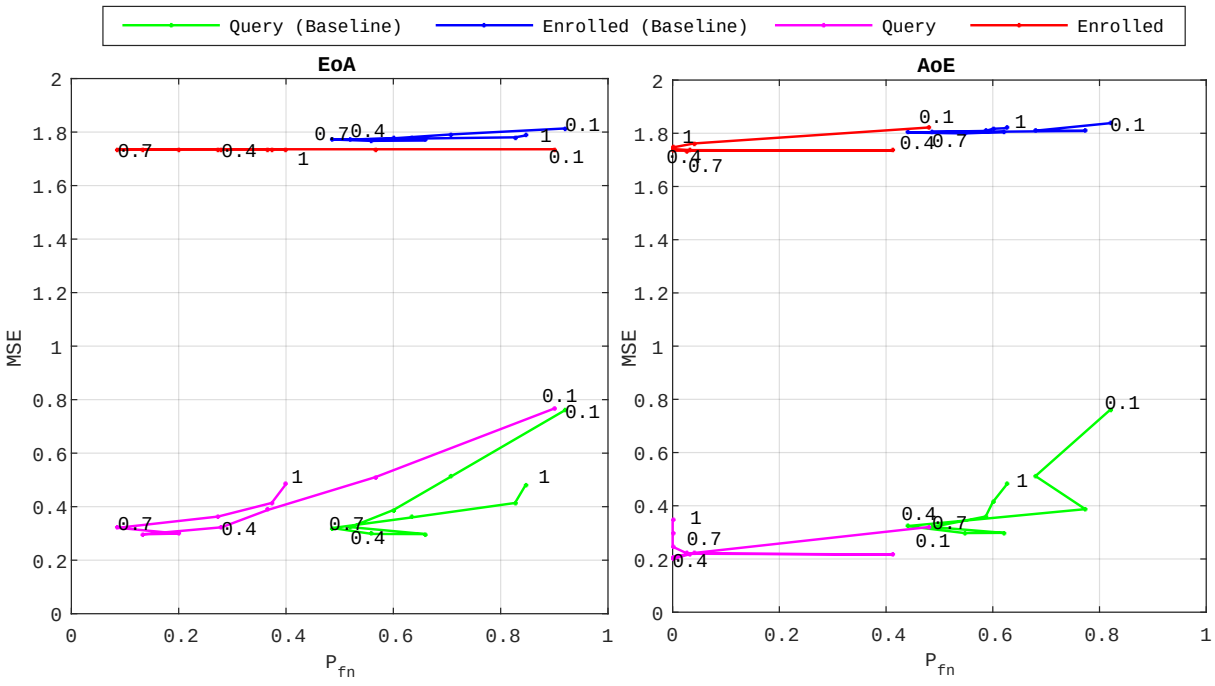


Figure 2.13 – The impact of sparsity factor S on the trade-off between security and performances, on FEI with $m = 32$ and $S/d \in (0.1, 1)$.

Figure 2.13 demonstrates the trade-off between security and performance for different sparsity levels. In this figure, the horizontal axis is P_{fn} so it represents the performance and the vertical axis is MSE. As we measure the security by the ability of the server to reconstruct enrolled signatures from group representations *i.e.* MSE_S and for privacy we consider the ability of the server to reconstruct the query from its embedding *i.e.* MSE_P.

There might be two values of S, one small, one large, providing the same reconstruction MSE. However, these two values do not yield the same performances. Also, reconstructing enrolled templates is even more difficult due to the aggregation (see (2.2)). Overall, we observe that our method has decreased the security a little, but the trade-off between security and performances is more interesting especially for AoE.

2.3 Joint learning of assignments and representations

In the previous section, we proposed a framework based on aggregation and embedding of several templates into a unique vector representing the members of a group. That work, however, is deterministic in the sense that it learns group representations based on predefined groups. This section introduces an optimization problem to learn jointly the group representations and group assignments and shows that it can achieve better performance without damaging the security. This addresses scenarios where the number of members is very large. Their signatures can not be packed into one unique group representation with a technique like section 2.2. Therefore, members are assigned to different groups automatically. An additional light cryptographic protocol is deployed to secure their privacy during group verification.

2.3.1 Formulation of the optimization problem

The embedding, the assignment, and the group representations are learned jointly at enrollment, and given to a server. Biometric signatures are modelled as vectors in $\mathbb{R}^d$. $\mathbf{X} \in \mathbb{R}^{d \times N}$ is the matrix of the signatures to be enrolled into M groups. The group representations are stored column wise in $\ell \times M$ matrix $\mathbf{R}$. The group representations are quantized and sparse *i.e.*, $\mathbf{r}_g \in \mathcal{A}^\ell$ with $\mathcal{A} = \{-1, 0, +1\}$ and $\|\mathbf{r}_g\|_0 \leq S < \ell, \forall g \in [M]$.

Our group membership protocol aims at jointly learning the partition, the embedding and the group representations. The key is to introduce the auxiliary data $\mathbf{E} = [\mathbf{e}_1, \ldots, \mathbf{e}_N] \in \mathcal{A}^{\ell \times N}$ the hash codes of enrolled signatures and $\mathbf{Y} \in \mathbb{R}^{M \times N}$ the group indicator matrix ($y_{i,j} = 1$ if $\mathbf{e}_j$ is assigned to i-th group). Then, the optimization problem is composed of a cost for embedding C^E and a cost for partitioning $C^{A,G}$:

$$\min_{\mathbf{W},\mathbf{R},\mathbf{Y}} C^E(\mathbf{X}, \mathbf{W}, \mathbf{E}) + C^{A,G}(\mathbf{E}, \mathbf{Y}, \mathbf{R}), \tag{2.37}$$

The embedding cost is the loss for quantizing signatures:

$$C^E(\mathbf{X}, \mathbf{W}, \mathbf{E}) \;:=\; \sum_{i=1}^{N} \left\| \mathbf{e}_i - \mathbf{W}^\top \mathbf{x}_i \right\|_2^2. \tag{2.38}$$

The assignment aims at grouping together signatures sharing similar hash codes: the overall dissimilarity between members and their group representation is minimized while the separation between two groups is maximized. Inspired by Linear Discriminant Analysis, we consider variance to measure dissimilarity. The within-group scatter matrix $\mathbf{S}_w$ and the between-group scatter matrix $\mathbf{S}_b$ are defined as

$$\begin{aligned} \mathbf{S}_w &= \sum_{g=1}^{M} \sum_{i \in \mathcal{Y}_g} (\mathbf{e}_i - \mathbf{r}_g)(\mathbf{e}_i - \mathbf{r}_g)^\top = (\mathbf{E} - \mathbf{R}\mathbf{Y})(\mathbf{E} - \mathbf{R}\mathbf{Y})^\top \\ \mathbf{S}_b &= \sum_{g=1}^{M} \mathbf{r}_g \mathbf{r}_g^\top = \mathbf{R}\mathbf{Y}(\mathbf{R}\mathbf{Y})^\top \end{aligned}$$

where $\mathcal{Y}_g = \{i \in [N] : y_{g,i} = 1\}$. The cost for partitioning is $C^{A,G} = \lambda Tr(\mathbf{S}_w) - \gamma Tr(\mathbf{S}_b)$ for some λ, γ in $\mathbb{R}_+$.

In the end, the objective function is formulated as:

$$\begin{aligned} \min_{\mathbf{W},\mathbf{R},\mathbf{Y}} \quad & \|\mathbf{E} - \mathbf{W}^\top \mathbf{X}\|_F^2 + \lambda Tr(\mathbf{S}_w) - \gamma Tr(\mathbf{S}_b) \\ \text{s.t.} \quad & \mathbf{W}^T \mathbf{W} = \mathbf{I}_\ell \\ & \mathbf{Y} \in \{0,1\}^{M \times N}, \;\; \|\mathbf{y}_i\|_1 = 1 \; \forall i \in [N] \\ & \mathbf{e}_i \in \mathcal{A}^\ell, \;\; \|\mathbf{e}_i\|_0 \leq S \\ & \mathbf{r}_g \in \mathcal{A}^\ell, \;\; \|\mathbf{r}_g\|_0 \leq S \end{aligned} \tag{2.39}$$

The constraint on $\mathbf{Y}$ ensures that each signature belongs to exactly one group.

2.3.2 Suboptimal solution

The solution of (2.39) is found by iterating the following steps:

W-Step. We fix $\mathbf{E}$, $\mathbf{R}$, $\mathbf{Y}$ and update $\mathbf{W}$ by solving:

$$\begin{aligned} \min_{\mathbf{W}} \quad & \left\|\mathbf{E} - \mathbf{W}^\top \mathbf{X}\right\|_F^2 \\ \text{s.t.} \quad & \mathbf{W}^\top \mathbf{W} = \mathbf{I}_\ell \end{aligned} \tag{2.40}$$

This problem is again a least square Procruste problem with orthogonality constraint. By setting $\mathbf{S} := \mathbf{X}\mathbf{E}^\top$, [Sch66] shows that $\mathbf{W} = \mathbf{U}\mathbf{V}^\top$, where $\mathbf{U}$ contains the eigenvectors corresponding to the ℓ ($\ell < d$) largest eigenvalues of $\mathbf{S}\mathbf{S}^\top$ and $\mathbf{V}$ contains the eigenvectors of $\mathbf{S}^\top\mathbf{S}$.

E-Step. Given $\mathbf{W}$, $\mathbf{Y}$ and $\mathbf{R}$, (2.39) amounts to:

$$\begin{aligned} \min_{\mathbf{E}} \quad & \left\|\mathbf{E} - \mathbf{W}^\top \mathbf{X}\right\|_F^2 + \lambda \left\|\mathbf{E} - \mathbf{R}\mathbf{Y}\right\|_F^2 \\ \text{s.t.} \quad & \mathbf{e}_i \in \mathcal{A}^\ell, \;\; \left\|\mathbf{e}_i\right\|_0 \leq S \end{aligned} \tag{2.41}$$

We first find the solution relaxing the constraints and then apply ternarization function T_S to obtain sparse codes:

$$\mathbf{E} = \mathsf{T}_S(\mathbf{W}^\top \mathbf{X} + \lambda \mathbf{R}\mathbf{Y}). \tag{2.42}$$

(R,Y)-Step. When fixing $\mathbf{W}$ and $\mathbf{E}$, the assignment and group representations are found by minimizing:

$$\begin{aligned} \min_{\mathbf{R},\mathbf{Y}} \quad & \left\|\mathbf{E} - \mathbf{R}\mathbf{Y}\right\|_F^2 - \frac{\lambda}{\gamma} Tr(\mathbf{R}\mathbf{Y}\mathbf{Y}^\top\mathbf{R}^\top) \\ \text{s.t.} \quad & \mathbf{Y} \in \{0,1\}^{M\times N}, \;\; \left\|\mathbf{y}_i\right\|_1 = 1 \; \forall i \in [N] \\ & \mathbf{r}_g \in \mathcal{A}^\ell, \;\; \left\|\mathbf{r}_g\right\|_0 \leq S \end{aligned} \tag{2.43}$$

As $\mathbf{E}$ is fixed, $Tr(\mathbf{E}\mathbf{E}^\top)$ is irrelevant to $\mathbf{Y}$, thus minimizing (2.43) is equivalent to:

$$\min_{\mathbf{R},\mathbf{Y}} \quad \left\|\frac{\lambda}{\lambda - \gamma}\mathbf{E} - \mathbf{R}\mathbf{Y}\right\|_F^2. \tag{2.44}$$

Relaxing the ternarization constraint, (2.44) is solved by a k-means clustering algorithm,

*i.e.*iteratively:

- *Update assignments*: Each item is assigned to its nearest group representative.
- *Update centroids*: g-th centroid is the mean of all $\tilde{\mathbf{e}}_i$ in group g.

Then the group representation $\mathbf{r}_g$ is found by applying ternarization function on g-th centroid.

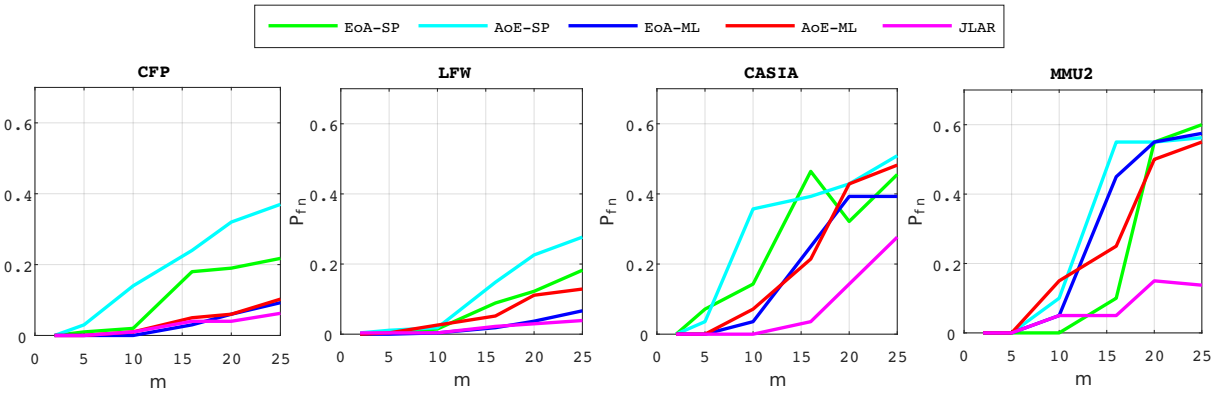


Figure 2.14 – Performances comparison for varying group size m. P_{fn} at $P_{\mathsf{fp}} = 0.05$ for group verification.

2.3.3 Experiments

This section presents the datasets used in our experiments and investigates the performance of the JLAR for two application scenarios. We compare JLAR with EoA-SP, AoE-SP (Signal Processing approach, Section 2.1) and EoA-ML, AoE-ML (Machine Learning approach, Section 2.2). For the baselines N individuals of each dataset are enrolled into M random groups but for JLAR the algorithm learns how to partition the enrolled templates.

Face Datasets

We use LFW and CFP datasets that introduced in Section 2.2.2. Here again, face descriptors are obtained from a pre-trained network based on VGG-Face architecture [PVZ+15] followed by PCA and then L_2-normalization with $d = 1,024$.

IRIS Datasets

Iris images are prepossessed by the following steps: iris localization, iris normalization and image enhancement. Then the feature vectors are extracted by Gabor filters.

CASIA-IrisV1 [oSIoA]. The database includes 756 iris images from 108 eyes of Chinese persons. The images stored in the database were captured within a highly constrained capturing environment. 3 images were collected in a first session and 4 images in a second session. The database is created by randomly sampling $N = 80$ individuals to be enrolled, and $N_q = 28$ impostors.

MMU2 [Uni]. This dataset contains 995 images corresponding to 100 people with different age and nationality from Asia, Middle East, Africa and Europe. Each of them contributes to 5 iris images for each eye. We exclude 5 left eye iris images due to cataract disease.

Group Verification

A user claims she/he belongs to group g. This claim is true under hypothesis $\mathcal{H}_1$ and false under hypothesis $\mathcal{H}_0$ (*i.e.* the user is an impostor). Her/his signature $\mathbf{q}$ is embedded into $\mathbf{p} = \mathsf{e}(\mathbf{q})$, and $(\mathbf{p}, g)$ is sent to the system, which compares $\mathbf{p}$ to the group representation $\mathbf{r}_g$. The system accepts ($t = 1$) or rejects ($t = 0$) the claim. This is a two hypothesis test with two probabilities of errors: $P_{\mathsf{fp}} := \mathbb{P}(t = 1|\mathcal{H}_0)$ is the false positive rate and $P_{\mathsf{fn}} := \mathbb{P}(t = 0|\mathcal{H}_1)$ is the false negative rate. The figure of merit is P_{fn} when $P_{\mathsf{fp}} = 0.05$.

Figure 2.14 compares the performance of JLAR with baselines for group membership verification. Overall, JLAR gives a better verification performance, especially on CASIA. Since our method tries to simultaneously learn group representations and assignment, it aggregates similar embedded vectors and this looses less information.

Note that, although LFW and CFP are difficult datasets due to the "in-the-wild" variations, the group membership verification task is handled well even for large group sizes. This is not the case for iris datasets. As mentioned before, we make use of VGG-Face for face datasets while for iris, traditional feature extraction algorithms are used. So, the big difference in overall analysis shows how the feature space affect the performance

of group membership tasks.

Group Identification

The scenario is an open set identification where the querying user is either enrolled or an impostor. The system proceeds in two steps. First, it decides whether or not this user is enrolled. This is verification as above, except that the group is unknow: The system computes $\delta_j = \|\mathbf{p} - \mathbf{r}_j\|$, $\forall j \in [M]$, and accepts ($t = 1$) if the minimum of these M distances is below a given threshold τ. The figure of merit is P_{fn} when $P_{\mathsf{fp}} = 0.05$.

When $t = 1$, the system proceeds to the second step. The estimated group is given by $\hat{g} = \arg\min_{j\in[M]} \delta_j$. The figure of merit for this second step is $P_\epsilon := \mathbb{P}(\hat{g} \neq g)$ or the Detection and Identification Rate $DIR := (1 - P_\epsilon)(1 - P_{\mathsf{fn}})$.

Figure 2.15 shows that JLAR brings improvement compared to the baselines and the improvement is also better as the size of groups increases.

The impact of the group size on DIR is illustrated in Figure 2.16. Obviously, packing more signatures into one group representation is detrimental. It gets worse when the queries are not well correlated with the enrolled signature.

Security and Privacy Analysis

A curious server can only reconstruct a single vector $\hat{\mathbf{r}}_g = \mathsf{rec}(\mathbf{r}_g)$ from the group representation $\mathbf{r}_g$, and this vector serves as an estimation of any signature in the group. We measure the security by 2.2 which is the mean square error over the dataset. Also, for the privacy of the query template, a curious server can reconstruct the query template $\mathbf{q}$ from its embedding as given in Equation 2.1. These reconstructions are possible only if matrix $\mathbf{W}$ is known. This is not the case in practice, so we give here an extra advantage to the curious server.

Figure 2.17 compares security with AoE-ML, where the assignment was imposed randomly, *i.e.* not learned. Different levels of sparsity are tested. Performance is represented on the horizontal axis by P_{fn}, and security and privacy are measured on the vertical axis by MSE. For security, we measure the server's ability to reconstruct enrolled signatures

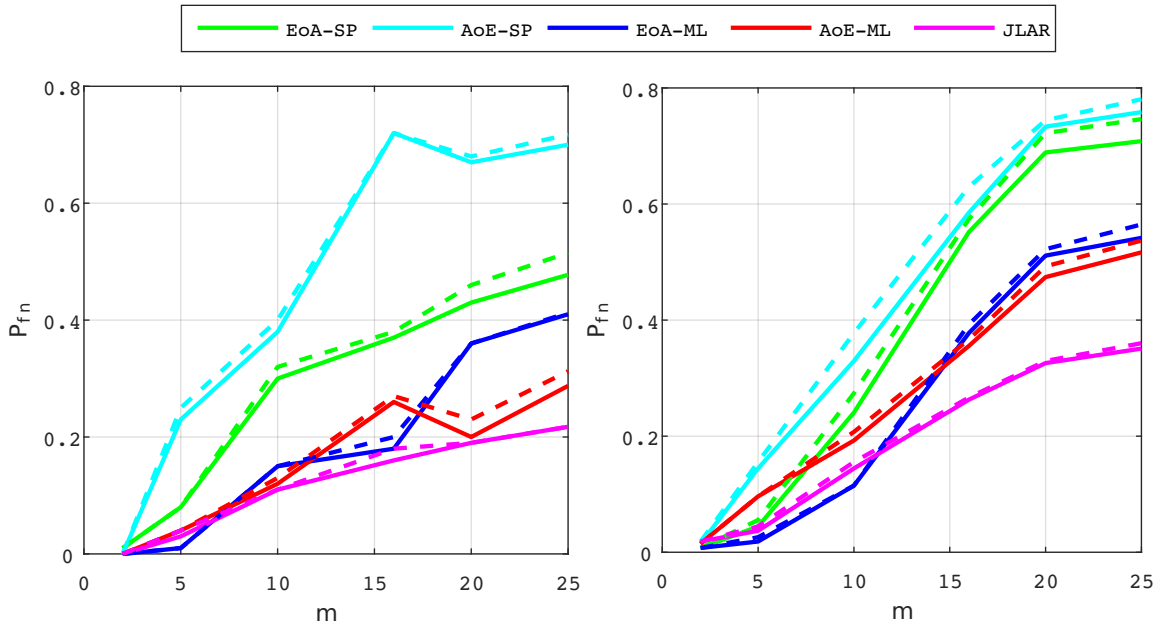


Figure 2.15 – Performances comparison for varying group size m on group identification for CFP(left) and LFW(right). P_{fn} at $P_{\text{fp}} = 0.05$ for the first step of group identification (solid) and P_ϵ for the second step of group identification (dashed).

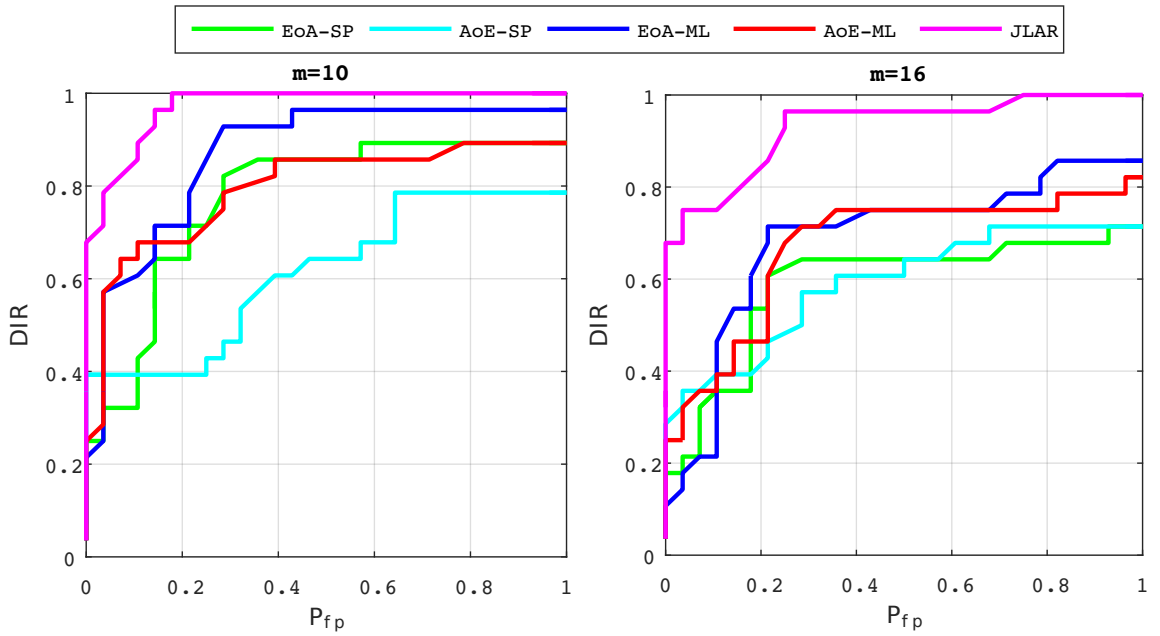


Figure 2.16 – The Detection and Identification Rate (DIR) vs. P_{fp} for group identification on CASIA-IRISV1.

from group representations, and for privacy, we measure the server's ability to reconstruct the query from its embedding.

The reconstruction error of queries are close in either case, yet learning the assignment improves verification performance. Reconstructing enrolled signatures is more difficult due to the aggregation. However, learning the assignment by similarity correspondence in the embedded domain decreases the security slightly while improving the performance a lot.

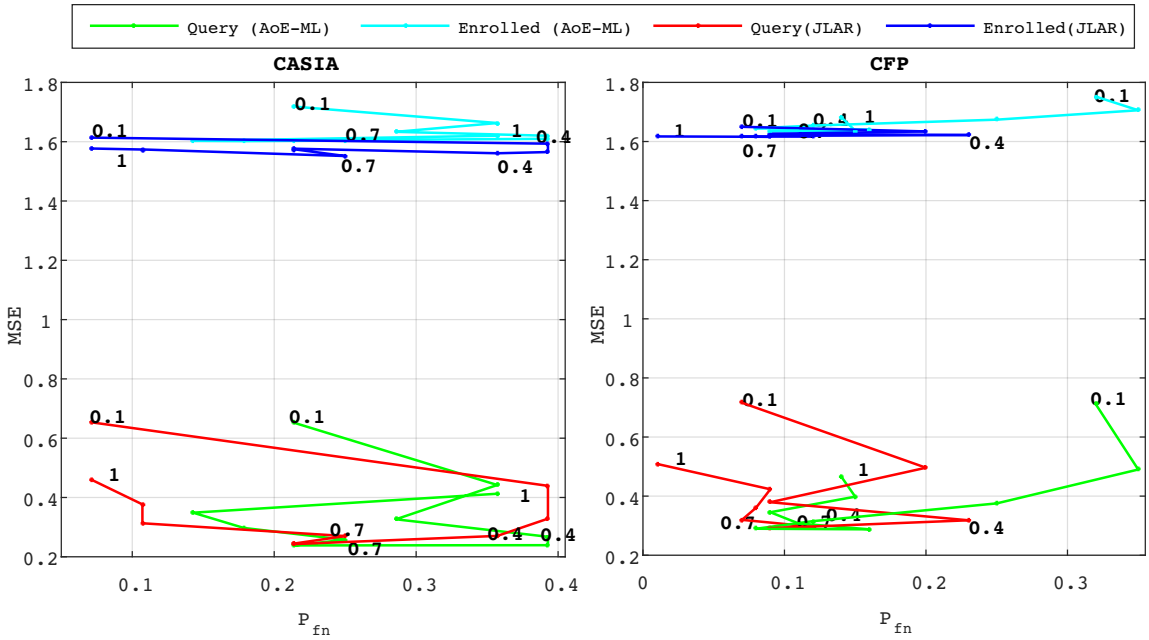


Figure 2.17 – Investigation of trade-off between security and performance for varying sparsity level S on CFP (with $m = 25$) and CASIA-IrisV1 (with $m = 16$).

2.3.4 Security Protocols

This section gives an example of a cryptographic protocol exploiting the group representations. The experimental section showed that grouping secures the enrolled signatures, but ternarization alone provides less protection to the query. Therefore, this protocol strengthens the protection of the querying user. For security reason, the server only manipulates query and the distances in the encrypted domain. For privacy reason, the server only learns that the query is close enough to one group representation, but it cannot tell which group exactly. We assume honest but curious user and server.

This protocol also justifies choices of our scheme: Queries and group representations are heavily quantized onto a small alphabet $\mathcal{A}$. They are long vectors but sparse: only S components will be processed in the encrypted domain. Moreover, we have $\|\mathbf{p}-\mathbf{r}\|^2 \in [0, 2S]$. These facts ease the use of partial homomorphic encryptions with limited module, whence a low complexity and expansion factor. The group representations remain in the clear on the server side, and we do not need fully homomorphic encryption.

The user generates a pair of secret and public keys (sk_U, pk_U) for an additive homomorphic cryptosystem $e(\cdot)$ (say [Pai99b]), and sends the query encrypted component-wise. The server computes its correlation with group representation $\mathbf{r}_g$:

$$e(\mathbf{p}^\top \mathbf{r}_g, pk_U) = \prod_{i:r_g(i)\neq 0} e(p(i), pk_U)^{r_g(i)}. \tag{2.45}$$

The server also generates a key pair (sk_S, pk_S) for a multiplicative homomorphic cryptosystem $E(\cdot)$ (say [Elg85]), and sends the user $(E(e(\mathbf{p}^\top \mathbf{r}_g, pk_U), pk_S))_g$. The user randomly permutes the order of these quantities and masks them by multiplying them by $E(1, pk_S)$. This yields another semantically secure version of the ciphertexts thanks to the multiplicative homomorphy of $E(\cdot)$. The server decrypts $(e(\mathbf{p}^\top \mathbf{r}_k, pk_U))_k$, but the permutation prevents connecting k back to the group index g. Again thanks to homomorphy, the server computes $(e(a_k(2S - 2\mathbf{p}^\top \mathbf{r}_k - \tau) + b_k), pk_U))_g$ where (a_k, b_k) are random signed integers. The user decrypts and sends $(a_k(\|\mathbf{p}-\mathbf{r}_k\|^2 - \tau) + b_k)_k$ to the server. The user cannot guess the distances $\|\mathbf{p}-\mathbf{r}_k\|^2$ thanks to the masking $(a_k, b_k)_k$, not even the sign of $(\|\mathbf{p}-\mathbf{r}_k\|^2 - \tau)$. The server can do this (since it knows (a_k, b_k)) and thus learns whether there is one group where $(\|\mathbf{p}-\mathbf{r}_k\|^2 - \tau)$ is negative.

2.4 Conclusion

This chapter proposed schemes for group membership verification and identification. The keystones are the aggregation and embedding functions. They prevent accurate reconstruction of the enrolled signatures while recognizing noisy version. The server is only able to link each signature to its group number. Yet, the full identity of the user is preserved.

The first proposed scheme is based on defining fixed aggregation and embedding func-

tions. Then we replaced those hand-crafted functions and their hard-coded parameters with functions producing identical types of output, but their nature and parameters are learned through optimization processes, learning provides a serious performance improvement over fixed settings.

After that, in addition to group representation, group assignments are also learned. Instead of random assignment, the groups formed based on similarity. The idea is to minimize the overall distance between group members while maximizing the separation between groups in the embedded domain, which provides an additional improvement.

However, this learning was not completely free. Some guidances were still imposed, especially the prototyping of the embedding based on a sparse ternary quantization. This is mainly for inheriting the security and privacy properties of this lossy information processing. It is not clear whether an alternative approach does exist.

Chapter 3

Sparse or Dense

Introduction

The approaches proposed in the previous chapter face severe limitations. Basically, it seems impossible to create features representing groups having many members. In this case, the probability to identify true positives vanishes and the false negative rate grows accordingly. Furthermore, the robustness of the matching procedure fades and becomes unable to absorb even the smallest amount of noise that inherently differentiate the enrolled template of one member and the template captured at query time for this same member. In contrast, features representing only few group members are robust to noise and cause almost no false negatives. It seems that these limitations originate from the sparsity level of the features representing group members.

This chapter investigates the impact of the sparsity level of the high dimensional features representing group members on the quality of (true positive) matches and their robustness to noise. We consider two setups: "sparse" and "dense," which refer to the number of zero vs. non-zero elements in a sequence. When a sequence is sparse, it contains mostly zeros and few non-zero elements, whereas sequences in dense settings contain mostly non-zero elements. This study shows that it is possible to trade compactness and sparsity for better security or better verification performance.

Section 3.1 first considers the aggregation of discrete random sequences, and models this trade-off with information theoretical tools. Section 3.2 applies this viewpoint to binary random sequences and shows that the noise on the query has an impact depending upon the sparsity of the sequences. Section. 3.3 bridges the gap between the templates, *i.e.* real d-dimensional vectors, and the discrete sequences considered in the previous sections. Section. 3.4 gathers the experimental results for a group membership verification based on faces. Section 3.5 explores some studies on ternary sequences.

3.1 Discrete Sequences

This section considers the problem of creating a representation $\mathbf{Y}$ of a group of n sequences $\{\mathbf{X}_1, \ldots, \mathbf{X}_n\}$. The use of $\mathbf{Y}$ is to test whether a query sequence $\mathbf{Q}$ is a noisy version of one of these n original sequences. This test is done at query time when the original sequences are no longer available and all that remains is the representation $\mathbf{Y}$.

The sequences are elements of $\mathcal{X}^m$ where $\mathcal{X}$ is a finite alphabet of cardinality $|\mathcal{X}|$, say $\mathcal{X} := \{0, 1, \ldots, |\mathcal{X}| - 1\}$. The sequence follows a statistical model giving a central role to the symbol 0. The symbols of the sequences are independent and identically distributed with

$$\mathbb{P}(X = s) = \begin{cases} 1 - p(|\mathcal{X}| - 1) & \text{if } s = 0 \\ p & \text{otherwise} \end{cases} \tag{3.1}$$

for $p \in (0, 1/|\mathcal{X}|]$. Sparsity means that probability p is small. The opposite, density means that p is close to $1/|\mathcal{X}|$ so that X is uniformly distributed over $\mathcal{X}$.

3.1.1 Structure of the group representation

We impose the following conditions on the aggregation $\mathsf{a}(\cdot)$ computing the group representation $\mathbf{Y} = \mathsf{a}(\mathbf{X}_1, \ldots, \mathbf{X}_n)$:

- $\mathbf{Y}$ is a discrete sequence of the same length $\mathbf{Y} \in \mathcal{Y}^m$,
- Symbol $Y(i)$ only depends on symbols $\{X_1(i), \ldots, X_n(i)\}$,
- The same aggregation is made index-wise: with abuse of notation $Y(i) = \mathsf{a}(X_1(i), \ldots, X_n(i))$, $\forall i \in [m]$,
- $Y(i)$ does not depend on any ordering of the set $\{X_1(i), \ldots, X_n(i)\}$,

These requirements are well known in traitor tracing and group testing as they usually model the collusion attack or the test results over groups. Here, they simplify the analysis reducing the problem to a single letter formulation where index i is dropped involving symbols $\{X_1, \ldots, X_n\}$, Y and Q.

These conditions motivate a 2-stage construction. The first stage computes the type (a.k.a. histogram or tally) T of the symbols $\{X_1, \ldots, X_n\}$. Denote by $\mathcal{T}_{|\mathcal{X}|,n}$ the set of

possible type values. Its cardinality equals $|\mathcal{T}_{|\mathcal{X}|,n}| = \binom{n+|\mathcal{X}|-1}{|\mathcal{X}|-1}$ which might be too big. The second stage applies a surjective function $\mathsf{r} : \mathcal{T}_{|\mathcal{X}|,n} \to \mathcal{Y}$, where $\mathcal{Y}$ is a much smaller set.

3.1.2 Noisy query

At enrollment time, the system receives n sequences, aggregates them into the compact representation $\mathbf{Y}$, and then forgets the n sequences. At query time, the system receives a new sequence $\mathbf{Q}$ conforming with one of the following hypotheses:

- $\mathcal{H}_1$: $\mathbf{Q}$ is a noisy version of one of the enrolled sequences. Without loss of generality, $\mathbf{Q} = \mathbf{X}_1 + \mathbf{N}$.
- $\mathcal{H}_0$: $\mathbf{Q} = \mathbf{X}_0 + \mathbf{N}$, where $\mathbf{X}_0$ shares the same statistical model but it is independent of $\{\mathbf{X}_1, \ldots, \mathbf{X}_n\}$.

We model the source of noise (due to different acquisition conditions) by a discrete communication channel. It is defined by function $\mathsf{W} : \mathcal{X} \times \mathcal{X} \to [0, 1]$ with $\mathsf{W}(q|x) := \mathbb{P}(Q = q|X = x)$. We impose some symmetry w.r.t. the symbol 0: $\mathsf{W}(s|0) = \eta_0$ and $\mathsf{W}(0|s) = \eta_1$, $\forall s \in \mathcal{X}\backslash\{0\}$.

At query time, the system computes a score $S = \mathsf{s}(\mathbf{Q}, \mathbf{Y})$ and compares to a threshold: hypothesis $\mathcal{H}_1$ is deemed true if $S \geq \tau$. This test leads to two probabilities of error:

- $P_{\mathsf{fp}}(n, m)$ is the probability of false positive: $P_{\mathsf{fp}}(n, m) := \mathbb{P}(S \geq \tau|\mathcal{H}_0)$.
- $P_{\mathsf{fn}}(n, m)$ is the probability of false negative: $P_{\mathsf{fn}}(n, m) := \mathbb{P}(S < \tau|\mathcal{H}_1)$.

The emphasis on (n, m) is natural. It is expected that: i) the more sequences are aggregated, the less reliable the test is, ii) the longer the sequences are, the more reliable the test is.

3.1.3 Figures of merit $(\mathsf{C}, \mathsf{S}, \mathsf{V})$

The section presents three information theoretic quantities (expressed in nats) measuring the performances of the scheme. The first two depends on the statistical model of X (especially p) and the aggregation mechanism a. The last one depends moreover on the channel.

Compactness C

The compactness of the group representation is measured by the entropy $\mathsf{C} := H(Y)$. It roughly means that the number of typical sequences $\mathbf{Y}$ scales exponentially as $e^{mH(Y)}$, which can be theoretically compressed to the rate of $H(Y)$ nats per symbol; obviously, the smaller, the better.

Security S

We consider an insider aiming at disclosing one of the n enrolled sequences. Observing the group representation $\mathbf{Y}$, its uncertainty is measured by the equivocation $\mathsf{S} := H(X|Y)$. This means that the insider does not know which of the $e^{mH(X|Y)}$ typical sequences the enrolled sequences are. Here, the larger the better.

Verification V

In our application, the requirement of utmost importance is to have a very small probability of false positive. We are interested in an asymptotical setup where $m \to +\infty$. This motivates the use of the false positive error exponent as a figure of merit:

$$E_{\mathsf{fp}}(n) := \lim_{m \to +\infty} -\frac{1}{m} \log P_{\mathsf{fp}}(n, m). \tag{3.2}$$

If $E_{\mathsf{fp}}(n) > 0$, it means that $P_{\mathsf{fp}}(n, m)$ exponentially vanishes as m becomes larger. The theory of test hypothesis shows that $E_{\mathsf{fp}}(n)$ is upper bounded by the mutual information $\mathsf{V} := I(Y; Q)$ where Q is a symbol of the query sequence, *i.e.* a noisy version of X_1. It means that the necessary length for achieving the requirement $P_{\mathsf{fp}}(n, m) < \epsilon$ is, according to [Sha59]:

$$m \geq \frac{-\log \epsilon}{\mathsf{V}}. \tag{3.3}$$

Therefore, the larger the quantity the better it is.

3.1.4 Noiseless setup

The bigger V and S, the better the performance in terms of verification and security. Yet, they can not be both big at the same time. The noiseless case when the channel

introduces no error and $Q = X$ simply illustrates the trade-off:

$$\mathsf{V} \leq \mathsf{C} \tag{3.4}$$
$$\mathsf{V} + \mathsf{S} = H(X), \tag{3.5}$$

with $H(X) = -\log p_0 + (1-p_0)\log\frac{p}{p_0}$ and $p_0 := \mathbb{P}(X=0)$ (3.1). For a given $|\mathcal{X}|$, $H(X)$ is maximised by the dense solution: $H(X) \leq \log|\mathcal{X}|$ with equality for $p = 1/|\mathcal{X}|$.

3.2 Binary alphabet

This section explores the binary case where $\mathcal{X} = \{0,1\}$. We first set the surjection as the identity function s.t. $Y = T$. Then, the impact of the surjection is investigated.

3.2.1 Working with types

In the binary case, there are $n+1$ type values. There can be uniquely labelled by the number of symbols '1' in $\{X_1, \ldots, X_n\}$, *i.e.* $T = \sum_{i=1}^{n} X_i \sim \mathcal{B}(n,p)$.

Verification

In the noiseless case, after some rewriting:

$$\mathsf{V} = \mathsf{h}(p) - \sum_{t=0}^{n} \mathbb{P}(T=t)\mathsf{h}\left(\frac{t}{n}\right), \tag{3.6}$$

with $\mathsf{h}(p) := -p\log(p) - (1-p)\log(1-p)$, the entropy of a Bernoulli r.v. $\mathcal{B}(p)$. In dense setup that $p = 1/2$ and n is large:

$$\mathsf{V} = \frac{1}{2n} + o\left(\frac{1}{n}\right). \tag{3.7}$$

This is not the maximum of this quantity. For large n, the best option is to set

$$p = \frac{\alpha}{n}, \quad \mathsf{V} = \frac{\beta}{n} + o\left(\frac{1}{n}\right), \tag{3.8}$$

with $\alpha = 1.338$ and $\beta = 0.580$, which is a sparse setup. This was proven in the totally different application of traitor tracing [Laa15, Prop. 3.8].

This section outlines two setups: the dense setup where $p = 1/2$, and the sparse setup where $p = \frac{\alpha}{n}$ goes to 0 when more sequences are packed in the group representation. Both setups share the asymptotical property that $\mathsf{V} \approx \kappa/n$ for large n. According to (3.3), *i.e.* $m \geq \frac{-n \log \epsilon}{\kappa}$, we can pack a big number n of sequences into one group representation provided that their length m scales proportionally to n.

Compactness

The figure of merit for compactness for types is just $\mathsf{C} = H(T)$ where T follows a binomial distribution: $T \sim \mathcal{B}(n, p)$. In the dense setup $p = 1/2$, the binomial distribution is approximated by a Gaussian distribution $\mathcal{N}(n/2; n/4)$ providing:

$$\mathsf{C} = \frac{1}{2} \log \left(\frac{\pi e n}{2} \right) + O\left(\frac{1}{n} \right). \tag{3.9}$$

In the sparse setup $p = \alpha/n$, the binomial distribution is approximated by a Poisson distribution $\mathcal{P}(\alpha)$ [Boe88]:

$$\mathsf{C} \approx \alpha(1 - \log(\alpha)) + e^{-\alpha} \sum_{j=0}^{+\infty} \frac{\alpha^j \log(j!)}{j!}. \tag{3.10}$$

This shows that the types are not compact in the dense setup, and the compactness increases with n, while it approximatively remains constant in the sparse setup.

Security

Thanks to (3.5), we only need to calculate $H(X) = \mathsf{h}(p)$. In the dense setup, $H(X) = \log(2)$ and S converges to $H(X)$ as n increases. Merging into a single representation protects an individual sequence. If sparse,

$$H(X) = \frac{\alpha}{n} \left(1 - \log \frac{\alpha}{n} \right) + o\left(\frac{1}{n} \right). \tag{3.11}$$

Therefore, S converges to zero as n increases, contrary to the dense setup. It might be more insightful to see that the ratio of uncertainties before and after observing T, *i.e.* $H(X)/H(X|T)$, converges to 1 in both cases. Merging does provide some security but sparsity is more detrimental.

3.2.2 Adding a surjection

The motivation of the surjection onto a smaller set $\mathcal{Y}$ is to bound C as $\mathsf{C} \leq \log|\mathcal{Y}|$, $\forall n$. The Markov chain $Q \to X_1 \to T \to Y$ imposes that $\mathsf{V} := I(Y;Q) \leq I(T;Q)$.

Let us first explain how V is computed. Denote $P_i(q,y) := \mathbb{P}(Q=q, Y=y|\mathcal{H}_i)$ and channel $W(q|x) := \mathbb{P}(Q=q|X=x)$, $\forall y \in \mathcal{Y}, q \in \mathcal{X}$ and $i \in \{0,1\}$. Then,

$$\mathsf{V} = \sum_{q,y} P_1(q,y) \log \frac{P_1(q,y)}{P_0(q,y)}, \tag{3.12}$$

with $P_0(q,y) = \mathbb{P}(Q=q)\mathbb{P}(Y=y)$, since under $\mathcal{H}_0$, the variables Q and Y are independent and

$$P_1(q,y) = \sum_{x \in \mathcal{X}} \mathbb{P}(Y=y, X=x) W(q|x). \tag{3.13}$$

We assume here the noiseless setup allowing to write $\mathbb{P}(Y=y, X=x)$ as $P_1(x,y)$. Inspired by traitor tracing, we consider a probabilistic surjection where $\mathbb{P}(\mathsf{r}(t)=1) = \theta_t$. The vector $\boldsymbol{\theta} \in [0,1]^{n+1}$ parametrizes the surjection. Denote by $\nabla_{\boldsymbol{\theta}}\mathsf{V}(t)$ the derivative w.r.t. θ_t. After some lengthy calculus:

$$\begin{aligned}
\nabla_{\boldsymbol{\theta}}\mathsf{V}(t) &= n^{-1} K_1(p,\boldsymbol{\theta})(t - nK_2(p,\boldsymbol{\theta})), \\
K_1(p,\boldsymbol{\theta}) &= \mathbb{P}(T=t)\Delta, \\
K_2(p,\boldsymbol{\theta}) &= \frac{\mathsf{h}'(P_1(0,1)) - \mathsf{h}'(\mathbb{P}(Y=1))}{\Delta}, \\
\Delta &= \mathsf{h}'(P_1(0,1)) - \mathsf{h}'(P_1(1,1)).
\end{aligned} \tag{3.14}$$

It is not possible to cancel the gradient $\nabla_{\boldsymbol{\theta}}\mathsf{V}$. The optimal $\boldsymbol{\theta}$ thus lies on the boundary of the hypercube $[0,1]^{n+1}$. This makes the surjection deterministic. Assuming $\mathbb{P}(Y=1|X=0) < \mathbb{P}(Y=1|X=1)$, then $0 < K_1(p,\boldsymbol{\theta})$ and $0 < K_2(p,\boldsymbol{\theta}) \leq 1$ because $\mathsf{h}'(\cdot)$ is strictly decreasing. This makes $\nabla_{\boldsymbol{\theta}}\mathsf{V}(0) < 0$ and θ_0 must be set to the lowest possible value, *i.e.* $\theta_0 = 0$, to increase V at most. This is indeed the case for any θ_t with $t < K_2(p,\boldsymbol{\theta})$. In the same way, $\theta_n = 1$ and so is θ_t if $t > K_2(p,\boldsymbol{\theta})$. Yet, for a given $\boldsymbol{\theta}$, $K_2(p,\boldsymbol{\theta})$ ranges from 0 to 1 as p increases from 0 to 1. Therefore, $\boldsymbol{\theta} = (0,\ldots,0,1,\ldots,1)$ is optimal only over an interval of p.

Two examples are the following:

- For n odd and $p = 1/2$, $\theta_t = 0$ if $t \leq (n+1)/2$, and 1 otherwise is optimal because $K_2(1/2, \boldsymbol{\theta}) = 1/2$ ($\mathbb{P}(Y = 1) = 1/2$ and $P_1(0,1) = 1 - P_1(1,1)$).
- The surjection $\boldsymbol{\theta} = (0, 1, \ldots, 1)$ makes $P_1(1,1) = 1$ so that $\nabla_{\boldsymbol{\theta}} \mathsf{V}(t) = +\infty$ if $t > 0$ and < 0 for $t = 0$.

Therefore it shows that for $|\mathcal{Y}| = 2$, this loss is minimized for:

$$\mathsf{r}(t) = \begin{cases} 0 & \text{if } t < t_p \\ 1 & \text{otherwise} \end{cases} \tag{3.15}$$

where t_p is a threshold depending on p. In the dense setup, $t_p = (n+1)/2$ and the surjection corresponds to a majority vote collusion in traitor tracing (or a threshold model in group testing). Hence, by [Laa15, Prop. 3.4]:

$$\mathsf{V} = \frac{1}{n\pi} + o\left(\frac{1}{n}\right). \tag{3.16}$$

In the sparse setup $t_p = 1$ which corresponds to an 'All-1' attack in traitor tracing (or the perfect model in group testing). Then the best option is to set $p = \log(2)/n$ and [Laa15, Prop. 3.3]:

$$\mathsf{V} = \frac{(\log(2))^2}{n} + o\left(\frac{1}{n}\right). \tag{3.17}$$

From (3.3), the necessary length is $m \geq -n \log(\epsilon)/(\log(2))^2$.

Figure 3.1 illustrates the trade-off between security, verification, and compactness for noiseless setup. It shows that the surjection provokes a loss in verification. In this figure, the above results are summarized by triangle and stars. The blue and green triangles correspond to dense setup (3.7), (3.16), and the blue, red stars correspond to sparse setup (3.8), (3.17). The main property $\mathsf{V} \approx \kappa/n$ still holds but the surjection lowers κ from 0.5 to 0.32 (dense), from 0.58 to 0.48 (sparse). The sparse setup is still the best option w.r.t. V.

3.2.3 Relationship with the Bloom filter

A Bloom filter is a well-known data structure $\mathbf{Y} \in \{0,1\}^m$ designed for set membership, embedding items to be enrolled into $\mathbf{Y}$ thanks to k hash functions. Its probability of false

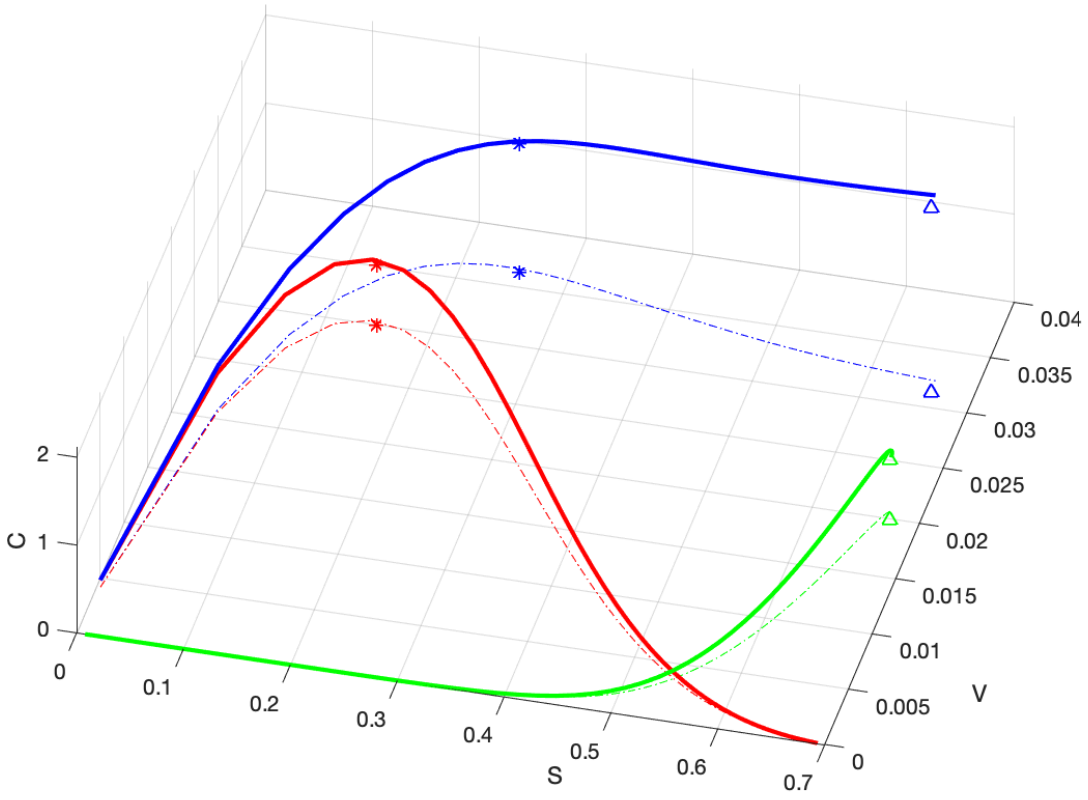


Figure 3.1 – The trade-off $(\mathsf{S}, \mathsf{V}, \mathsf{C})$ for $\mathcal{X} = \{0, 1\}$, $n = 16$, $Y = T$ (blue), $Y = \mathsf{r}(T)$ for 'All-1' (red) and majority vote (green). Dashed plot represents the projection onto $\mathsf{C} = 0$. Triangles and stars summarize results (3.7) to (3.17).

negative is exactly 0, whereas the probability of false positive is not null. The number of hash functions minimizing $P_{\mathsf{fp}}(n, m)$ is $k = \lfloor \log(2)m/n \rfloor$. Then, the necessary length to meet a required false positive level ϵ is $m \geq -n \log(\epsilon)/(\log(2))^2$.

These numbers show the connection with our scheme (3.17). At the enrollment phase, the hash functions indeed associate to the j-th item a binary sequence $\mathbf{X}_j$ indicating which bits of $\mathbf{Y}$ have to be set. This sequence is indeed sparse with $k/m \approx \log(2)/n$. The necessary length is the same. Indeed, the enrollment phase of a Bloom filter is nothing more than the 'All-1' surjection.

The only difference resides in the statistical model. There is at most k symbols '1' in sequence $\mathbf{X}_j$ whereas, in our model, that follows a binomial distribution $\mathcal{B}(m, p)$. Yet, asymptotically as $m \to \infty$, by some concentration phenomenon, the two models get similar. This explains why we end up with similar optimal parameters. Yet, the Bloom filter only works when the query object is strictly identical to the one enrolled, whereas the next section shows that our scheme is robust to noise.

3.3 Real vectors

This section deals with real vectors: n vectors to be enrolled $\{\vec{x}_1, \ldots, \vec{x}_n\} \subset \mathbb{R}^d$, and the query vector $\vec{q} \in \mathbb{R}^d$. All have unit norm. An embedding mechanism $\mathsf{E} : \mathbb{R}^d \to \mathcal{X}^m$ makes the connection with the previous section. As in [AIL+15], this study models the embedding as a probabilistic function.

3.3.1 Binary embedding

For instance, for $\mathcal{X} = \{0, 1\}$, a popular embedding is:

$$X(i) = [\vec{x}^\top \vec{U}_i > \lambda_x], \forall i \in [m] \tag{3.18}$$

where $\vec{U}_i \overset{i.i.d.}{\sim} \mathcal{N}(\vec{0}_d, I_d)$. This in turn gives i.i.d. Bernoulli symbols $\{X(i)\}$ with $p = 1 - \Phi(\lambda_x)$ if $\|\vec{x}\| = 1$.

As illustrated in Figure 3.2, at the query time, the embedding mechanism uses the

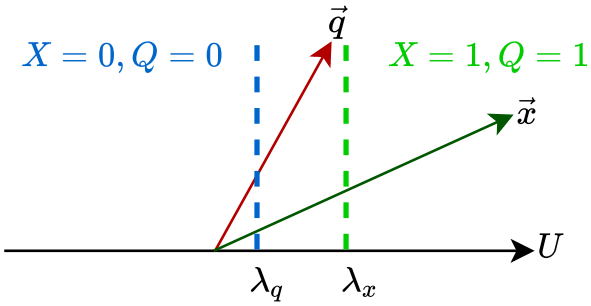


Figure 3.2 – Binary embedding using random projection.

same random vectors but a different threshold:

$$Q(i) = [\vec{q}^\top \vec{U}_i > \lambda_q], \forall i \in [m]. \tag{3.19}$$

Under $\mathcal{H}_1$, suppose that $\vec{q}^\top \vec{x} = c < 1$. This correlation defines the channel $X \to Q$ with the error rates:

$$\eta_0 = \mathbb{P}(Q = 1|X = 0) = \mathbb{P}(\vec{q}^\top \vec{U} > \lambda_q | \vec{x}^\top \vec{U} \leq \lambda_x), \tag{3.20}$$
$$\eta_1 = \mathbb{P}(Q = 0|X = 1) = \mathbb{P}(\vec{q}^\top \vec{U} \leq \lambda_q | \vec{x}^\top \vec{U} > \lambda_x). \tag{3.21}$$

Let $z = \vec{x}^\top \vec{U}$, then $\vec{q}^\top \vec{U}$ can be represented as $cz + \sqrt{1-c^2}\gamma$, where z and γ are i.i.d standard normal with density functions f_z and f_γ. The collision probability $\mathbb{P}(\vec{x}^\top \vec{U} > \lambda_x,\ \vec{q}^\top \vec{U} > \lambda_q)$ is:

$$\mathbb{P}(X = 0, Q = 0) = \int_{-\infty}^{\lambda_x} \int_{-\infty}^{\frac{\lambda_q - cz}{\sqrt{1-c^2}}} f_\gamma(\gamma)\ f_z(z)\ d\gamma\ dz. \tag{3.22}$$

The error rate $\eta_0 = 1 - \frac{\mathbb{P}(X=0,Q=0)}{\mathbb{P}(X=0)}$ is computed as:

$$\eta_0 = 1 - \frac{1}{(1-p)\sqrt{2\pi}} \int_{-\infty}^{\lambda_x} \Phi\left(\frac{\lambda_q - cz}{\sqrt{1-c^2}}\right) e^{-\frac{z^2}{2}} dz. \tag{3.23}$$

and similarly η_1 has the expression:

$$\eta_1 = 1 - \frac{1}{p\sqrt{2\pi}} \int_{\lambda_x}^{\infty} \left[1 - \Phi\left(\frac{\lambda_q - cz}{\sqrt{1-c^2}}\right)\right] e^{-\frac{z^2}{2}} dz. \tag{3.24}$$

3.3.2 Induced channel

For this embedding, the parameters $(\lambda_x, \lambda_q, c, d)$ for the vectors define the setup (p, η_0, η_1) for the sequences. It is a priori difficult to find the best tuning (λ_x, λ_q). For a fixed λ_x, η_0 decreases with λ_q while η_1 increases.

Suppose that η is a parameter of the channel $W(\cdot|\cdot)$. Then

$$\frac{\partial \mathsf{V}}{\partial \eta} = \sum_{q,y} \frac{\partial P_1(q,y)}{\partial \eta} \log \frac{P_1(q,y)}{P_0(q,y)}, \tag{3.25}$$

because $\sum_{q,y} \frac{\partial P_1(q,y)}{\partial \eta} = \frac{\partial \sum_{q,y} P_1(q,y)}{\partial \eta} = 0$ and $\sum_{q,y} \frac{P_1(q,y)}{P_0(q,y)} \frac{\partial P_0(q,y)}{\partial \eta} = \sum_q \frac{\partial \mathbb{P}(Q=q)}{\partial \eta} = 0$.

Suppose now that $\eta = \eta_0 := W(q|0), \forall q \in \mathcal{X}\backslash\{0\}$. Then,

$$\frac{\partial P_1(q,y)}{\partial \eta_0} = \mathbb{P}\left(X = 0, Y = y\right), \forall q \in \mathcal{X}\backslash\{0\}. \tag{3.26}$$

Taking (3.25) around the noiseless channel where $\eta_0 = 0$ and $\mathbb{P}(X = 0, Y = y) = P_1(0, y)$ because $Q = X$:

$$\left.\frac{\partial \mathsf{V}}{\partial \eta_0}\right|_{\eta_0 = 0} = \sum_{y, x \neq 0} P_1(0,y) \log \frac{P_1(x,y)}{P_0(x,y)} + \dots \tag{3.27}$$

We only express the first terms to outline that if $P_1(x, y) = 0$ while $P_1(0, y)$ and hence $P_0(x, y)$ are not null, then this derivative goes to $-\infty$. A small deviation from the noiseless case with $\eta_0 \neq 0$ has a major detrimental impact on V. That situation happens for sure when working with type, *i.e.* $Y = T$: Consider the null type t_0 obtained when $X_1 = \dots = X_n = 0$: $P_1(0, t_0) > 0$ while $P_1(x, t_0) = 0$, $\forall x \neq 0$.

One can prove that the surjection can mitigate this effect if $\exists t \neq t_0 : \mathsf{r}(t) = \mathsf{r}(t_0)$ and $P_1(0, t) > 0$. This happens with the majority vote of the dense setup, but unfortunately, not with of the 'All-1' surjection in the sparse setup. Therefore V is sensitive to η_0 especially with the 'All-1' surjection of the sparse solution.

Figure 3.3 shows indeed that the dense solution, *i.e.* the green line, where $(\lambda_x, \lambda_q) = (0, 0)$) is more robust, unless c is very close to 1. Here, we enforce a surjection (identity, All-1, or majority vote) and make a grid search to find the optimum (λ_x, λ_q) for a given c.

It happens that these parameters are better set to 0, *i.e.* dense solution, for the identity and majority vote. As for the 'All-1' surjection, we observe that λ_x is s.t. $p \approx 1/n$ and λ_q is slightly bigger than λ_x to lower η_0. Yet, this sparse solution is not as good as the dense solution unless c is close to 1, *i.e.* the query vector is very close to the enrolled vector.

This observation holds only for the embedding function (3.18). Hashing functions less prone to error η_0 may exist.

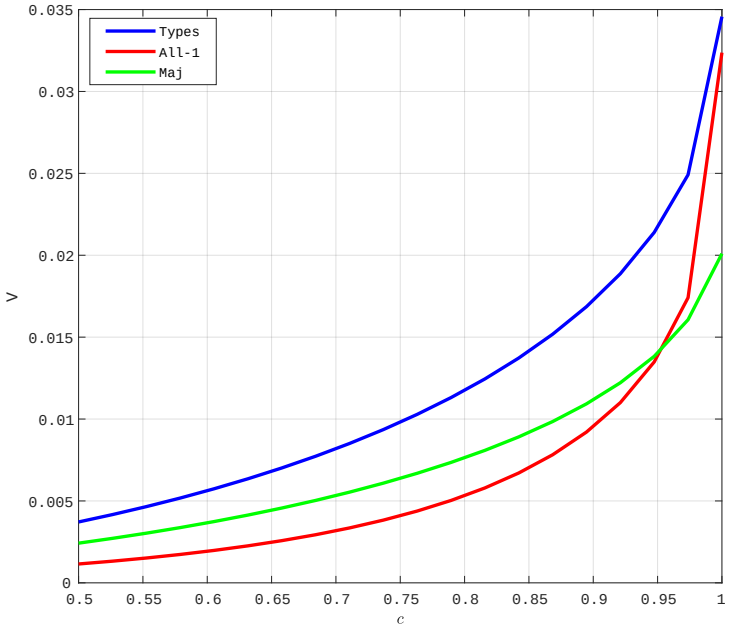


Figure 3.3 – V as a function of correlation c, $d = 256$, $n = 15$.

3.3.3 Score computation

Without surjection, at query time, the system computes a score $S = \mathsf{s}(\mathbf{Q}, \mathbf{T})$ and compares to a threshold. Here the scores are computed based on:

$$\mathsf{s}(\mathbf{Q}, \mathbf{T}) = \sum_{i=1}^{m} W(T(i), Q(i)), \tag{3.28}$$

where each element of the weight matrix $\mathbf{W} \in \mathbb{R}^{|\mathcal{T}| \times |\mathcal{X}|}$ is estimated as:

$$W(T(i), Q(i)) = \log \left(\frac{\mathbb{P}(T = T(i) | Q = Q(i))}{\mathbb{P}(T = T(i))} \right). \tag{3.29}$$

For example in binary case we have:

$$\mathbb{P}(T=t|Q=1)=\mathbb{P}(T=t|X=1)\mathbb{P}(X=1|Q=1)+\mathbb{P}(T=t|X=0)\mathbb{P}(X=0|Q=1) \tag{3.30}$$

where $\mathbb{P}(X=1|Q=1)=\frac{p(1-\eta_1)}{\eta_0(1-p)+p(1-\eta_1)}$ and $\mathbb{P}(T=t|X=1)=\binom{n-1}{t-1}p^{t-1}(1-p)^{n-t}$

3.4 Experimental work

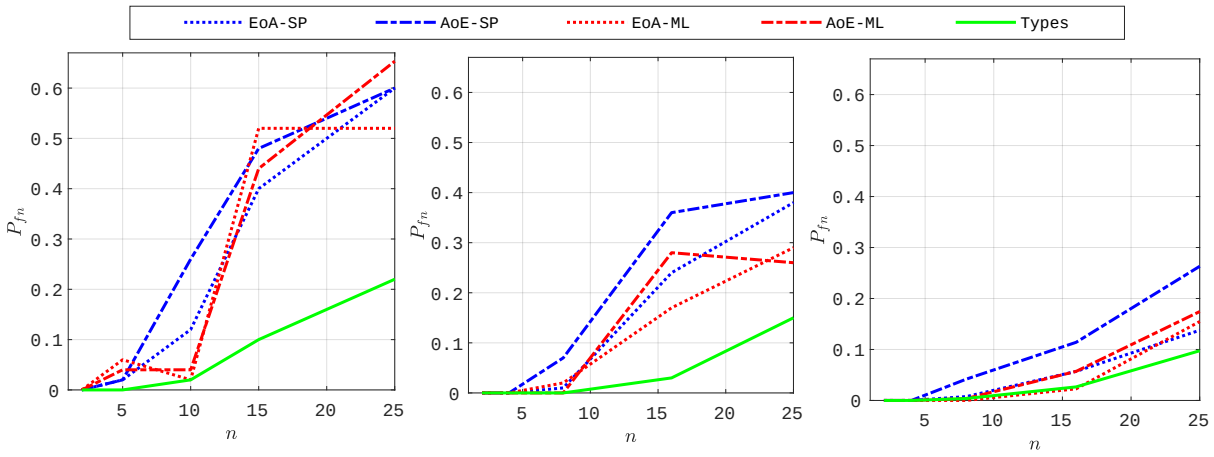


Figure 3.4 – Verification performance $P_{\mathsf{fn}}@P_{\mathsf{fp}}=0.05$ *vs.* group size n for the baselines (see Section 3.4.2) and Types.

We evaluate this scheme with face recognition. Face images are coming from LFW, CFP and FEI databases that introduced in Section 2.2.2. For each dataset, N individuals are enrolled into random groups. There is the same number N_q of positive and negative (impostors) queries.

3.4.1 Experimental Setup

Face descriptors are obtained from a *pre-trained* network based on VGG-Face architecture followed by PCA [PVZ+15] . FEI corresponds to the scenario of employees entering in a building with face recognition, whereas CFP is more difficult, and LFW even more difficult. To equalize the difficulty, we apply a dimension reduction (Probabilistic Principal Component Analysis [TB99]) to $d=128$ (FEI), 256 (CFP), and 512 (LFW). The parameters of PPCA are learned on a different set of images, not on the enrolled templates and queries. The vectors are also L_2 normalized. With such post-processing, the average correlation between positive pairs equals 0.83 (FEI), 0.78 (CFP), and 0.68 (LFW) with a

standard deviation of 0.01. Despite the dimension reduction, the hardest dataset is LFW and the easiest FEI.

In one simulation run, the enrollment phase makes random groups with the same number n of members. A user claims she/he belongs to group g. This claim is true under hypothesis $\mathcal{H}_1$ and false under hypothesis $\mathcal{H}_0$ (*i.e.* the user is an impostor). Her/his template is quantized to the sequence $\mathbf{Q}$, and $(\mathbf{Q}, g)$ is sent to the system, which compares $\mathbf{Q}$ to the group representation $\mathbf{Y}_g$. This is done for all impostors and all queries of enrolled people. One Monte-Carlo simulation is composed of 20 runs. The figure of merit is P_{fn} when $P_{\mathsf{fp}} = 0.05$.

3.4.2 Exp. #1: Comparison to the baselines

The Types scheme is compared to the following baselines:

- EoA-SP and AoE-SP: Signal Processing approach, Section 2.1
- EoA-ML and AoE-ML: Machine Learning approach, Section 2.2

The drawback of these baselines is that the length m of the data structure is bounded. Here, it is set to maximum value, *i.e.* $m = d$ the dimension of templates.

This scheme allows more freedom. Setting $m = 8 \times d$ produces a much bigger representation. It is not surprising that this scheme is better than the baselines. Figure 3.4 validates our motivation to get rid of the drawback of the baselines with limited m, to achieve better verification performance. These results are obtained with the dense solution. Indeed, despite all our efforts, we could not achieve better results with the sparse solution. This confirms the lesson learnt from Figure 3.3: the dense solution outperforms the sparse solution when the average correlation between positive pairs is lower than 0.95.

The improvement is also better as the size of groups increases. We explain this by the use of the types, *i.e.* $Y = T$. Equation (3.9) shows that C increases with n for the dense solution, compensating for aggregating more templates.

3.4.3 Exp. #2: Reducing the size of the group representation

There are two ways for reducing the size of the group representation. The first means is to decrease m, the second means is to lower C thanks to a surjection. Section 3.2.2 presented optimal surjections from $\mathcal{T}_{2,n}$ to $\mathcal{Y} = \{0, 1\}$. We found experimentally good surjections to sets $\mathcal{Y}$ for $|\mathcal{Y}| \in \{3, 4, 8\}$.

This is done according to the following heuristic. Starting from $\mathcal{T}_{2,n}$, we iteratively decrease the size of $\mathcal{Y}$ by one. This amounts to merge two symbols of $\mathcal{Y}$. By brute force, we analyse all the pairs of symbols measuring the loss in V induced by their merging. By merging the best pair, we decrease the number of symbols in $\mathcal{Y}$ by one. This process is iterated until the targeted size of $\mathcal{Y}$ is achieved. This heuristic is not optimal, but it is tractable. Figure 3.5 compares these two means of reducing the size of the group representation. Employing a coarser surjection is slightly better in terms of verification performances.

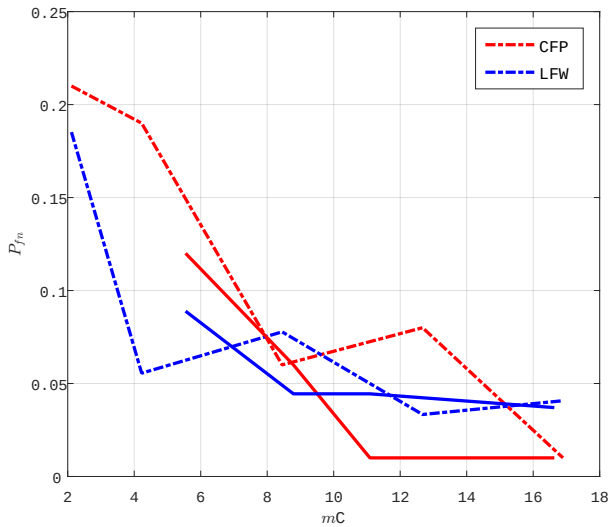


Figure 3.5 – Verification performance $P_{\mathsf{fn}}@P_{\mathsf{fp}} = 0.05$ *vs.* $m\mathsf{C}$, for $n = 16$. This quantity is reduced by decreasing m (dashed lines) or by decreasing C thanks to a surjection (solid lines).

3.5 Ternary Alphabet

The binary alphabet was discussed in Section 3.3, and some results using this binary embedding are presented in Section 3.4. Now, this section investigates the ternary embedding where $\mathcal{X} = \{-1, 0, 1\}$. The symbols are i.i.d. distributed with:

$$\mathbb{P}(X = s) = \begin{cases} 1 - 2p & \text{if } s = 0 \\ p & \text{otherwise} \end{cases} \tag{3.31}$$

for $p \in (0, 1/3]$. Sparsity means that probability p is small, density means that p is close to $1/3$ so that X is uniformly distributed over $\mathcal{X}$. Also, the statistical model for vectors $\vec{x}$ and $\vec{q}$ differs from the previous section. This illustrates the versatility of our framework.

Denote n_p and n_n the number of symbols 1 and -1 in $\{X_1, ..., X_n\}$ respectively. The type values in ternary case can be labeled by $T = \{s, n_n\}$, where $s = n_p + n_n$. Then, s ranges from 0 to n and n_n goes from 0 to s. The cardinality of the set of possible type values equals $\binom{n+2}{2}$. To reduce the size of this set, two surjective function can be applied: sum and majority vote. The sum computes sum over n symbols that ranges in $\{-n, ..., 0, ..., n\}$, while majority vote is the symbol that appears most frequently that ranges in $\{-1, 0, +1\}$.

This embedding is defined as:

$$X(i) = \mathsf{T}_{\lambda_x}(\vec{x}) = sign(\vec{x}^\top \vec{U}_i - \lambda_x)[|\vec{x}^\top \vec{U}_i| > \lambda_x], \forall i \in [m] \tag{3.32}$$

where $\mathbf{U}$ is a matrix whose entries are drawn randomly from independent and identically distributed (i.i.d.) Gaussian distribution *i.e.* $\vec{U}_i \overset{i.i.d.}{\sim} \mathcal{N}(\vec{0}_d, I_d)$. Assume that enrolled vectors $\vec{x}$ are distributed as $\mathcal{N}(\vec{0}_d, \sigma_x^2 \mathbf{I}_d)$. At query time a noisy version of $\vec{x}$ is received as $\vec{q} = \vec{x} + \vec{\omega}$, where $\vec{\omega} \sim \mathcal{N}(\vec{0}_d, \sigma_\omega^2 \mathbf{I}_d)$. Then the same embedding but with different threshold is applied on query:

$$Q(i) = \mathsf{T}_{\lambda_q}(\vec{q}) = sign(\vec{q}^\top \vec{U}_i - \lambda_q)[|\vec{q}^\top \vec{U}_i| > \lambda_q], \forall i \in [m] \tag{3.33}$$

Then, the enrolled and query symbols computed as $X = T_{\lambda_x}(z)$ and $Q = T_{\lambda_q}(z + \gamma)$ respectively, where $z = \vec{x}^\top \vec{U}$ and $\gamma = \vec{\omega}^\top \vec{U}$.

Consider f_z and f_γ as the probability density function of the normal distribution. As the joint probability distribution of X and Q is illustrated in Figure 3.6:

$$P_{0,1} = \int_{-\lambda_x}^{\lambda_x} \int_{\lambda_q - z}^{+\infty} f_z(z) f_\gamma(\gamma) d\gamma dz \tag{3.34}$$

$$P_{1,0} = \int_{\lambda_x}^{+\infty} \int_{-\lambda_q - z}^{\lambda_q - z} f_z(z) f_\gamma(\gamma) d\gamma dz \tag{3.35}$$

$$P_{-1,1} = \int_{-\infty}^{-\lambda_x} \int_{\lambda_q - z}^{+\infty} f_z(z) f_\gamma(\gamma) d\gamma dz \tag{3.36}$$

The error rates of the channel are expressed as:

$$\eta_0 = \mathbb{P}(Q = 1|X = 0) = \frac{1}{(1-2p)\sqrt{2\pi}} \int_{-\lambda_x}^{\lambda_x} \Phi\left(\frac{-\lambda_q - z}{\sigma_\omega}\right) e^{-\frac{z^2}{2}} dz, \tag{3.37}$$

$$\eta_1 = \mathbb{P}(Q = 0|X = 1) = \frac{1}{p\sqrt{2\pi}} \int_{\lambda_x}^{\infty} \left[\Phi\left(\frac{\lambda_q - z}{\sigma_\omega}\right) - \Phi\left(\frac{-\lambda_q - z}{\sigma_\omega}\right)\right] e^{-\frac{z^2}{2}} dz, \tag{3.38}$$

$$\eta_2 = \mathbb{P}(Q = -1|X = 1) = \frac{1}{p\sqrt{2\pi}} \int_{\lambda_x}^{\infty} \Phi\left(\frac{-\lambda_q - z}{\sigma_\omega}\right) e^{-\frac{z^2}{2}} dz. \tag{3.39}$$

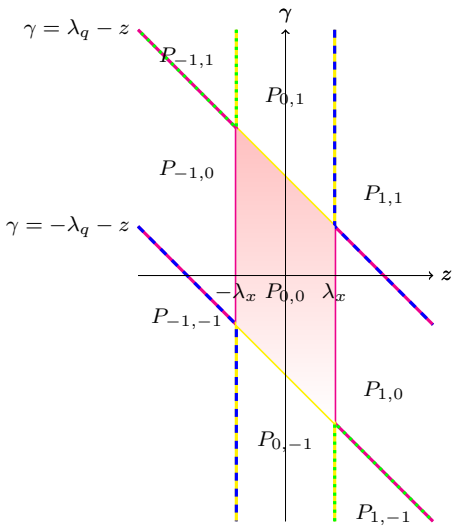


Figure 3.6 – The figure depicts the regions $\mathcal{A}_{x,q}$, where $P_{x,q} = \mathbb{P}(X = x, Q = q) = \int\int_{\mathcal{A}x,q} f_z(z) f_\gamma(\gamma) d\gamma dz$.

From Figure 3.6, following relations can be derived $P_{1,1} = P_{-1,-1}$, $P_{1,0} = P_{-1,0}$ and $P_{1,-1} = P_{-1,1}$. The expected value of squared distances between enrolled and query

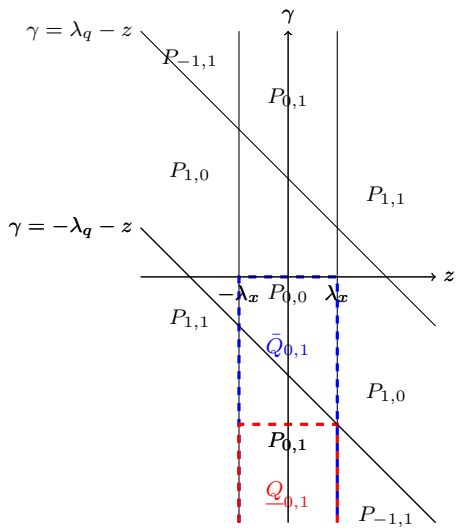


Figure 3.7 – The bounds for $P_{0,1}$

symbols $E[d(X,Q)]$ is computed as:

$$E[||X-Q||_2^2] = \sum_{x,q\in\{-1,0,+1\}} P_{x,q}||x-q||_2^2 = 2P_{0,1} + 2P_{1,0} + 8P_{-1,1} \tag{3.40}$$

Now, we are interested in finding what happens if the amount of noise gets too large, i.e. $\lim_{\sigma_\omega \to \infty} E[||X-Q||_2^2]$. As illustrated in Figure 3.7, the bounds for $P_{0,1}$ is found as $\underline{Q}_{0,1} \leq P_{0,1} \leq \bar{Q}_{0,1}$:

$$\underline{Q}_{0,1} = \int_{-\lambda_x}^{\lambda_x} f_z(z)dz \int_{-\infty}^{-\lambda_q-\lambda_x} f_\gamma(\gamma)d\gamma = \Phi\left(\frac{-\lambda_q-\lambda_x}{\sigma_\omega}\right)\left[\Phi\left(\frac{\lambda_x}{\sigma_x}\right) - \Phi\left(\frac{-\lambda_x}{\sigma_x}\right)\right] \tag{3.41}$$

$$\bar{Q}_{0,1} = \int_{-\lambda_x}^{\lambda_x} f_z(z)dz \int_{-\infty}^{0} f_\gamma(\gamma)d\gamma = \Phi(0)\left[\Phi\left(\frac{\lambda_x}{\sigma_x}\right) - \Phi\left(\frac{-\lambda_x}{\sigma_x}\right)\right], \tag{3.42}$$

By setting t as $max(\lambda_q, k\sqrt{\sigma}_\omega)$, the bounds for $P_{1,0} + P_{-1,1}$ and $P_{-1,1}$ are computed as:

$$\int_{\lambda_x}^{t} f_z(z)dz \int_{-\infty}^{\lambda_q-t} f_\gamma(\gamma)d\gamma \leq P_{1,0} + P_{-1,1} \leq \int_{\lambda_x}^{\infty} f_z(z)dz \int_{-\infty}^{\lambda_q-\lambda_x} f_\gamma(\gamma)d\gamma =$$
$$\Phi\left(\frac{\lambda_q-t}{\sigma_\omega}\right)\left[\Phi\left(\frac{t}{\sigma_x}\right) - \Phi\left(\frac{\lambda_x}{\sigma_x}\right)\right] \leq P_{1,0} + P_{-1,1} \leq \Phi\left(\frac{\lambda_q-\lambda_x}{\sigma_\omega}\right)\left[1 - \Phi\left(\frac{\lambda_x}{\sigma_x}\right)\right] \tag{3.43}$$

$$\int_{\lambda_x}^{t} f_z(z)dz \int_{-\infty}^{-\lambda_q - t} f_\gamma(\gamma)d\gamma \leq P_{-1,1} \leq \int_{\lambda_x}^{\infty} f_z(z)dz \int_{-\infty}^{-\lambda_q - \lambda_x} f_\gamma(\gamma)d\gamma =$$
$$\Phi\left(\frac{-\lambda_q - t}{\sigma_\omega}\right)\left[\Phi\left(\frac{t}{\sigma_x}\right) - \Phi\left(\frac{\lambda_x}{\sigma_x}\right)\right] \leq P_{-1,1} \leq \Phi\left(\frac{-\lambda_q - \lambda_x}{\sigma_\omega}\right)\left[1 - \Phi\left(\frac{\lambda_x}{\sigma_x}\right)\right] \quad (3.44)$$

which then results in $\lim_{\sigma_\omega \to \infty} E[||X - Q||_2^2] = 3 - 2\Phi\left(\frac{\lambda_x}{\sigma_x}\right)$, which increases from 1 to 1.667 as p ranges from 0 (sparse setup) to 1/3 (dense setup).

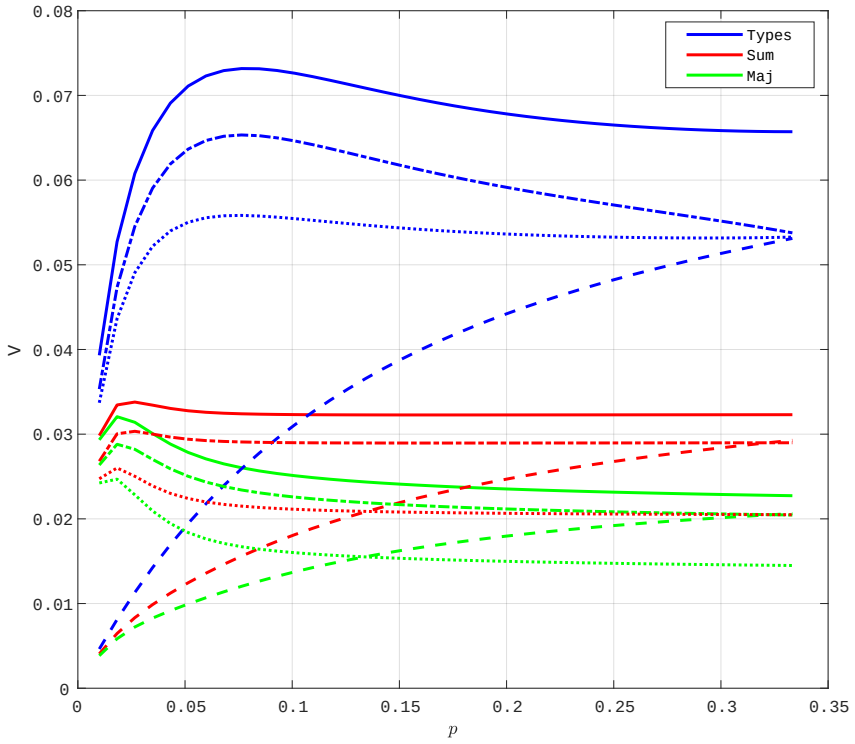


Figure 3.8 – Verification performance V *vs.* p for ternary embedding. $Y = T$ (blue), $Y = \mathsf{r}(T)$ for sum (red) and majority vote (green). Solid line indicates a noiseless case, dashed line indicates $(\eta_0 = 0.1, \eta_1 = \eta_2 = 0)$, dot-dashed line indicates $(\eta_1 = 0.1, \eta_0 = \eta_2 = 0)$, and dotted line indicates $(\eta_2 = 0.1, \eta_0 = \eta_2 = 0)$.

As discussed in Section 3.1.2 verification performance can be measured by the mutual information $\mathsf{V} := I(Y;Q)$. We compute this quantity for varying $p \in (0, 1/3]$ for both noiseless and noisy setups. Figure 3.8 compares verification performance for different aggregation mechanisms.

Obviously applying surjection on type values causes information loss, so the mutual information for sum and majority vote is lower. Similarly, compared to sum, this quantity reduces by applying majority vote function.

Like the binary case, the optimal thing is to pack sequences that are sparse with $p = \frac{\alpha}{n}$. If we want to pack more sequences, we need sparse sequences to have the maximum of mutual information. This figure also shows that the mutual information is more sensitive to η_0 for the sparse setup, so the sparse setup can be optimal when there is no noise. Unlike the binary case, V takes the bigger values in general.

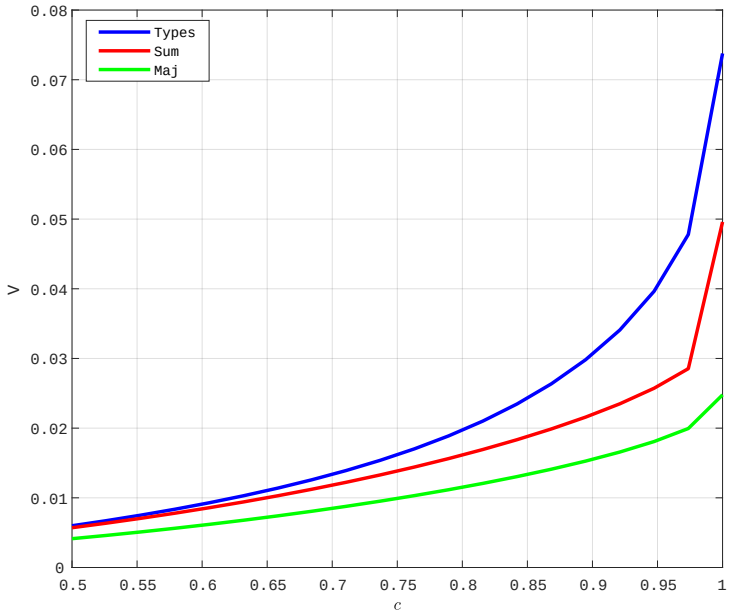


Figure 3.9 – Verification performance V for varying c in Ternary case.

The error probabilities (η_0, η_1, η_2) are defined based on parameters $(\lambda_x, \lambda_q, c)$. For a given c, we can find the best tuning of (λ_x, λ_q). Figure 3.9 shows the mutual information quantity for varying correlation c (σ_ω in this model). We observed that when correlation is lower, these parameters are better set to something like 0.43 to have $p \approx 1/3$, *i.e.* dense setup. Also, the sum outperforms majority vote and the improvement is also better as the the query vector is very close to the enrolled vector.

3.6 Conclusion

Since the sparsity of the embeddings seemingly plays a crucial role in the performance verification. This chapter proposed a mathematical model for group membership verification, allowing to reveal the impact of sparsity on both security, compactness, and

verification performances.

We studied both binary and ternary alphabets. It shows that in terms of verification performance, the sparse setup can be optimal when there is no noise, but η_0 will never be equal to zero in practice. It means that it is not easy to have a sparse solution.

When we are operating in the low-SNR regime where the positive queries are less correlated with the enrolled templates, the dense setup is more interesting in terms of both verification performance V and security level S.

Nevertheless, there are still some shortcomings in this study. First, tuning of λ is challenging because it fixes both p and channel error probability. Second, this observation holds only for this embedding function. We only considered a simple model where we have a symbol 0 with a different probability, while the others are equal. We do not know what the best embedding is. Maybe it is better to learn the embedding model from a training set.

Chapter 4

Design of practical IoT authentication

Regarding the goals of the ID_IOT project, in this chapter, we study experimental data from Optical Physical Unclonable Functions (PUFs) provided by our partner from the University of Eindhoven to design schemes that will enable authentication and group membership verification of PUF data.

Section 4.1 introduces the basics of PUFs. In Section 4.2, we provide an overview of the ID_IOT project and how we handle the data that is available to us. Section 4.3 discusses producing compact representations for Optical PUFs to make them suitable for authentication protocols. In section 4.4, we describe the design of a practical group membership verification scheme by using descriptors provided in Section 4.3.

4.1 PUF Initial definitions

A Physically Unclonable Function (PUF) is a physical object that takes an input challenge and produces the output response in a way that is unique to its physical structure and cannot be reproduced (unclonable). For a particular challenge, no two objects generate the same response. The combination of a challenge and its response is referred to as a Challenge-Response Pair (CRP).

Physically unclonable functions are most commonly used for authentication. PUF-based authentication does not require classical cryptographic assets, making it fitting for IoT devices. The PUF authentication process consists of two phases: enrollment and authentication. Traditional PUF-based authentication protocols work as follows: During the authentication phase, the server applies a randomly chosen challenge to the PUF. The device sends back the PUF response. Then the device is authenticated if the measured

response matches the corresponding stored response in the database. However, many of these lightweight protocols are vulnerable against spoofing attacks. We need to study PUF properties to choose the type of the protocol that optical PUFs can be used for.

Optical PUF

Optical PUF is a scattering medium that scrambles the coherent light propagating inside them and produces apparently random speckle patterns. The propagation of light through this medium is complex and unpredictable. Thus, the output pattern is dependent on the physical properties of the scattering medium.

The input to a PUF is considered as a challenge, and the PUF produces a unique output response. Challenge C_i to an optical PUF consists of a location at which the laser beam directs towards and the incoming laser beam's properties such as the angle of incidence, etc. The corresponding response R_i of the PUF is the raw speckle pattern captured by the CCD camera.

4.2 ID_IOT

The project "IDentification for the Internet Of Things" is a collaboration between the Technical University of Eindhoven, INRIA (France), and the University of Geneva. There will be a huge number of devices with very low or no processing and communication resources, coupled with a small number of high-power devices in the Internet of Things. ID_IOT addresses privacy-preserving algorithms for authentication and identification of huge numbers of low-power devices in the IoT. The essential tasks in the IoT will be to verify if an object is authentic or to identify an object. Weak devices, which are most ubiquitous, cannot rely on cryptography to authenticate themselves.

PUFs are ideal for IoT applications because they enable authentication and identification of physical objects without the need for any cryptography or storage of secret information. ID_IOT intends to address these problems using privacy-preserving database structures and algorithms with good scaling behavior. A brief recap of the project is that the project will contribute to improving both the theory and practice of PUF technologies for IoT authentication and identification scenarios.

The Optical PUF data that we will discuss in the following is prepared by our partner from the Technical University of Eindhoven. We have not been involved in the production of the data. Our task is to design the protocols that will enable authentication, identification, and group membership verification scenarios using the PUF data. The following explores some properties of the PUF data, which points out that an essential requirement is to have short descriptors for the PUF measurement data. Then, we implement authentication and group membership verification schemes for this PUF data using the learned descriptors.

4.2.1 Data Preparation

In our dataset, 20 different Zinc Oxide (ZnO) medium of different thickness are used as PUFs. A Spatial Light Modulator (SLM) is also used to transform the laser light into a random challenge. A camera is placed at the output side, and is only sensitive to the intensity of the output light field. Therefore the speckle images captured by a camera contains information only about the absolute value of the output light field, and the phase information is lost.

We use raw speckle images provided by Ravitej Uppu [UWG+19] from the University of Twente. The raw data is whatever the camera recorded, i.e., the absolute value of the output response. We refer to this dataset as Real-Dataset. The experimental setup is illustrated in Figure 4.1.

Moreover, in another experiment, instead of using a SLM, a scanning mirror is used to challenge the PUF from different angles. Then the complex response, i.e., phase and the intensity of the outgoing light, is measured using a method called off-axis holographic method. We call the complex speckle images gathered in this experiment as Complex-Dataset.

The output speckle pattern produced by the experiments are images of size 600×600. The data collected in this experiment consists of 919 images corresponding to 919 challenges. The experiment measures both horizontal and vertical polarization, and also both types of polarized light are used as input, resulting in four cases: horizontal input and horizontal output, vertical input and vertical output, vertical input and horizontal

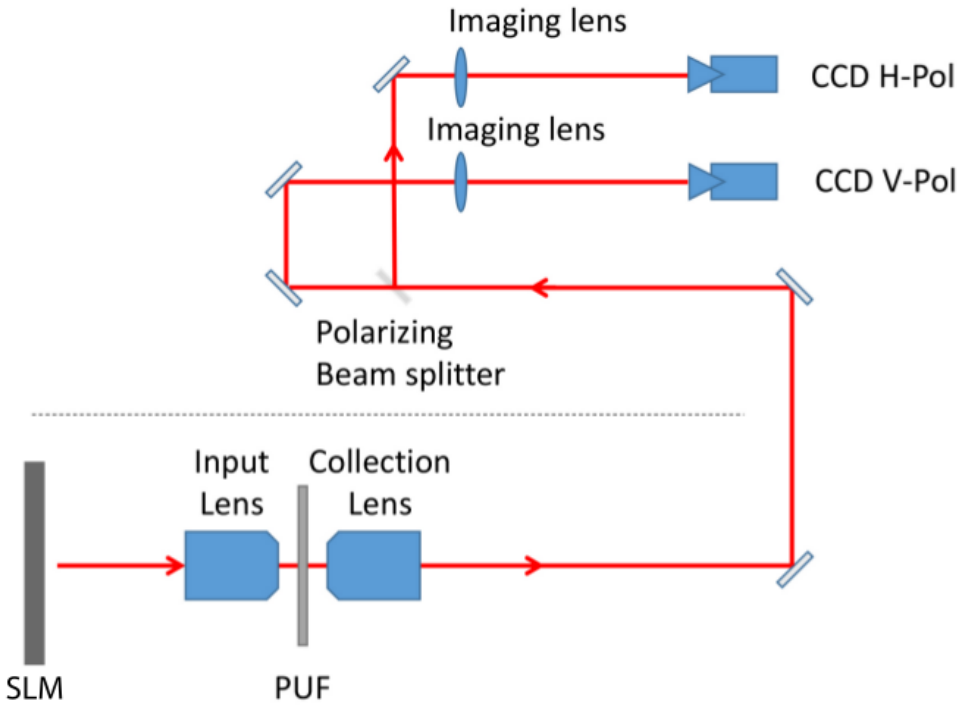


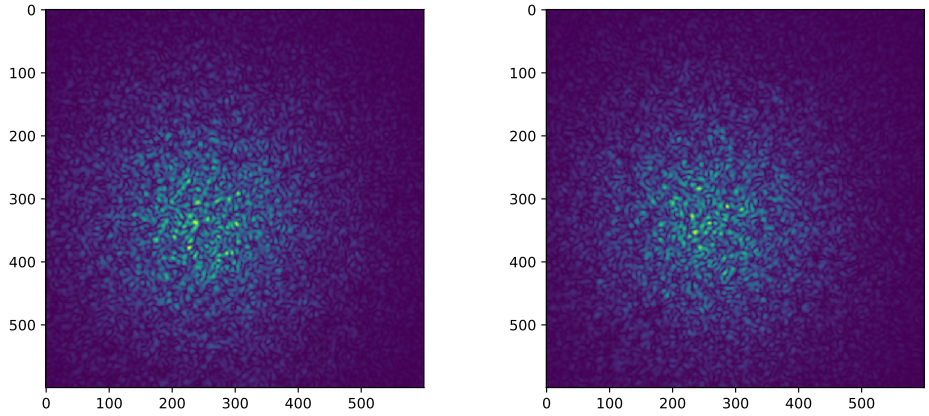
Figure 4.1 – Experimental setup used in Real-Dataset measurement.

output, horizontal input and vertical output.

4.2.2 Distinguishing PUFs

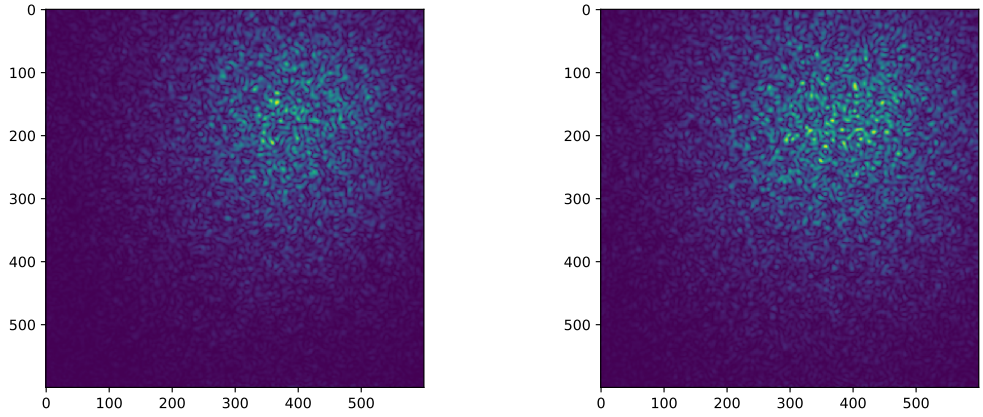
The intra-, inter-PUF, and inter-Challenge-distance characteristics which assess PUF properties are calculated as follows:

- For a particular pair of challenge and PUF, the intra-distance measures the distance between the two responses resulting from applying this challenge to this PUF with these two different polarizations.
- For a particular pair of challenge and polarization, the inter-PUF-distance between two different PUFs measures the distance between the two responses resulting from applying this challenge with this polarization to both PUFs.
- For a particular pair of PUF and polarization, the inter-Challenge-distance measures the distance between the two responses resulting from applying these two different challenges to this PUF with this polarization.



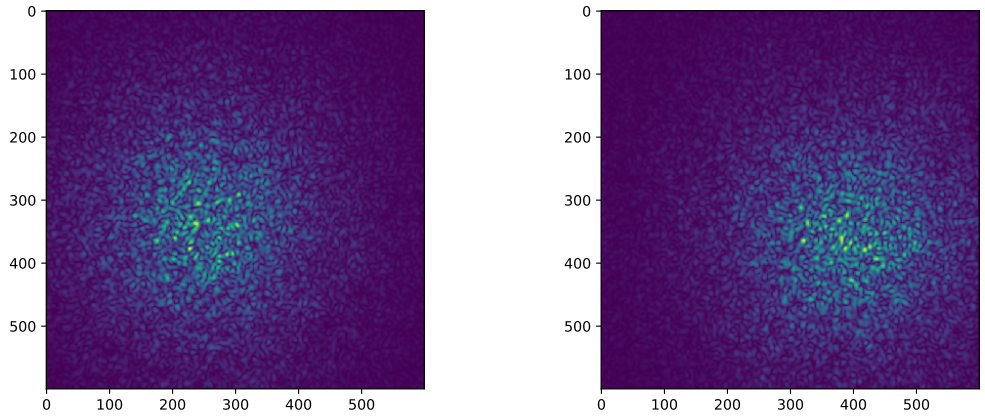
(a) PUF #1, challenge #1, Polarization hh (b) PUF #1, challenge #1, Polarization vv

Figure 4.2 – Examples of speckle images for visualizing Intra-distance. (hh:horizontal input and output polarization, vv: vertical input and output polarization)



(a) PUF #1, challenge #3, Polarization hv (b) PUF #2, challenge #3, Polarization hv

Figure 4.3 – Examples of speckle images for visualizing Inter-PUF distance. (hv: horizontal input and vertical output polarization)

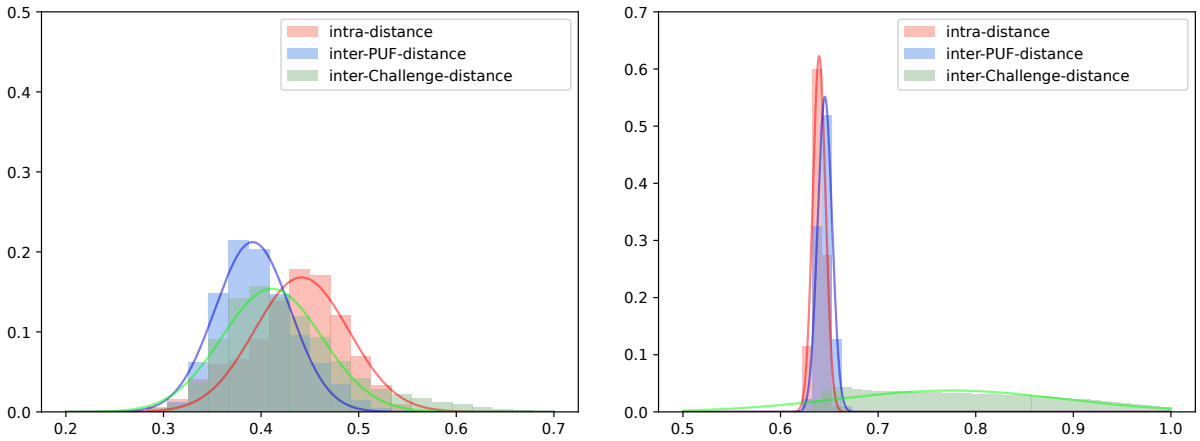


(a) PUF #1, challenge #1, Polarization hh (b) PUF #1, challenge #2, Polarization hh

Figure 4.4 – Examples of speckle images for visualizing Inter-Challenge distance

Figure 4.2, 4.3 and 4.4 show examples of speckle images (amplitudes) from Complex-Dataset for visualization of intra-, inter-PUF, and inter-Challenge-distance concepts.

In Figure 4.5, the three Euclidean distances are summarized by presenting histograms displaying the frequency of occurrence of each distance observed over several different challenges, polarization cases, and PUFs. The distances are computed by Euclidean norm of the differences between pixel values of two speckle patterns.



(a) Real-Dataset (b) Complex-Dataset

Figure 4.5 – Intra-, inter-PUF, and inter-Challenge-distance distributions. The x-axis represents Euclidean distance and the y-axis represents normalized frequency.

Inter-PUF-distance signifies a concept of uniqueness, i.e., it determines how much it is possible to distinguish two PUFs based on their output speckle images. This plot indicates that given a speckle image, it is not possible to detect from which PUF the image originates. Moreover, for a fixed PUF, the correlations between all responses are not significant. Thus, they can not be distinguished from the responses of another PUF. Since we observe that there is no clear separation between these three distributions, we decide to 1) use optical PUFs in a passive way and 2) have a description stage extracting more distinguishable features.

Like biometric authentication systems, we consider a pair of PUF and challenge as an individual. Here the dataset consists of 20×919 identities with four images for each individual, including speckle images of all four different polarizations. Ultimately, in this setting, we learn descriptors such that the inter-PUF and inter-Challenge-distance will be large, and the intra-distance will be small, which allows reliable authentication of individuals.

4.3 Learning Descriptors

A fundamental requirement for an efficient and scalable privacy-preserving authentication is to have short descriptors for the PUF measurement data. In this section, we aim to provide descriptors for speckle patterns obtained from Optical PUF responses. Such descriptors have not been studied in the literature in the context of PUF authentication protocols.

4.3.1 Preprocessing

In order to reduce the processing time for analyzing speckle images, we need to resize them. Two resizing process we can think of are cropping or scaling the images. The size of the images is reduced by a scale of 0.5 to obtain 300×300 images, and so do the cropping methods.

We also have two cropping strategies: cropping #1 that crop the image such that the center of mass of the image is in the middle. This center is found by computing the image moments which are defined as the average intensity of the pixels of an im-

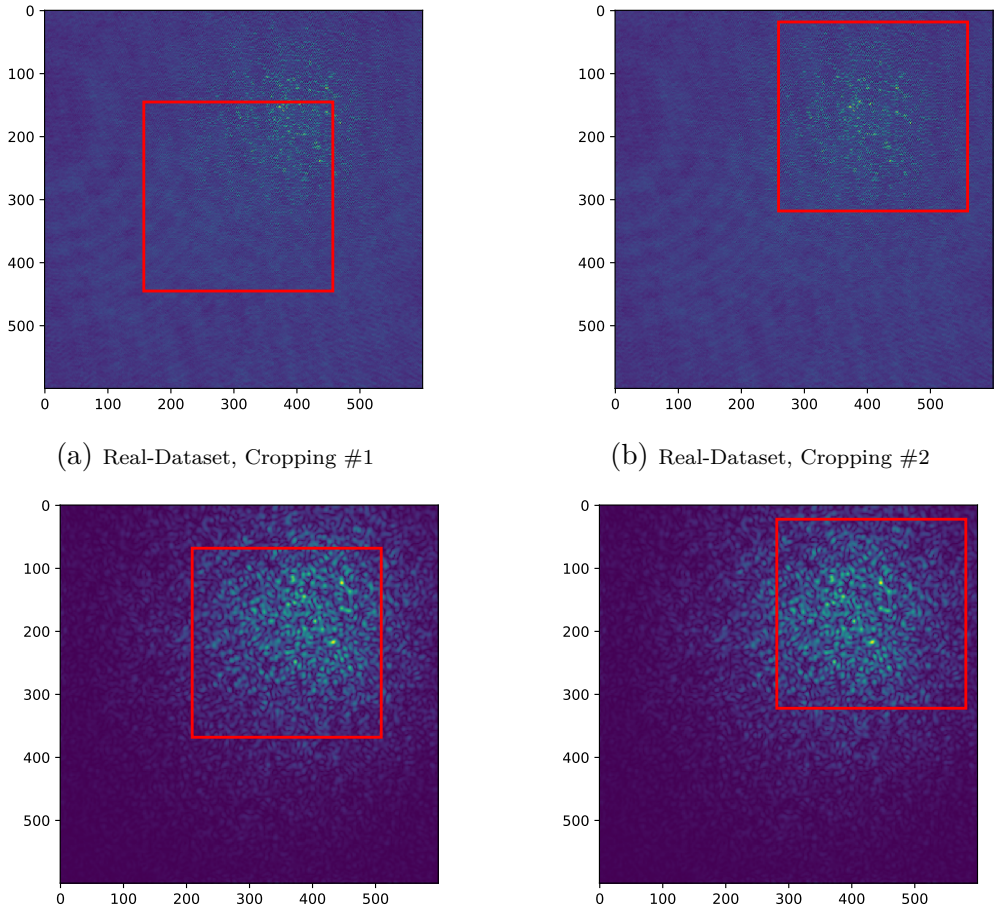
age. The $(p, q)-$moment for a greyscale image where the pixel intensities are $I(i, j)$ is $M_{p,q} = \sum_i \sum_j i^p j^q I(i, j)$. By normalizing the first-order moments $M_{0,1}$ and $M_{1,0}$ by $M_{0,0}$, we get the center of mass' coordinates, *i.e.*, $C_x = \frac{M_{1,0}}{M_{0,0}}$ and $C_y = \frac{M_{0,1}}{M_{0,0}}$. Cropping #2 detects the center of the location of the brightest area in the image, where the coordinate of the brightest area is found by thresholding the image and clearing the borders, then crops the area centered on this point.

Figure 4.6 shows the result of two cropping process applied on a speckle image from both datasets. The area to be cropped is shown with the red square. It looks that the two cropping strategies work more similarly on Complex-Dataset, whereas they work differently on the Real-Dataset. In real images, it might be due to the noise that the center of mass is not located in the center of the brightest area. Indeed, in the Real-dataset, the speckle patterns consist of bright spots, yet, the dark area is much more dominant, so cropping the image by computing the center of mass does not give us the part of the image we are interested in. On the contrary, there is high contrast in the images of the Complex-dataset, and the bright spot looks much stronger.

To determine how the three different ways of resizing (*i.e.*, scale reduction and cropping #1 and cropping #2) affect the recognition process, we used raw images in an authentication protocol where the system performs a one-to-one comparison of a query response image against a specific speckle image stored in a database at enrollment.

At each time, half of the individuals in the dataset are selected randomly for enrollment. One random template of each individual is enrolled in the system, playing the role of $\mathbf{x}_i$ and one other is used for the query. The remaining individuals not enrolled in the system $(N/2)$ play the role of impostors. We repeat this process several times and average the results in terms of probability of true positive P_{tp} when the probability of false positive $P_{\mathsf{fp}} = 0.01$.

Table 4.1 shows the performance on raw images of Real-Dataset and Complex-Dataset with three resizing strategies. It shows as it is expected the Complex-Dataset contain more information than Real-Dataset. It is supposed that converting the raw images to complex images is done with kind of denoising which enhances the signal.



(a) Real-Dataset, Cropping #1

(b) Real-Dataset, Cropping #2

(c) Complex-Dataset, Cropping #1

(d) Complex-Dataset, Cropping #2

Figure 4.6 – Cropping

Data	$P_{\mathsf{tp}}@P_{\mathsf{fp}} = 0.01$
Cropping #1 (Real-Dataset)	0.049
Cropping #2 (Real-Dataset)	0.030
Scaling (Real-Dataset)	0.059
Cropping #1 (Complex-Dataset)	0.179
Cropping #2 (Complex-Dataset)	0.018
Scaling (Complex-Dataset)	0.729

Table 4.1 – The authentication performance on raw images

We need some kind of image enhancing that enables the system to distinguish the images even more. In the following, we feed the resized images to a deep neural network and see to what extent the network will be able to learn this denoising to improve the verification results.

4.3.2 Siamese Networks

A Siamese Neural Network is a type of neural network containing two or more instances of the same model with the same architecture, with the same parameters and weights. Here, the goal is to make sure that two images with the same label have their embedding close together in the embedding space while two inputs with different labels have their embedding far away. This is the central idea behind the Siamese Networks.

In the case of standard classification, where a neural network learns to predict multiple classes, deep neural networks need a large number of images for each of the classes during the training. Besides, when we need to add or remove new classes to the data, we must re-train the model again. Since Siamese Networks learns to differentiate between images, learning their similarity instead of classifying the images, we can classify new classes of data by comparing the images without training the network again.

As in our application, descriptors should be learned to minimize the overlap between intra and inter-distances. In addition, we do not have enough data for each PUF, and also, the number of PUFs is huge. This type of networks would be perfect.

Triplet loss [SKP15] is used to train a Siamese Network. Triplet loss [3] takes three inputs: an anchor a (or the reference input), a positive p of the same identity as the

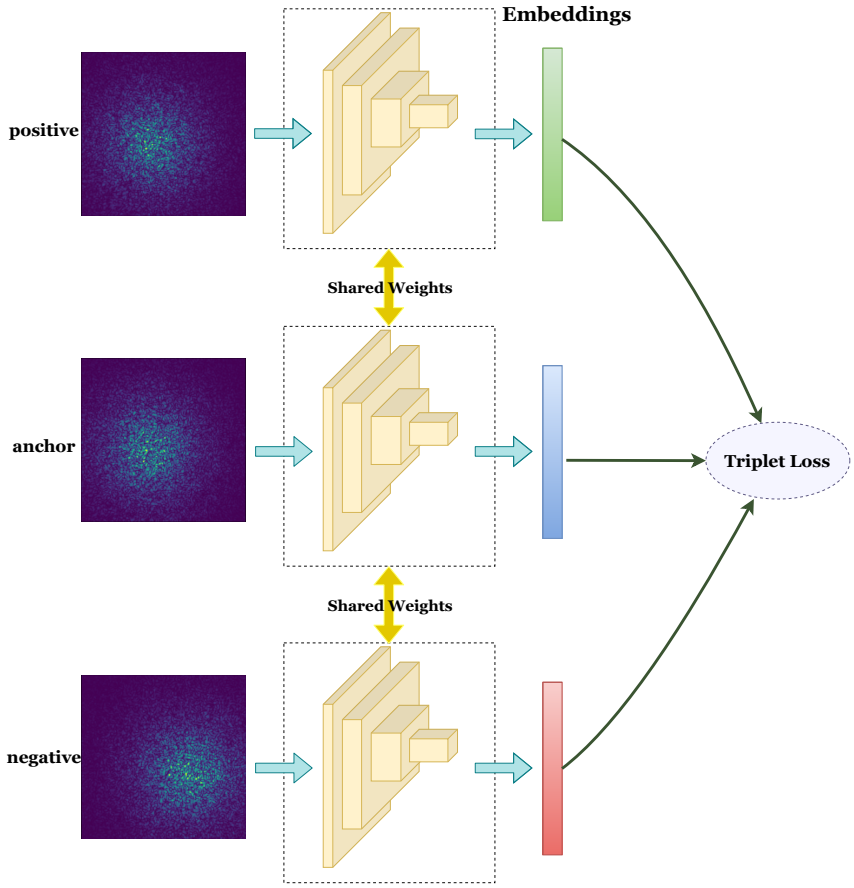


Figure 4.7 – Training of Siamese Network with Triplet Loss.

anchor, and a negative n with a different identity from the anchor.

As illustrated in Figure 4.7, first, we compute the d dimensional embedding of each image by feeding them to the same CNN with the same parameters. Then, using some distance function d, these three embeddings are passed to the Triplet Loss function, which is defined as:

$$\mathcal{L} = \max(d(a,p) - d(a,n) + margin, 0) \tag{4.1}$$

Minimizing the loss function 4.1, pushes $d(a,p)$ to 0 and $d(a,n)$ to be greater than $d(a,p) + margin$. In other words the anchor is pushed away from the negative while pushed closer to the positive.

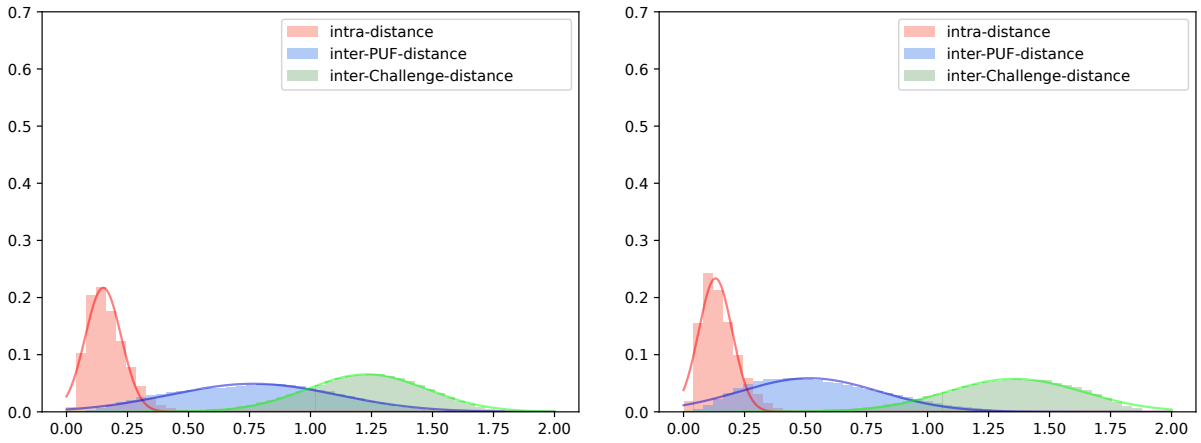
The critical point is how to prepare triplet batches for training as some triplets are more useful than others. By online triplet mining, we compute useful triplets for each batch of inputs. Given a batch of B examples consisting of P different PUFs with K challenges each, we compute the B embeddings, which provides us a maximum of B^3 triplets. Most of these triplets are not valid since they do not include two positives and one negative. Then we use the Batch-hard strategy, which works as follows: for each anchor input, it selects the hardest positive, i.e., a positive with the biggest distance $d(a,p)$ and hardest negative, i.e., a negative with smallest $d(a,n)$ among the batch. This procedure produces PK triplets, which are the hardest among the batch.

Implementation Details We employed ResNet-50 [HZRS16] as the backbone CNN model to produce the embeddings. The embedding dimension is set to 1024. We split the dataset into train, validation, and test subsets using a 70/20/10 ratio. The mini-batch consists of 64 images with 16 different PUFs and four images per identity in each mini-batch. The margin is set to 0.3. The training process takes 400 epochs in total. Adam optimizer, with a learning rate of 0.001 optimizes the model parameters. We implemented the experiments in Pytorch. We saved the checkpoint with the highest authentication accuracy on the validation set and evaluated it on the test set as the accuracy of the model.

Results Using one of the preprocessing procedures, we train a network to learn descriptors. Here, we use the same input data as in Section 4.2.2 and compute intra-, inter-PUF,

and inter-Challenge distances, to see how the learned descriptors are helpful to distinguish between PUF responses.

As Figure 4.8 illustrates, the overlap between intra and inter-distance distributions has been reduced significantly. Especially on Real-Dataset, the discrimination is stronger. Moreover, the separation between inter-PUF and inter-Challenge distances in Figure 4.8b is exciting and might help further analysis.



(a) Real-Dataset (b) Complex-Dataset

Figure 4.8 – Intra-, inter-PUF, and inter-Challenge-distance distributions after learning descriptors.

In order to assess the efficiency of the trained model on test split, the descriptors are computed by feeding the speckle images into the trained network and evaluated in an authentication scenario. Table 4.2 shows the authentication performance on both datasets with three preprocessing strategies.

As Table 4.2 shows, compared to the authentication results computed on the raw image (Table 4.1), the network learns very well on Real-Dataset, which are noisy. Indeed, the network succeeds in removing the noise from real images and achieving as much performance as when learned on complex images that are supposed to have better quality and contain much more information. Besides, we noticed that cropping is not the best way to reduce the size of the images; especially on Complex-dataset cropping causes huge information loss.

Data	$P_{\mathsf{tp}}@P_{\mathsf{fp}} = 0.01$
Cropping #1 (Real-Dataset)	0.952
Cropping #2 (Real-Dataset)	0.964
Scaling (Real-Dataset)	1
Cropping #1 (Complex-Dataset)	0.512
Cropping #2 (Complex-Dataset)	0.016
Scaling (Complex-Dataset)	0.9996

Table 4.2 – The authentication performance on test set after learning descriptors

Learning the descriptors with image scaling achieves the best performance and works on both datasets equally well. Thus, it is still unclear which dataset is more efficient to use in this context. Yet, the Real dataset seems closer to real-life application, which is less complicated to capture. In the following, we make the problem more difficult to assess how different these two datasets behave ultimately.

4.4 Group Membership Verification

By manipulating descriptors designed in the last section, we develop a group membership verification scheme that checks whether the PUF query matches one of the previously enrolled PUFs of a given group.

Similar to experimental setup described in Section 2.2.2, the set of PUFs is partitioned into M groups such that all the groups have the same cardinality, m. The PUF responses that enrolled are the vectors $\{\mathbf{x}_1, \ldots, \mathbf{x}_N\} \subset \mathbb{R}^d$. The output of the enrollment is a $l \times M$ matrix $\mathbf{R} = [\mathbf{r}_1, \ldots, \mathbf{r}_M]$ composed of the representations of the M groups. The group representations are quantized and sparse: $\mathbf{r}_g \in \mathcal{A}^l$ with $\mathcal{A} := \{-1, 0, 1\}$ and $\|\mathbf{r}_g\|_0 \leq S < l$, $\forall g \in \{1, 2, \ldots, M\}$.

A PUF query claims that it belongs to group g. This claim is valid under hypothesis $\mathcal{H}_1$ and false under hypothesis $\mathcal{H}_0$. The PUF response $\mathbf{q}$ is embedded onto $\mathcal{A}^l$, the embedding, and the claimed group number g is sent to the system, which compares the embedded query to the group representation $\mathbf{r}_g$. The system accepts ($t = 1$) or rejects ($t = 0$) the claim. This is a two hypothesis test with two probabilities of errors: $P_{\mathsf{fp}} := \mathbb{P}(t = 1|\mathcal{H}_0)$ is the false positive rate and $P_{\mathsf{tp}} := \mathbb{P}(t = 1|\mathcal{H}_1)$ is the true positive rate.

We evaluate the group membership verification protocol based on the following metrics:

- **Verification performance**: The ability of the protocol to correctly perform the verification task that we considered $P_{\mathsf{tp}}(\tau)$ for τ s.t. $P_{\mathsf{fp}}(\tau) = 0.05$
- **Privacy**: A curious server can reconstruct the query response from its embedding. The mean squared error assesses how accurate is this reconstruction.
- **Security**: As for the security of the enrolled templates, a curious server can reconstruct a single vector $\hat{\mathbf{x}}$ from the group representation, which is the same for all the members of that group.

Exp #1: Complex-Dataset vs. Real-Dataset Figure 4.9 compares the verification performance and the security of group representations for varying group size m on two datasets.

Here, images are resized in two different scales to determine how various sizes affect the verification process. The images are either resized to the input size of Res-Net50, which is 224×224 or 300×300. In the latter case, because the input image size to the network is 300×300 instead of 224×224, an adaptive pooling with the output size of 1×1 is applied before the fully connected layer (the kernel size is selected automatically to produce an output of the given dimensionality).

We observe that verification performances decreases and security increases as the number of enrolled signatures increases. Yet the trade-off between security and performance is more impressive on Real-Dataset with images of size 224×224. It makes sense since images with smaller sizes carry less information. Besides, real images contain less information than complex images.

As Real-Dataset results in better performance in terms of both security and performance, we use this dataset for the following experiments.

Exp #2: Comparison of Performance of different schemes In this experiment, we compared different schemes proposed in Chapter 2; group representation can be computed by one of the following schemes:

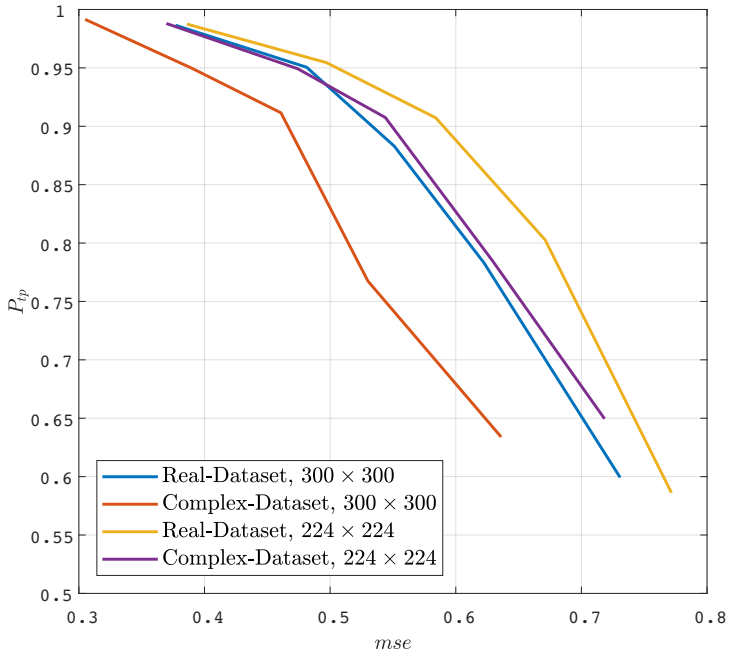


Figure 4.9 – Trade-off between security and performance for varying group size m on two datasets with different image sizes

- **EoA-pinv** first aggregates the raw signatures $\mathbf{X}_g := [\mathbf{x}_{g,1}, \ldots, \mathbf{x}_{g,m}]$ into a unique vector as $(\mathbf{X}_g^{\dagger})^{\top}\mathbf{1}_m$, then the group signature is computed by embedding the aggregated vector.

- **AoE-sum** embeds each signature before aggregating with sum pooling and then applies sign to obtain group representation.

- **EoA-ML** described in Section 2.2.1 learns group representation and embedding jointly based on the embedding of aggregated vectors.

- **AoE-ML** learns the group signatures with the aggregation of embedding construction, explained in Section 2.2.1.

- **JLAR** introduced in Section 2.3 aims to learn assignments and group representations jointly.

Figure 4.10 demonstrates that the plot corresponding to the JLAR is way above the rest of the plots. In fact, the small verification performance for those plots is caused by the significant losses in information. In contrast, JLAR tries to learn group representation

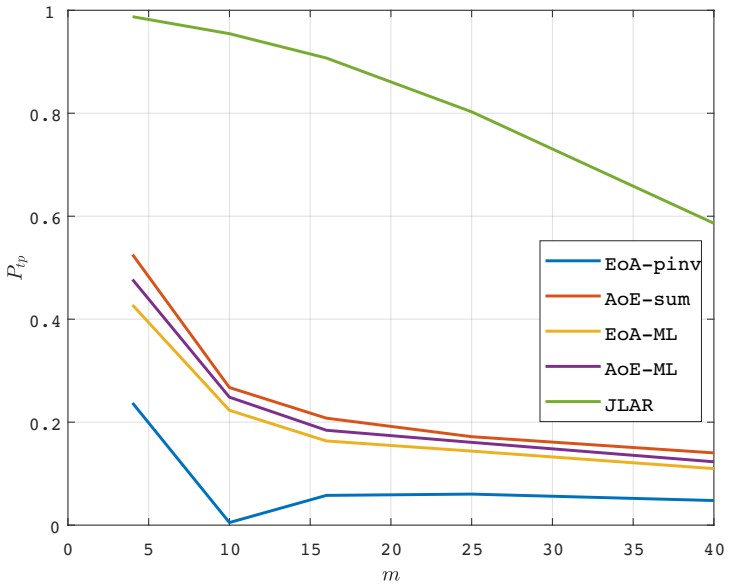


Figure 4.10 – Performance comparison of different schemes for varying group size m

and assignments simultaneously, which assigns similar vectors into groups so as to lose less information.

Exp #3: Investigating the impact of sparsity Security and privacy are measured by the ability to reconstruct signatures from group representations and embeddings respectively. In this experiment, we examine how the sparsity level affect the reconstruction of query and group representation.

Figure 4.11 illustrates following fact. First, thanks to aggregation, reconstruction of enrolled PUFs from group representation is more complicated than reconstructing the query from its embedding. Second, even though in JLAR assignment is based on similarity of PUF responses, it achieves the desired balance between security and performance.

4.5 Conclusion

This chapter explored two types of experimental data obtained from optical PUFs: PUF measurements captured by a camera and the complex speckle images measured by applying a few corrections. In order to use the available PUF data in a setting similar to biometric applications, descriptors must be learned to differentiate different PUF mea-

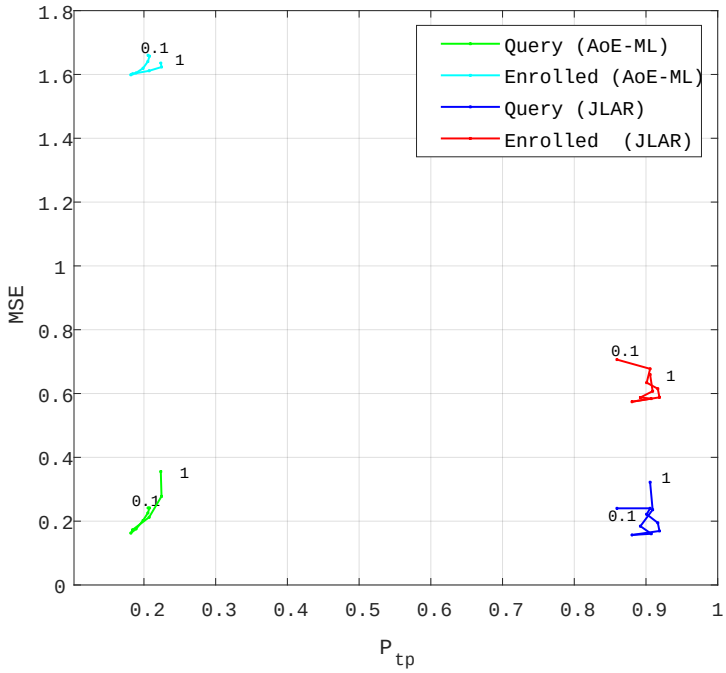


Figure 4.11 – The impact of sparsity factor with $m = 16$, $S/d \in [0.1, 1]$

surements. The learning is modeled so as to minimize the distance between responses that result from the same PUF (intra-distance), whereas the distance between responses from different PUF is maximized (inter-distance).

Both datasets proved to have quite interesting results during the authentication procedure. In addition, we examined the group membership verification schemes using the learned descriptors. The study shows that joint learning of group representations and group assignments can lead to better results. This implies that rather than random assignment, groups should be formed in such a way that similar PUF responses would be assigned to the same group.

Here we considered a pair of PUF and challenge as one individual. It is more desirable if each PUF is regarded as one individual and the descriptors learned so that the correlation between challenges of a given PUF is maximized. Yet, there are some limitations with available data in this setting. First, the limited number of classes (we have 20 classes); Second, we still do not know how to generate noisy measurements, i.e., imposter examples.

However, in Figure 4.8b, disregarding intra-distances, we observe robust discrimination

between inter-PUF and inter-Challenge distance distribution of learned descriptors, which means that even training under the current setting distinguishes between PUFs in the other setting.

Conclusion and Future Work

This chapter summarizes the contributions to this thesis, followed by some future research directions.

Conclusion

Throughout this thesis, we study the privacy-preserving group membership verification problem. The procedure that determines whether an item or individual is a member of a group. The core component of our work constructs a compact group representation. The procedure is based on aggregating and embedding of templates of the group member that keeps track of the membership property at test time as well as providing privacy and security.

We demonstrate that our schemes provide security as the group representation is sufficiently protected. This makes it impossible to reconstruct the signatures from group representations while identifying noisy versions of them. Further, the server is only able to link each signature to its group number. Identity is not revealed during verification, so privacy is preserved as well.

The keystones are the aggregation and embedding functions. So, in Chapter 2, first, we explore approaches in which aggregation and embedding of group members are based on deterministic functions. Afterward, we replace those passive functions and hard-coded parameters with functions producing the same types of output, but their parameters are learned through optimization.

We also explore the group identification protocol that identifies which specific group the individual belongs to without disclosing its identity. However, group identification, a 1-to-many task is more challenging than group verification, a 1-to-1 problem. We demonstrate that this can also be handled well.

Subsequently, rather than considering group assignments that are predetermined, group assignments are also learned together with representations of the groups. With this approach, the enrolled signatures are grouped according to similarities, aiming to reduce the overall distance between group members while increasing separation between groups within the embedded domain.

We argue that the performance of group membership verification may be affected by the sparsity of the embeddings. Chapter 3 presents a mathematical model for group membership verification, which reveals the impact of sparsity on both security and compactness, along with verification performance.

Our studies on binary and ternary alphabets point to the following conclusions. First, the sparse setup can be considered optimal with regard to verification performance under the assumption that there is no noise. Yet, in practice, the channel error will always be greater than zero, which indicates that it is relatively challenging to have a sparse solution. Second, the dense setup is impressive in terms of security level as well as verification performance under low-SNR conditions that the genuine are less correlated with the enrolled templates.

Chapter 4 discusses the experiments we conducted as part of the ID_IOT project. We examine two datasets of experimental data obtained from optical PUFs. The first includes PUF measurements captured by a camera, and the second contains complex speckle images measured by applying a few corrections. We assume that PUF authentication will have a similar structure as biometrics. The problem is how to generate descriptors for PUF responses so that enabling the system to distinguish between PUFfs. Descriptors are learned by training a neural network such that the distance between responses from a given PUF is minimized (intra-distance) while the distance between responses from different PUFs is maximized.

The result of the authentication procedure on both datasets is interesting. Following that, we analyze the group membership verification protocol using the learned descriptors. It demonstrates that learning jointly group representation and group assignments results in a better performance. Therefore, the verification performance depends largely on how the groups are formed. It is suggested that similar PUF responses be assigned to the

same group.

Future Work

Here, we discuss some possible directions for future work.

Group membership verification There are some limitations with the group membership verification schemes proposed in Chapter 2. First, the learning of group representations is not entirely free. Some guidances are still imposed, especially the prototyping of the embedding based on a sparse ternary quantization. This is mainly for inheriting the security and privacy properties of this lossy information processing. We can explore an alternative approach, such as binary embedding. Second, it is assumed that every individual belongs to exactly one group. However, in some applications, an individual can belong to different groups with different access privileges. Third, another strict assumption we have taken is that groups have the same size. It would be more interesting to develop group membership protocols that allow members to belong to more than one group and groups can be of different sizes, as well.

Processing in the encrypted domain The protection of enrolled templates is provided by aggregation. The protection level of the query can be boosted by leveraging low-cost partially homomorphic encoding schemes on top of our work. In this manner, we will be able to build more secure protocols by combining such encryption with our group membership verification scheme. Therefore, unauthorized parties cannot learn anything that they are not supposed to. Our system is designed in line with this protocol. Quantizing all aggregated vectors ensures adoption by cryptography algorithms. Developing this protocol can be explored further.

PUF experiments: Assuming a similar structure as biometrics for the PUF authentication, we consider a pair of PUF and a challenge to be as if they were one individual. The best course would be for each PUF to be considered as one individual where the same PUF might have multiple Challenge-Response pairs (in contrast, the current system is enrolled as a single pair per item). The descriptors are thus learned so that the correlation between challenges within a given PUF is maximized. Yet, we notice robust discrimination between the inter-PUF and inter-Challenge distance distributions of learned descriptors

in Figure 4.8b. So that even by training under the current setup, the PUFs in the other setting can also be distinguished from one another. With available data, however, some limitations exist when working in this setting such as: the limited number of PUFs (20 PUFs), and the difficulty in generating noisy measurements, i.e., imposter examples.

The model of embedding: Chapter 3 only examines a simple model where we have a symbol 0 with a different probability, while the rest are equal. The findings are only valid for this embedding function. We do not know what the best embedding is. There could be embedding functions that are less prone to channel errors, so it would be more interesting to learn the embedding model from a training set. To this end, we propose a quantization learning scheme presented in the Appendix A.1, which would be a potential direction for future work.

Learning descriptors and group representations In order to compute the group representations, we assume that the descriptors have already been obtained (from a pre-trained network) during a separate phase. However, we can design a neural network for learning the descriptors as well as their aggregation over a group. In this regard, we develop a deep architecture, AggNet, which aims to simultaneously learn face descriptors and the binary group representations suitable for group membership verification tasks. This is achieved by training the network so that the loss in verification performance caused by aggregating multiple faces into a single descriptor is minimized. The network architecture and the loss function of AggNet is introduced in Appendix A.2. This work is being evaluated, and the results are currently under preparation.

Appendix A

Appendix

A.1 Quantization Learning

We aim to learn an embedding function motivated by the similarity score defined in (3.28). The embedding function $\mathsf{h} : \mathbb{R}^d \mapsto \mathcal{Z}$ partitions the feature space into z cells by mapping a vector to an integer in $\mathcal{Z} = \{1, ..., z\}$. Define $\mathbf{W} \in \mathbb{R}^{z \times z}$ as a weight matrix, where $W(i, j)$ is an estimation of similarity between cell $\mathcal{C}_i$ and $\mathcal{C}_j$. In other words, $W(i, j)$ indicates that if $\mathbf{x}$ is quantized to the i-th cell, how much the probability of mapping $\mathbf{q} = \mathbf{x} + \mathbf{n}$ to the j-th cell is. Then the similarity between two points $\mathbf{x}$ and $\mathbf{q}$ is computed as:

$$\mathsf{Sim}(\mathbf{x}, \mathbf{q}) = W_{\mathsf{h}(\mathbf{x}),\mathsf{h}(\mathbf{q})} \tag{A.1}$$

Consider $\mathcal{X}_{ref}$ is a set of reference points where each point is associated with a set of positive points $\mathcal{X}_p$. The embedding function h and weight matrix $\mathbf{W}$ are learned by simultaneously maximizing the similarity score of positive pairs and also minimizing the score for negative (nonpositive) pairs. The objective function for each reference point $\mathbf{x}_{ref} \in \mathcal{X}_{ref}$ is defined as:

$$\max_{\mathbf{W}, i_{ref}, i_p, \mathcal{C}} \frac{1}{|\mathcal{X}_p|} \sum_{\mathbf{x}_p \in \mathcal{X}_p} (\mathcal{J}(i_{ref}, i_p, \mathbf{W}) - \lambda \mathcal{D}(\mathbf{x}_p, \mathcal{C}_{i_p})) - \lambda \mathcal{D}(\mathbf{x}_{ref}, \mathcal{C}_{i_{ref}}) \tag{A.2}$$

$\mathcal{J}$ and $\mathcal{D}$ are considered as the gain of detection and the penalty of quantization, respectively. i_{ref} and i_p indicate the index of cells the reference point and the positive point mapped to. The following two different schemes are proposed based on the definition of detection gain function $\mathcal{J}$.

Mean Cost Function

Consider $\mathcal{X}_n$ as the set of negative pairs, then the similarity loss is formulated as:

$$\mathcal{J}_{i_{ref}} = \frac{1}{|\mathcal{X}_p|} \sum_{\mathbf{x}_p \in \mathcal{X}_p} W(i_p, i_{ref}) - \frac{1}{|\mathcal{X}_n|} \sum_{\mathbf{x}_n \in \mathcal{X}_n} W(i_n, i_{ref}) \tag{A.3}$$

The similarity of negative pairs can be estimated by:

$$\mathbb{E}[W(I_n, i_{ref})] = \sum_{i=1}^{z} p_i W(i, i_{ref}); \quad p_i = P(I_n = i) = \frac{N_i}{N} \tag{A.4}$$

N and N_i denote the total number of points and the number of negative points assigned to $\mathcal{C}_i$, respectively. Then, we define the overall objective function as:

$$\begin{aligned} &\max_{\mathbf{W}, \mathcal{C}, i_{ref}, i_p} \quad \sum_{\mathbf{x}_{ref} \in \mathcal{X}_{ref}} [\frac{1}{|\mathcal{X}_p|} \sum_{\mathbf{x}_p \in \mathcal{X}_p} (W(i_p, i_{ref}) - \lambda ||\mathbf{x}_p - \mathbf{c}_{i_p}||_2) - \sum_{k=1}^{z} p_k W(k, i_{ref}) - \lambda ||\mathbf{x}_{ref} - \mathbf{c}_{i_{ref}}||_2] \\ &\text{s.t.} \qquad \mathsf{g}(\mathbf{W}) = \mathbf{d} \end{aligned} \tag{A.5}$$

where g defines a constraint on $\mathbf{W}$ and $\mathbf{c}_i \in \mathcal{C}_i$. This problem can be solved iteratively by alternative optimization, updating one parameter while the others are fixed.

Update W: When partitioning is fixed, the optimization problem is:

$$\begin{aligned} &\max_{\mathbf{W}} \quad \sum_{i=1}^{z} \sum_{j=1}^{z} a_{ij} W(i, j) \\ &\text{s.t.} \quad \mathsf{g}(\mathbf{W}) = \mathbf{d} \end{aligned} \tag{A.6}$$

Here $a_{ij} = \sum_{x_{ref} \in \mathcal{A}_j} \frac{\sum_{|\mathcal{X}_p|} \mathbf{1}_{[i_p = j]}}{|\mathcal{X}_p|} - p_i$ and $\mathcal{A}_j = \{x_{ref} | i_{ref} = j\}$ is a set of reference points assigned to the j-th cell. Based on the definition of the constraint, we have the following solutions:

- Unit norm (column-wise): $\sum_i W(i, j)^2 = 1$
  The Lagrange multipliers found as $\lambda_j = \frac{\sqrt{\sum_{i=1}^{z} a_{ij}^2}}{2}$ and the weight matrix elements computed by $w_{ij} = \frac{-a_{ij}}{2\lambda_j}$.

- Unit variance and zero mean (negative scores): The mean and variance for the

distribution of negative scores is defined as:

$$\mathbb{E}(S_n) = \sum_j \sum_i p_i p_j W(i,j),\ \mathbb{V}(S_n) = \sum_j \sum_i p_i p_j W(i,j)^2 - \mathbb{E}(S_n)^2 \tag{A.7}$$

The weights founds as $W(i,j) = \frac{-a_{ij} - \lambda p_i p_j}{2\mu p_i p_j}$ where $\lambda = \frac{-\sum_{i,j} a_{ij}}{\sum_{i,j} p_i p_j}$ and $\mu = \sqrt{\sum_{i,j} \frac{(a_{ij} + \lambda p_i p_j)^2}{4 p_i p_j}}$.

Update partitioning: The weight matrix is fixed; then, the assignments and centroids are updated by stochastic gradient ascent given in Algorithm 1.

Algorithm 1 Stochastic Gradient Step

for $x_{ref} \in \mathcal{X}_{ref}$ **do**
  $i'_{ref} \leftarrow i_{ref}$
  for $j \in \{1, ..., z\}$ **do**
    Compute $\mathcal{L}_1(j) =$
      $\frac{1}{|\mathcal{X}_p|} \sum_{\mathbf{x}_p \in \mathcal{X}_p} W(i_p, j) - \sum_{i=1}^{z} p_i W(i,j) - \lambda ||\mathbf{x}_{ref} - \mathcal{C}_j||_2$
  end for
  $i_{ref} \leftarrow \arg\max_j \mathcal{L}_1(j)$
  for $x_p \in \mathcal{X}_p$ **do**
    for $j \in \{1, ..., z\}$ **do**
      Compute $\mathcal{L}_2(j) =$
        $\frac{1}{|\mathcal{X}_p|} W(j, i_{ref}) - \lambda ||\mathbf{x}_p - \mathcal{C}_j||_2$
    end for
    $i_p \leftarrow \arg\max_j \mathcal{L}_2(j)$
  end for
  if $i_{ref} = i'_{ref}$ **then**
    $\mathcal{C}_{i_{ref}} \leftarrow \mathcal{C}_{i_{ref}} + \eta_1 (x_{ref} - \mathcal{C}_{i_{ref}})$ ▷ Reinforcing it
  else
    $\mathcal{C}_{i'_{ref}} \leftarrow \mathcal{C}_{i'_{ref}} - \eta_2 (x_{ref} - \mathcal{C}_{i'_{ref}})$ ▷ getting away
    $\mathcal{C}_{i_{ref}} \leftarrow \mathcal{C}_{i_{ref}} + \eta_3 (x_{ref} - \mathcal{C}_{i_{ref}})$ ▷ moving closer
  end if
end for
return $\mathcal{C}$

KL Cost Function

Based on KL divergence the distance between positive and negative distributions is:

$$\mathcal{D}_{KL}(P||N) = \sum_{i,j=1}^{z} P_p(i_p = i, i_{ref} = j) \log \frac{P_p(i_p = i, i_{ref} = j)}{P_n(i_n = i, i_{ref} = j)} \tag{A.8}$$

The wights are considered as $\hat{W}(i,j) = \log \frac{\hat{P}_p(i_p=i|i_{ref}=j)}{\hat{P}_n(i_n=i,|i_{ref}=j)}$ and the two probabilities are estimated as:

$$\begin{aligned} \hat{P}_p(i_p = i|i_{ref} = j) &= \frac{n_{i|j}}{|\mathcal{X}_p| * n_{ref|j}} \\ \hat{P}_n(i_n = i|i_{ref} = j) &= \frac{n_q - n_{i|j}}{|\mathcal{X}_p| * (|\mathcal{X}_{ref}| - n_{ref|j})} \end{aligned} \tag{A.9}$$

where $n_{ref|j} = |\{\mathbf{x}_{ref} \in \mathcal{X}_{ref}|i_{ref} = j\}|$.

Now, the similarity loss is expressed as $\mathcal{J} = \sum_{x_{ref} \in \mathcal{X}_{ref}} \mathcal{J}_{ref}(i_{ref})$ that:

$$\mathcal{J}_{ref}(i_{ref}) = \sum_{i=1}^{z} \frac{|\{\mathbf{x}_p \in \mathcal{X}_p|i_p = i\}|}{|\mathcal{X}|_p} \hat{W}(i, i_{ref}) \tag{A.10}$$

Here, $\mathbf{W}$ is optimized like (A.6) and partitioning is updated in the similar procedure as Algorithm1 by substituting $\mathcal{L}_1$ and $\mathcal{L}_2$ as below:

$$\begin{aligned} \mathcal{L}_1(j) &= \sum_{i=1}^{z} \frac{|\{\mathbf{x}_p \in \mathcal{X}_p|i_p = i\}|}{|\mathcal{X}|_p} \hat{w}_{ij} - \lambda||\mathbf{x}_{ref} - \mathcal{C}_j||_2 \\ \mathcal{L}_2(j) &= \frac{\hat{W}(j, i_{ref})}{|\mathcal{X}|_p} - \lambda||\mathbf{x}_p - \mathcal{C}_j||_2 \end{aligned} \tag{A.11}$$

A.2 Deep Aggregation

The deep aggregation network, AggNet is used to compute the face descriptors for each member and aggregate multiple templates to produce a compact binary code for the group of individuals. Basically, this procedure is carried out for every group so that each group is represented by a single binary vector. Then, at verification time, a binary hash code is computed for the query using AggNet. Based on the inner product between the group representation and the query, a logistic regression classifier is used to score the membership. After this score is determined, it is compared to a threshold, and access to the system is granted if the score is greater than the threshold.

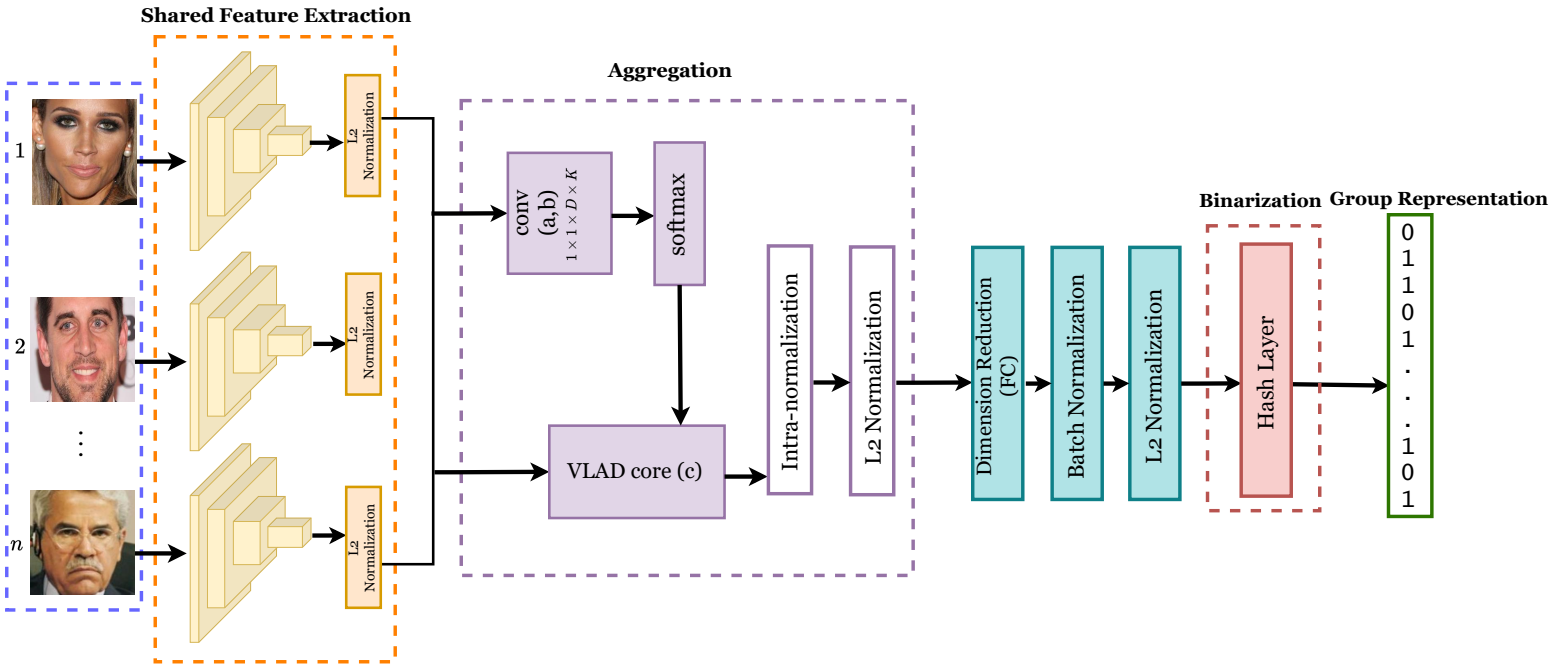


Figure A.1 – The Network architecture.

As illustrated in Figure A.1, the network architecture is made up of:

- **Feature extractor**: Each face is passed through it to extracting one descriptor for each group member.
- **Aggregator**: Multiple face descriptors into a single vector by a learnable pooling layer.
- **Hash layer**: The aggregated vector is quantized to a binary code.

Note, AggNet is used to generate both the group representation and the individual descriptors. In the following, we will discuss each component of the network in detail.

Feature Extraction

Face descriptors are extracted from input images using convolutional neural networks. We can use any network for this framework, but here we take an adapted ResNet50 architecture as the backbone. As illustrated in Figure A.2, the ResNet-50 is adapted such that the last fully connected layer has been removed and instead a fully connected layer of

size $2048 \times d$ is added after the global average pooling layer. Indeed, ResNet-50 reduces the dimensionality of face descriptors to d. We used ResNet-50 $128-D$ network pretrained on the VGGFace2. Therefore, in this part, individual descriptors $\{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ are produced, where n indicates the number of members in the group. Finally, L2 normalization is applied to each face descriptor.

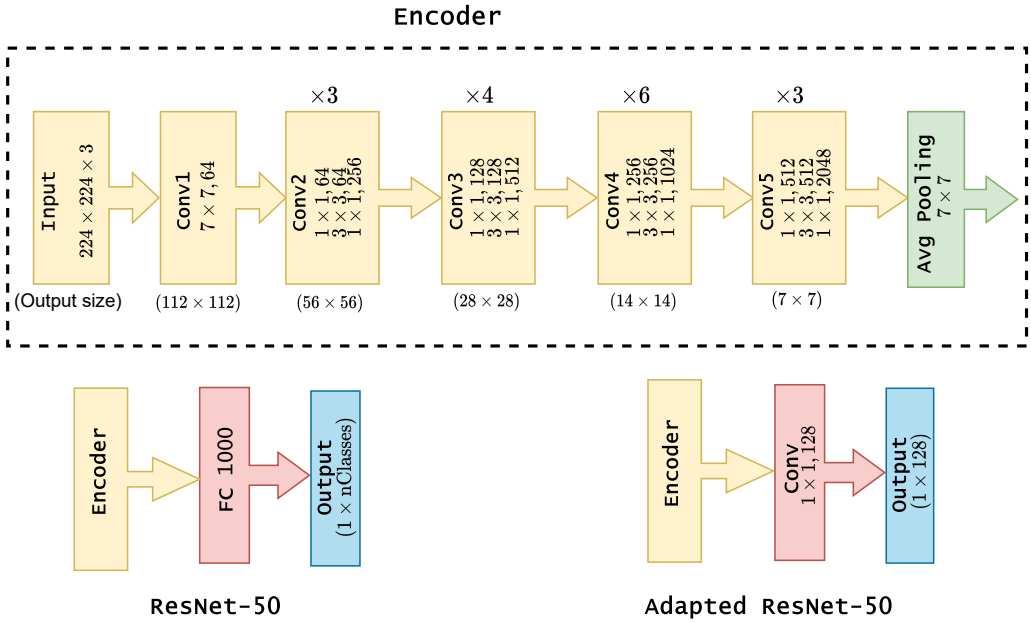


Figure A.2 – The adapted ResNet.

Aggregation

The aggregation block uses NetVLAD [AGT+16] which is a trainable pooling layer based on VLAD [JPD+12]. This block aggregates multiple face descriptors into a single $d \times K$ vector (K is the parameter of the NetVLAD). A fully-connected layer then reduces the dimension of the feature vector to d. A batch-normalization layer and L2-normalized are also used to produce the aggregated vector. NetVLAD is briefly discussed in the following.

Given n d-dimensional input descriptors $\mathbf{x}_i$ and a chosen number of clusters K, the NetVLAD layer computes a single $d \times K$ dimensional vector $\mathbf{h}_g$, which is written as a $d \times K$ matrix using the following equation:

$$H_g(j,k) = \sum_{i=1}^{n} \frac{e^{\mathbf{a}_k^T \mathbf{x}_i + b_k}}{\sum_{k'=1}^{K} e^{\mathbf{a}_{k'}^T \mathbf{x}_i + b_{k'}}} (\mathbf{x}_i(j) - \mathbf{c}_k(j)) \tag{A.12}$$

$\mathbf{a}_k$, b_k and $\mathbf{c}_k$ for $k \in \{1, \ldots, K\}$ are trainable parameters of this block and j indicates the j-th cluster. For cluster k, there is also a residual term in addition to the first term that computes the soft-assignment weight of the input vector $\mathbf{x}_i$. Then, the aggregated vector $\mathbf{h}_g$ is obtained by L2 normalization.

Hashing

The final representation should be binary rather than real-valued. The third part of the AggNet applies a binarization layer to the deep features to produce binary codes. One possibility for achieving binary representation is to post-process and convert the learned real-valued representation, for instance by applying the sign function, mapping positive values to 1 and negative values to -1 (e.g., thresholding each dimension at 0). As an alternative, the network $\mathcal{F}$ can be directly trained to produce a binary representation. A simple solution could be to add a layer at the end that applies a sign function to the outputs of $\mathcal{F}$.

However, the main challenge that prevents the training of deep hashing from being truly end-to-end is that due to binary constraints on the codes, deep hashing is basically a discrete optimization problem that cannot be directly solved by back-propagation. In other words, the use of the sign function in the last layer of a neural network to convert continuous features into binary codes result in a variant of the vanishing gradient problem since the gradient of the sign function is zero for all nonzero input and therefore conveys no information at all.

Because of this ill-posed gradient problem in the optimization with sign activations, these methods make use of relaxation or approximation. A few works have addressed the problem of training deep models under the binary constraint. To tackle this problem, we employ two techniques which are explained in detail in the following.

Greedy Hash [SZHT18] With the newly added layer, $\mathbf{b} = \mathsf{sgn}(\mathbf{h})$ is used in the forward pass, but the backward pass keeps the gradient intact, just as if the layer were the identity, *i.e.*, $\frac{\partial L}{\partial \mathbf{h}} = \frac{\partial L}{\partial \mathbf{b}}$. It turns out just making this change is enough in training $\mathcal{F}$ to produce a binary representation. It is generally necessary to initialize the weights of the network $\mathcal{F}$ from a model trained without a binarizing layer to avoid divergence during training.

HashNet [CLWY17] This method handles the non-smooth problem of the sign function by continuation. HashNet starts the training with a smoothed activation function $\tanh(\alpha_t \mathbf{h})$ and gradually reduces the smoothness as the training proceeds, *i.e.* increases α_t until it eventually becomes almost like the sign function as $\lim_{\alpha_t \to +\infty} \tanh(\alpha_t \mathbf{h}) = \mathsf{sign}(\mathbf{h})$. The operation can be modeled by a multi-stage pretraining process, i.e. training the network with $\tanh(\alpha_t \mathbf{h})$ activation function is used to initialize the network with $\tanh(\alpha_{t+1} \mathbf{h})$ activation function.

Loss Function

Assume that n identities belong to a group and the training batch consists of faces of m individuals. Once at training time, in a forward pass, the descriptors corresponding to the n individuals will be aggregated into a single binary hash code $\mathbf{b}_g$. As well, for every identity in the batch, a binary hash code, $\mathbf{b}_i$, is computed using the same network.

The association to the group is then determined for each individual by applying a logistic regression classifier to the scalar product of the two binary codes, *i.e.*:

$$score = \sigma(\theta_1 \langle \mathbf{b}_g, \mathbf{b}_i \rangle + \theta_2) \tag{A.13}$$

where $\sigma(z) = \frac{1}{1+exp(-z)}$ is a sigmoid function, and θ_1 and θ_2 are the slope and bias parameters of the logistic regression classifier, respectively. For each group member, this score should ideally be one, and for all other $m - n$ identities in the batch, it should be zero. The loss indicates how much of a difference there is between this ideal score and the actual score.

As is the case with most machine learning algorithms, the choice of loss function plays an important role. In the following, we define two different loss functions.

Weighted Cross Entropy loss For each group, the loss is defined as:

$$\mathcal{L} = \sum_{i=1}^{m} w_i \left[g_i \log(\sigma(\theta_1 \langle \mathbf{b}_g, \mathbf{b}_i \rangle + \theta_2)) + (1 - g_i) \log(1 - \sigma(\theta_1 \langle \mathbf{b}_g, \mathbf{b}_i \rangle + \theta_2)) \right] \tag{A.14}$$

where m is the size of the training batch and g_i is a binary indicator whether i-th individual belongs to the group or not and also w_i is the weight for i-th training image, which is

used to tackle the data imbalance problem as there are more negatives than positives in a batch (most g_i's are equal to 0). So the training images are weighted according to the number of individuals in the group.

Wilcoxon-Mann-Whitney loss The ideal loss function is one, which directly corresponds to the metric by which we evaluate performance (*i.e.*, AUC or $P_{tp}@P_{fp} = \alpha$). The AUC can be computed using Wilcoxon-Mann-Whitney (WMW) statistic:

$$A = \frac{\sum_{\substack{i,j=1 \\ g_i > g_j}}^{n} I\left(\sigma(\theta_1 \langle \mathbf{b}_g, \mathbf{b}_i \rangle + \theta_2) - \sigma(\theta_1 \langle \mathbf{b}_g, \mathbf{b}_j \rangle + \theta_2)\right)}{n(m-n)} \tag{A.15}$$

where $I(z) = \mathbb{1}[z > 0]$ is the unit step function.

However, in the equation above, the AUC is not a smooth function. This can be smoothed out so that it is differentiable. One way to deal with this, as discussed in the paper [YDMW03], is approximating Equation (A.15) by:

$$\mathcal{L} = \sum_{\substack{i,j=1 \\ g_i > g_j}}^{n} R(\mathbf{b}_i, \mathbf{b}_j) \tag{A.16}$$

where:

$$R(\mathbf{b}_i, \mathbf{b}_j) = \begin{cases} (-(S_{ij} - \gamma))^p & \text{if } S_{ij} < \gamma \\ 0 & \text{otherwise} \end{cases} \tag{A.17}$$

and $S_{ij} = \sigma(\theta_1 \langle \mathbf{b}_g, \mathbf{b}_i \rangle + \theta_2) - \sigma(\theta_1 \langle \mathbf{b}_g, \mathbf{b}_j \rangle + \theta_2)$. Also, $0 < \gamma \leq 1$ (usually $0.1 < \gamma \leq 0.7$) and $p > 1$ (usually 2 or 3) are hyperparameters.

Therefore, this loss function penalizes any occurrence where the group membership probability of the group members is less than the group membership probability of the other identities in the group.

Hashing with greedy strategy The optimization problem is defined as:

$$\begin{aligned} \min_{\mathbf{b}} \quad & \mathcal{L}(\mathbf{b}) \\ \text{s.t.} \quad & \mathbf{b} \in \{-1, +1\}^d \end{aligned} \tag{A.18}$$

without the binarization constraint, $\mathbf{b}$ is being updated as $\mathbf{b}^{t+1} = \mathbf{b}^t - lr * \frac{\partial \mathcal{L}}{\partial \mathbf{b}^t}$. The closest binary point to the continuous $\mathbf{b}^{t+1}$ is $\mathsf{sgn}(\mathbf{b}^{t+1})$. Thus, based on the greedy principle, we update $\mathbf{b}$ toward this value as it is probable to be the optimal binary solution in each iteration:

$$\mathbf{b}^{t+1} = \mathsf{sgn}\left(\mathbf{b}^t - lr * \frac{\partial \mathcal{L}}{\partial \mathbf{b}^t}\right) \tag{A.19}$$

Let $\mathbf{h}$ be the output of the NetVLAD layer, to deal with the binarization constrain with Greedy Hash technique, we need to split Equation (A.19) as:

$$\begin{cases} \mathbf{b}^{t+1} = \mathsf{sgn}(\mathbf{h}^{t+1}) \\ \mathbf{h}^{t+1} = \mathbf{b}^t - lr * \frac{\partial \mathcal{L}}{\partial \mathbf{b}^t} \end{cases} \tag{A.20}$$

In order to implement the first item of Equation (A.20), we need to apply sign function in the forward pass of the hash layer. For the second term, we add a penalty term $\|\mathbf{h} - \mathsf{sgn}(\mathbf{h})\|_3^3$ to the loss function $\mathcal{L}$, so that it will be as close to zero as possible. Then, we see that implementing the second item will be obtained by setting $\frac{\partial \mathcal{L}}{\partial \mathbf{h}^t} = \frac{\partial \mathcal{L}}{\partial \mathbf{b}^t}$ in the backward propagation, which means the gradient of $\mathbf{b}$ is transmitted to $\mathbf{h}$ entirely [SZHT18].

List of Figures

List of Tables

Bibliography

[ABC+15] Frederik Armknecht, Colin Boyd, Christopher Carr, Kristian Gjøsteen, Angela Jäschke, Christian A Reuter, and Martin Strand. A guide to fully homomorphic encryption. *IACR Cryptol. ePrint Arch.*, 2015:1192, 2015. 28

[AGT+16] Relja Arandjelovic, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef Sivic. Netvlad: Cnn architecture for weakly supervised place recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5297–5307, 2016. 32, 122

[AIL+15] Alexandr Andoni, Piotr Indyk, Thijs Laarhoven, Ilya P. Razenshteyn, and Ludwig Schmidt. Practical and optimal LSH for angular distance. *NIPS*, 2015. 80

[And72] James A Anderson. A simple neural network generating an interactive memory. *Mathematical biosciences*, 14(3-4):197–220, 1972. 34

[BBC+10] Mauro Barni, Tiziano Bianchi, Dario Catalano, Mario Di Raimondo, Ruggero Donida Labati, Pierluigi Failla, Dario Fiore, Riccardo Lazzeretti, Vincenzo Piuri, Fabio Scotti, et al. Privacy-preserving fingercode authentication. In *Proceedings of the 12th ACM workshop on Multimedia and security*, pages 231–240, 2010. 27

[BBL12] Giuseppe Bianchi, Lorenzo Bracciale, and Pierpaolo Loreti. "better than nothing" privacy with bloom filters: To what extent? In *Proceedings of the International Conference on Privacy in Statistical Databases*, 2012. 29

[BETVG08] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Speeded-up robust features (surf). *Computer vision and image understanding*, 110(3):346–359, 2008. 30

[BK13] M. Beck and F. Kerschbaum. Approximate two-party privacy-preserving string matching with linear complexity. In *Proceedings of the IEEE International Congress on Big Data*, 2013. 29

[BKOS07] Dan Boneh, Eyal Kushilevitz, Rafail Ostrovsky, and William E. Skeith. Public key encryption that allows pir queries. In *Proceedings of the International Cryptology Conference, Advances in Cryptology*, 2007. 29

[BL15] Artem Babenko and Victor Lempitsky. Aggregating deep convolutional features for image retrieval. *arXiv preprint arXiv:1510.07493*, 2015. 32

[Boe88] J. Boersma. Solution to problem 87-6* : The entropy of a poisson distribution. *SIAM Review*, 30(2):314–317, 1988. 76

[Bou06] T Boult. Robust distance measures for face-recognition supporting revocable biometric tokens. In *7th International Conference on Automatic Face and Gesture Recognition (FGR06)*, pages 560–566. IEEE, 2006. 26

[BSCL14] Artem Babenko, Anton Slesarev, Alexandr Chigorin, and Victor Lempitsky. Neural codes for image retrieval. In *European conference on computer vision*, pages 584–599. Springer, 2014. 32

[CLWY17] Zhangjie Cao, Mingsheng Long, Jianmin Wang, and Philip S Yu. Hashnet: Deep learning to hash by continuation. In *Proceedings of the IEEE international conference on computer vision*, pages 5608–5617, 2017. 124

[CMKV16] Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, and Andrea Vedaldi. Deep filter banks for texture recognition, description, and segmentation. *International Journal of Computer Vision*, 118(1):65–94, 2016. 32

[DRS04] Yevgeniy Dodis, Leonid Reyzin, and Adam Smith. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. In *International conference on the theory and applications of cryptographic techniques*, pages 523–540. Springer, 2004. 24

[EFG+09] Zekeriya Erkin, Martin Franz, Jorge Guajardo, Stefan Katzenbeisser, Inald Lagendijk, and Tomas Toft. Privacy-preserving face recognition. In *Proceedings of the International Symposium on Privacy Enhancing Technologies*, 2009. 27

[Elg85] T. Elgamal. A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Transactions on Information Theory*, 31(4):469–472, July 1985. 68

[FCAB00] Li Fan, Pei Cao, Jussara Almeida, and Andrei Z Broder. Summary cache: a scalable wide-area web cache sharing protocol. *IEEE/ACM transactions on networking*, 8(3):281–293, 2000. 44

[GB09] Craig Gentry and Dan Boneh. *A fully homomorphic encryption scheme*, volume 20. Stanford university Stanford, 2009. 27

[GBMG+17] Marta Gomez-Barrero, Emanuele Maiorana, Javier Galbally, Patrizio Campisi, and Julian Fierrez. Multi-biometric template protection based on homomorphic encryption. *Pattern Recognition*, 67:149–163, 2017. 27

[GBRL+18] Marta Gomez-Barrero, Christian Rathgeb, Guoqiang Li, Raghavendra Ramachandra, Javier Galbally, and Christoph Busch. Multi-biometric template protection based on bloom filters. *Information Fusion*, 42:37–50, 2018. 29

[GWGL14] Yunchao Gong, Liwei Wang, Ruiqi Guo, and Svetlana Lazebnik. Multi-scale orderless pooling of deep convolutional activation features. In *European conference on computer vision*, pages 392–407. Springer, 2014. 32

[HAD06] Feng Hao, Ross Anderson, and John Daugman. Combining crypto with biometrics effectively. *IEEE transactions on computers*, 55(9):1081–1088, 2006. 24

[Heb49] Donald Olding Hebb. *The organization of behavior: a neuropsychological theory*. J. Wiley; Chapman & Hall, 1949. 34

[HMBLM08] Gary B Huang, Marwan Mattar, Tamara Berg, and Eric Learned-Miller. Labeled faces in the wild: A database forstudying face recognition in unconstrained environments. In *Workshop on faces in'Real-Life'Images: detection, alignment, and recognition*, 2008. 53

[Hop82] John J Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the national academy of sciences*, 79(8):2554–2558, 1982. 34

[HZRS16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 104

[IFG+17] Ahmet Iscen, Teddy Furon, Vincent Gripon, Michael Rabbat, and Hervé Jégou. Memory vectors for similarity search in high-dimensional spaces. *IEEE Transactions on Big Data*, 2017. 33, 42, 44, 47

[IW09] Tanya Ignatenko and Frans Willems. Achieving secure fuzzy commitment scheme for optical pufs. In *2009 Fifth International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, pages 1185–1188. IEEE, 2009. 24

[JLK+06] MinYi Jeong, Chulhan Lee, Jongsun Kim, Jeung-Yoon Choi, Kar-Ann Toh, and Jaihie Kim. Changeable biometrics for appearance based face recognition. In *2006 biometrics symposium: special session on research at the biometric consortium conference*, pages 1–5. IEEE, 2006. 26

[JPD+12] Hervé Jégou, Florent Perronnin, Matthijs Douze, Jorge Sánchez, Patrick Pérez, and Cordelia Schmid. Aggregating local image descriptors into compact codes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(9):1704–1716, 2012. 30, 31, 122

[JS06] Ari Juels and Madhu Sudan. A fuzzy vault scheme. *Designs, Codes and Cryptography*, 38(2):237–257, 2006. 24

[JW99] Ari Juels and Martin Wattenberg. A fuzzy commitment scheme. In *Proceedings of the 6th ACM conference on Computer and communications security*, pages 28–36, 1999. 24

[Koh72] Teuvo Kohonen. Correlation matrix memories. *IEEE transactions on computers*, 100(4):353–359, 1972. 34

[Kos88] Bart Kosko. Bidirectional associative memories. *IEEE Transactions on Systems, man, and Cybernetics*, 18(1):49–60, 1988. 34

[Laa15] Thijs Laarhoven. *Search problems in cryptography From fingerprinting to lattice sieving*. PhD thesis, Eindhoven University of Technology, 2015. 75, 78

[LEB12] Reginald L Lagendijk, Zekeriya Erkin, and Mauro Barni. Encrypted signal processing for privacy protection: Conveying the utility of homomorphic

encryption and multiparty computation. *IEEE Signal Processing Magazine*, 30(1):82–105, 2012. 27

[Low04] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004. 30

[LSM06] Qiming Li, Yagiz Sutcu, and Nasir Memon. Secure sketch for biometric templates. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 99–113. Springer, 2006. 24

[MMO+16] Eva Mohedano, Kevin McGuinness, Noel E O'Connor, Amaia Salvador, Ferran Marques, and Xavier Giro-i Nieto. Bags of local convolutional features for scalable instance search. In *Proceedings of the 2016 ACM on International Conference on Multimedia Retrieval*, pages 327–331, 2016. 32

[oSIoA] Chinese Academy of Sciences' Institute of Automation. Casia-irisv1 iris image database [online]. *Available: http://www.cbsr.ia.ac.cn/IrisDatabase.htm*. 64

[Pai99a] Pascal Paillier. Public-key cryptosystems based on composite degree residuosity classes. In *International conference on the theory and applications of cryptographic techniques*, pages 223–238. Springer, 1999. 27

[Pai99b] Pascal Paillier. Public-key cryptosystems based on composite degree residuosity classes. In Jacques Stern, editor, *Advances in Cryptology — EUROCRYPT '99*, pages 223–238, Berlin, Heidelberg, 1999. Springer Berlin Heidelberg. 68

[PD07] F. Perronnin and C. Dance. Fisher kernels on visual vocabularies for image categorization. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, 2007. 30

[PPCR10] Jaishanker K Pillai, Vishal M Patel, Rama Chellappa, and Nalini K Ratha. Sectored random projections for cancelable iris biometrics. In *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 1838–1841. IEEE, 2010. 25

[PPCR11] Jaishanker K Pillai, Vishal M Patel, Rama Chellappa, and Nalini K Ratha. Secure and robust iris recognition using random projections and sparse representations. *IEEE transactions on pattern analysis and machine intelligence*, 33(9):1877–1893, 2011. 25

[PVZ+15] Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, et al. Deep face recognition. In *Proceedings of the British Machine Vision Conference*, 2015. 53, 63, 84

[RCB01] Nalini K. Ratha, Jonathan H. Connell, and Ruud M. Bolle. Enhancing security and privacy in biometrics-based authentication systems. *IBM systems Journal*, 40(3):614–634, 2001. 25

[RGBB+15] Christian Rathgeb, Marta Gomez-Barrero, Christoph Busch, Javier Galbally, and Julian Fierrez. Towards cancelable multi-biometrics based on bloom filters: a case study on feature level fusion of face and iris. In *3rd international workshop on biometrics and forensics (IWBF 2015)*, pages 1–6. IEEE, 2015. 29

[RSCM16] Ali S Razavian, Josephine Sullivan, Stefan Carlsson, and Atsuto Maki. Visual instance retrieval with deep convolutional networks. *ITE Transactions on Media Technology and Applications*, 4(3):251–258, 2016. 32

[RU10] Christian Rathgeb and Andreas Uhl. Adaptive fuzzy commitment scheme based on iris-code error analysis. In *2010 2nd European Workshop on Visual Information Processing (EUVIP)*, pages 41–44. IEEE, 2010. 24

[RV18] Behrooz Razeghi and Slava Voloshynovskiy. Privacy-preserving outsourced media search using secure sparse ternary codes. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, 2018. 41, 49

[RVKT17] Behrooz Razeghi, Slava Voloshynovskiy, Dimche Kostadinov, and Olga Taran. Privacy preserving identification using sparse approximation with ambiguization. In *Proceedings of the IEEE International Workshop on Information Forensics and Security*, 2017. 26, 41, 49, 51

[SCC+16] Soumyadip Sengupta, Jun-Cheng Chen, Carlos Castillo, Vishal M Patel, Rama Chellappa, and David W Jacobs. Frontal to profile face verification in the wild. In *Proceeding of the IEEE Winter Conference on Applications of Computer Vision*, 2016. 53

[Sch66] Peter H. Schönemann. A generalized solution of the orthogonal procrustes problem. *Psychometrika*, 31(1):1–10, 1966. 51, 62

[Sha59] C. E. Shannon. Probability of error for optimal codes in a gaussian channel. *Bell System Tech. J.*, 38:611–656, 1959. 74

[SKP15] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823, 2015. 102

[SPH99] Stuart Schechter, Todd Parnell, and Alexander Hartemink. Anonymous authentication of membership in dynamic groups. In *Proceedings of the International Conference on Financial Cryptography*, 1999. 24

[SRASC14] Ali Sharif Razavian, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson. Cnn features off-the-shelf: an astounding baseline for recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 806–813, 2014. 32

[SSW10] Ahmad-Reza Sadeghi, Thomas Schneider, and Immo Wehrenberg. Efficient privacy-preserving face recognition. In *Proceedings of the International Conference on Information, Security and Cryptology*, 2010. 27

[SZ03] J. Sivic and A. Zisserman. Video google: a text retrieval approach to object matching in videos. In *Proceedings of the IEEE International Conference on Computer Vision*, 2003. 30

[SZHT18] Shupeng Su, Chao Zhang, Kai Han, and Yonghong Tian. Greedy hash: Towards fast optimization for accurate hash coding in cnn. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 806–815, 2018. 123, 126

[TB99] Michael E Tipping and Christopher M Bishop. Probabilistic principal component analysis. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 61(3):611–622, 1999. 84

[TBS14] Wilson Abel Alberto Torres, Nandita Bhattacharjee, and Bala Srinivasan. Effectiveness of fully homomorphic encryption to preserve the privacy of biometric data. In *Proceedings of the 16th International Conference on Information Integration and Web-based Applications & Services*, pages 152–158, 2014. 28

[TG10] Carlos Eduardo Thomaz and Gilson Antonio Giraldi. A new ranking method for principal components analysis and its application to face image analysis. *Image and Vision Computing*, 28(6):902–913, 2010. 53

[TGN06] Andrew BJ Teoh, Alwyn Goh, and David CL Ngo. Random multispace quantization as an analytic mechanism for biohashing of biometric and random identity inputs. *IEEE transactions on pattern analysis and machine intelligence*, 28(12):1892–1901, 2006. 25

[TSJ15] Giorgos Tolias, Ronan Sicre, and Hervé Jégou. Particular object retrieval with integral max-pooling of cnn activations. *arXiv preprint arXiv:1511.05879*, 2015. 32

[Uni] The Multimedia University. Mmu2 iris image database [online]. *Available: http://pesona.mmu.edu.my/ ccteo/*. 64

[UWG+19] Ravitej Uppu, Tom AW Wolterink, Sebastianus A Goorden, Bin Chen, Boris Škorić, Allard P Mosk, and Pepijn WH Pinkse. Asymmetric cryptography with physical unclonable keys. *Quantum Science and Technology*, 4(4):045011, 2019. 95

[VS08] Roberto A Vazquez and Humberto Sossa. Associative memories applied to pattern recognition. In *International Conference on Artificial Neural Networks*, pages 111–120. Springer, 2008. 34

[YDMW03] Lian Yan, Robert H Dodier, Michael Mozer, and Richard H Wolniewicz. Optimizing classifier performance via an approximation to the wilcoxon-mann-whitney statistic. In *Proceedings of the 20th international conference on machine learning (icml-03)*, pages 848–855, 2003. 125

[ZRC08] Jinyu Zuo, Nalini K Ratha, and Jonathan H Connell. Cancelable iris biometric. In *2008 19th International Conference on Pattern Recognition*, pages 1–4. IEEE, 2008. 26

DOCTORAT BRETAGNE LOIRE / MATHSTIC

UNIVERSITÉ DE RENNES 1

Titre : Identification sécurisée pour Internet des objets

Mot clés : Représentation de groupe, plongement, agrégation, vérification, sécurité, confidentialité.

Résumé : Cette thèse aborde le problème de l'authentification des dispositifs à faible puissance dans l'Internet des objets en introduisant de nouvelles fonctionnalités : la vérification de l'appartenance à un groupe et l'identification. La procédure vérifie si un dispositif IoT donné est membre d'un groupe sans révéler l'identité de ce membre. De même, l'identification de l'appartenance à un groupe indique à quel groupe le dispositif appartient sans connaître son identité. Nous proposons un protocole par l'utilisation conjointe de deux mécanismes : la quantification des motifs dans des plongement discrets, rendant la reconstruction difficile, et l'agrégation de plusieurs motifs dans une représentation de groupe, entravant l'identification. Tout d'abord, nous considérons deux procédures indépendantes, l'une pour l'plongement, l'autre pour l'agrégation. Ensuite, nous remplaçons ces fonctions déterministes par des fonctions dont les paramètres sont appris par optimisation. Enfin, plutôt que de considérer des affectations de groupes prédéterminées, les affectations de groupes sont également apprises avec les représentations des groupes. Nos expériences montrent que l'apprentissage permet un excellent compromis entre les performances de sécurité/confidentialité et de vérification/identification. Nous étudions également l'impact du niveau de sparsité des fonctionnalités représentant les membres du groupe sur les performances de sécurité et de vérification. Nous montrons qu'il est possible d'échanger la compacité et la sparsité pour une meilleure sécurité ou de meilleures performances de vérification.

Title: Secure identification for the Internet of Things

Keywords: Group Representation, Embedding, Aggregation, Verification, Security, Privacy.

Abstract: This thesis addresses the problem of authentication of low power devices in the Internet of Things by introducing new functionalities: group membership verification and identification. The procedure verifies if a given IoT device is a member of a group without revealing the identity of that member. Similarly, group membership identification states which group the device belongs to without knowing the identity. We propose a protocol through the joint use of two mechanisms: quantizing templates into discrete embeddings, making reconstruction difficult, and aggregating several templates into one group representation, impeding identification. First, we consider two independent procedures, one for embedding, the other for aggregating. Then, we replace those deterministic functions with functions whose parameters are learned through optimization. Finally, rather than considering group assignments that are predetermined, group assignments are also learned together with representations of the groups. Our experiments show that learning yields an excellent trade-off between security/privacy and verification/identification performances. We also investigate the impact of the sparsity level of the features representing group members on both security and verification performances. It shows it is possible to trade compactness and sparsity for better security or better verification performance.