**POLYTECHNIQUE MONTRÉAL**
affiliée à l'Université de Montréal

# Secure Authentication for Mobile Users

**SEPEHR KEYKHAIE**
Département de génie informatique et génie logiciel

Thèse présentée en vue de l'obtention du diplôme de *Philosophiæ Doctor*
Génie informatique

Janvier 2021

**POLYTECHNIQUE MONTRÉAL**
affiliée à l'Université de Montréal

Cette thèse intitulée :

**Secure Authentication for Mobile Users**

présentée par **Sepehr KEYKHAIE**
en vue de l'obtention du diplôme de *Philosophiæ Doctor*
a été dûment acceptée par le jury d'examen constitué de :

**Martine BELLAÏCHE**, présidente
**Samuel PIERRE**, membre et directeur de recherche
**Soumaya CHERKAOUI**, membre
**Abderrahim BENSLIMANE**, membre externe

## DEDICATION

*To my parents Mehri and Hossein,*
*for their self-sacrifice...*

*To my companion and my spouse Mahsa,*
*for her endless love and patience...*

*To my sister Setareh, and my brother Soheil,*
*from whom I learned a lot...*

*and,*
*To my lovely daughter Selena,*
*the only reason for my happiness...*

## ACKNOWLEDGEMENTS

First of all, I wish to express my sincere appreciation to my research director Professor Samuel Pierre, for his professional supervision, his affableness, and his academic and financial support during my Ph.D. study. I really thank you for the confidence you had in my work, and for giving me this invaluable opportunity to raise new scientific skills in your lab, to grow my perspective wider and to gain more insight into things. Your achievements, your attitude, your humanity, and your modesty are exemplary in my life.

My sincere gratitude goes to Mr. Mostafa Chafi, the CEO of Flex Group company, for giving me the opportunity to join his team of professionals and learn more practical concepts in telecommunications. Thank you for your all help and support.

I would also like to thank my colleagues at Flex Group, Mr. Cristian Penaloza, and Mr. Omid Ghari, the project managers of the company, for their friendly attitude alongside their professional administration during my work in the company. Particularly, Omid for helping me with the French review. I wish to extend my thanks to Mr. Salim Melaz, the senior Card developer, for his endless support, his patience, his friendliness, and for sharing his knowledge with me. Without his precious support this research would take much longer time.

I would also like to thank Mrs. Martine Bellaïche, Mrs. Soumaya Cherkaoui, and Mr. Abderrahim Benslimane for their interest in my research and accepting to participate in the jury.

I cannot forget my colleagues and friends at the Mobile Computing and Network Research Laboratory (LARIM) for making the lab a suitable place to concentrate and work collaboratively in a warm and friendly atmosphere. Thanks to Dre. Franjieh El Khoury, the project supervisor, for her unconditional help and support, and for making LARIM like a family. Thanks to Lamia, Farnoush, Nasrin, Olson, Claudy, Loic, Sanaz, and Parand for their collaboration, and their constructive comments on my research. Special thank goes to my friend Amir, for not only being a good colleague but also a supportive friend. I will not forget long discussions we have about life and research at lunch breaks.

I am forever indebted to my parents, Hossein and Mehri for their continuous support, love, and the confidence they placed in me. Words are not enough to thank them. Their prayers have always been with me. Thanks to my sister and brother for being good friends, and pushing me forward during my difficult times in the life.

My deepest appreciation belongs to my companion, my friend, and my spouse Mahsa. Your

endless love, your patience, and your humanity have made you more beautiful and my life more delightful.

Finally, I wish to give special heartfelt thanks to my little angel Selena for her presence in my life, my heart beating outside of my chest, a true source of happiness, innocence, and love. Success is not out of my reach when I look at your beautiful eyes.

# RÉSUMÉ

L'authentification biométrique telle que les empreintes digitales et la biométrie faciale a changé la principale méthode d'authentification sur les appareils mobiles. Les gens inscrivent facilement leurs modèles d'empreintes digitales ou de visage dans différents systèmes d'authentification pour profiter de leur accès facile au smartphone sans avoir besoin de se souvenir et de saisir les codes PIN/mots de passe conventionnels. Cependant, ils ne sont pas conscients du fait qu'ils stockent leurs caractéristiques physiologiques ou comportementales durables sur des plates-formes non sécurisées (c'est-à-dire sur des téléphones mobiles ou sur un stockage en nuage), menaçant la confidentialité de leurs modèles biométriques et de leurs identités. Par conséquent, un schéma d'authentification est nécessaire pour préserver la confidentialité des modèles biométriques des utilisateurs et les authentifier en toute sécurité sans compter sur des plates-formes non sécurisées et non fiables.

La plupart des études ont envisagé des approches logicielles pour concevoir un système d'authentification sécurisé. Cependant, ces approches ont montré des limites dans les systèmes d'authentification sécurisés. Principalement, ils souffrent d'une faible précision de vérification, en raison des transformations du gabarit (cancelable biometrics), de la fuite d'informations (fuzzy commitment schemes) ou de la réponse de vérification non en temps réel, en raison des calculs coûteux (homomorphic encryption).

Au regard de tout ce qui précède, cette thèse vise à concevoir un nouveau schéma d'authentification sécurisé et préservant la confidentialité pour les utilisateurs mobiles. Nous utilisons la technique match-on-card (MOC) sur les appareils mobiles qui en profite des caractéristiques de sécurité matérielle des cartes de module d'identité d'abonné (SIM) ou de carte SIM intégrée (eSIM) disponibles sur tous les smartphones. Toutefois, en raison des limitations de ressources des cartes SIM, la conception d'un système d'authentification préservant la confidentialité en utilisant la technique MMOC est une tâche difficile à accomplir. Afin d'atteindre cet objectif, le travail a été réparti sur trois principales phases.

Tout d'abord, compte tenu de la plus grande sécurité des systèmes d'authentification active, nous proposons un système d'authentification active comportementale sécurisé pour les utilisateurs mobiles utilisant la biométrie tactile. Une architecture assistée par le cloud est proposée, où l'interaction de l'utilisateur avec le dispositif à écran tactile est surveillée de manière transparente en arrière-plan, et les caractéristiques les plus discriminantes de chaque coup tel que la vitesse, l'accélération, la pression sur l'écran, la zone couverte, la - la distance de fin, la durée de course sont extraites et stockées sur la carte SIM/eSIM pour au-

thentification. Un réseau neuronal profond quantifié (quantized DNN) est implémenté sur la carte SIM/eSIM pour la vérification de l'utilisateur avec un module de formation modèle sur un serveur cloud. Cette architecture nous aide à augmenter la précision des performances tout en augmentant la sécurité et la confidentialité du système. De plus, une technique d'optimisation du compilateur est également employée pour accélérer le passage avant du DNN sur la carte.

Dans la deuxième phase, nous étendons notre système d'authentification active sécurisé aux traits biométriques avec des vecteurs de caractéristiques plus grands. La biométrie physiologique telle que la biométrie faciale peut également être utilisée dans l'authentification active. Nous utilisons les méthodes d'apprentissage par transfert (transfer learning) et d' extraction profonde des caractéristiques (deep feature extraction) pour générer une représentation profonde du visage qui est robuste contre les changements de pose et d'illumination, et surtout, elle est assez légère pour être migrée vers SIM/eSIM pour la vérification finale. Une authentification active sur carte est proposée pour authentifier en permanence les utilisateurs légitimes. Nous proposons deux systèmes d'authentification pour l'authentification active basée sur le visage. Le premier qui est une authentification basée sur un modèle (model-based authentication) utilise les ressources du cloud à des fins de sélection de modèle et de formation, tandis que le second qui est une authentification basée sur un gabarit (template-based authentication) ne dépend que des ressources SIM/eSIM pour l'inscription et la vérification, étant plus sécurisé que la première architecture. Lors de la phase d'inscription sur la carte, la distance L2 est utilisée pour trouver la distance entre le modèle d'ancrage et d'autres modèles négatifs ou positifs, et le meilleur seuil de classification est obtenu. Une méthode numérique est proposée pour calculer la distance L2 sur la carte.

Enfin, un système générique d'authentification de réservation de confidentialité basé sur MMOC est présenté. L'apprentissage par transfert en utilisant des approches basées sur l'apprentissage en profondeur a considérablement amélioré les performances des systèmes de reconnaissance et peut également être utilisé dans les systèmes d'authentification biométriques. Nous utilisons l'apprentissage par transfert pour créer une architecture de réseau spécialisée pour l'extraction et la vérification des fonctionnalités sur la carte. Nous modifions l'architecture d'un modèle préentraîné et le peaufinons pour le rendre adapté à l'implémentation SIM/eSIM. Un nouveau schéma de quantification et une architecture d'optimisation sont proposés pour réduire le temps d'exécution du sous-réseau de classification sur la carte tout en maintenant la précision des performances proche du modèle à valeur réelle.

Les performances du système d'authentification proposé sont évaluées à l'aide d'ensembles de données accessibles au public. Grâce à des expériences approfondies, nous montrons que

le système proposé avec un bon schéma de quantification et une architecture d'optimisation efficace atteint une grande précision en temps réel avec une faible empreinte mémoire sur les cartes à puce, et convient à l'authentification multiplateforme. Une implémentation sur de vrais smartphones montre également que le système a moins de surcharge de performances que même une simple méthode d'authentification sécurisée basée sur le cryptage.

En résumé, cette thèse est le premier travail qui étudie le potentiel de la technique MMOC en tant que système d'authentification sécurisé, léger et en temps réel pour les smartphones, même en utilisant des classificateurs informatiques coûteux tels que les réseaux neuronaux profonds.

# ABSTRACT

Biometric authentication such as fingerprint and face biometrics has changed the main authentication method on mobile devices. People easily enroll their fingerprint or face templates on different authentication systems to take advantage of their easy access to the smartphone with no need to remember and enter the conventional PINs/passwords. However, they are not aware that they store their long-lasting physiological or behavioral characteristics on insecure platforms (i.e., on mobile phones or on cloud storage), threatening the privacy of their biometric templates and their identities. Therefore, an authentication scheme is required to preserve the privacy of users' biometric templates and securely authenticate them without relying on insecure and untrustworthy platforms.

Most studies have considered software-based approaches to design a privacy-reserving authentication system. However, these approaches have shown limitations in secure authentication systems. Mainly, they suffer from low verification accuracy, due to the template transformations (in cancelable biometrics), information leakage (in fuzzy commitment schemes), or non real-time verification response, due to the expensive computations (in homomorphic encryption).

To this end, this thesis aims to design a new secure and privacy-preserving authentication scheme for mobile users. We use match-on-card (MOC) technique on mobile devices that takes advantage of hardware security characteristics of subscriber identity module (SIM) or embedded SIM (eSIM) cards available on all smartphones. However, due to the resource limitations of SIM cards, designing an authentication system that preserves privacy using MMOC technique is a difficult task to accomplish. In order to achieve this goal, the work has been divided into three main phases.

First, considering the higher security of active authentication systems, we propose a secure behavioral active authentication system for mobile users using touchscreen biometric. A cloud-assisted architecture is proposed, where the user's interaction with the touchscreen device is monitored transparently in the background, and the most discriminative features of each stroke such as velocity, acceleration, pressure on the screen, the covered area, direct end-to-end distance, stroke duration are extracted and stored on the SIM/eSIM card for authentication. A quantized deep neural network (DNN) is implemented on the SIM/eSIM card for user verification with a model training module on a cloud server. This architecture helps us to increase the performance accuracy while increasing the security and privacy of the system. Moreover, a compiler optimization technique is also employed to speed up the

forward pass of DNN on the card.

In the second phase, we extend our secure active authentication system to biometric traits with larger feature vectors. Physiological biometric such as face biometric also can be used in active authentication. We employ transfer learning and deep feature extraction methods to generate a deep representation of the face that is robust against pose and illumination changes, more importantly, lightweight enough to be migrated to the SIM/eSIM for final verification. An on-card active authentication is proposed to continuously authenticate legitimate users. We propose two authentication systems for face-based active authentication. The first one which is a model-based authentication uses the cloud resources for model selection and training purposes while the latter one which is a template-based authentication only relies on SIM/eSIM resources for enrollment and verification, and is more secure than the first architecture. In the enrollment phase on the card, L2 distance is used to find the distance between the anchor template and other negative or positive templates, and the best classification threshold is obtained. A numerical method is proposed to compute L2 distance on the card.

Finally, a generic MMOC-based privacy-preserving authentication system is presented. Transfer learning by using deep learning-based approaches has drastically improved the performance of recognition systems and can be used in biometric-based authentication systems as well. We use transfer learning to build specialized network architecture for feature extraction and verification on the card. We modify the architecture of a pre-trained model and fine-tune it to make it suitable for SIM/eSIM implementation. A novel quantization scheme and an optimization architecture is proposed to reduce the classification sub-network execution time on the card while keeping the performance accuracy close to the real-valued model.

The performance of the proposed authentication system is evaluated using publicly available datasets. With extensive experiments, we show that the proposed system with a good quantization scheme and an efficient optimization architecture achieves high accuracy in real-time with a small memory footprint on smart cards, and is suitable for cross-platform authentication. An implementation on real smartphones also shows that the system has less performance overhead compared to even a simple encryption-based secure authentication method.

In summary, this thesis is the first work that studies the potential of the MMOC technique as a secure, privacy-preserving, lightweight, and real-time authentication system on smartphones, even using computationally expensive classifiers such as deep neural networks.

# TABLE OF CONTENTS

# LIST OF TABLES

# LIST OF FIGURES

# LIST OF SYMBOLS AND ACRONYMS

| | |
|---|---|
| AA | Active Authentication |
| AES | Advanced Encryption Standard |
| AKA | Authentication and Key Agreement |
| APDU | Application Protocol Data Unit |
| API | Application Programming Interface |
| ARA | Access Control Applet |
| AUC | Area Under Curve |
| BIM | Biologically Inspired Model |
| CA | Continuous Authentication |
| CA-MMOC | Cloud-Assisted Mobile Match On Card |
| CE | Consumer Electronics |
| CNN | Convolutional Neural Network |
| CO | Cloud Operator |
| COS | Comprehensive Optimization Strategy |
| CPU | Central Processing Unit |
| CRC-RLS | Collaborative Representation Classifier with Regularized Least Squares |
| CompCode | Competitive Code |
| CS-LDA | Client Specific Linear Discriminant Analysis |
| DCNN | Deep Convolutional Neural Network |
| DMIPS | Dhrystone Million Instructions Per Second |
| DNN | Deep Neural Network |
| DTW | Dynamic Time Warping |
| DWT | Discrete Wavelet Transformation |
| ECC | Elliptic Curve Cryptography |
| ECC | Error Correcting Code |
| ECDH | Elliptic Curve Diffie-Hellman |
| EEPROM | Electrically Erasable Programmable Read-Only Memory |
| EER | Equal Error Rate |
| EOR | Extremal Openset Rejection |
| eSE | Embedded Secure Element |
| eSIM | Embedded Subscriber Identity Module |
| F-MMOC | Full Mobile Match On Card |
| FPR | False Positive Rate |

| | |
|---|---|
| FPU | Floating Point Unit |
| FNR | False Negative Rate |
| GA | Genetic Algorithm |
| GC | Garbled Circuits |
| GAR | Genuine Acceptance Rate |
| GLM | Generalized Linear Model |
| GMM | Gaussian Mixture Model |
| HC | Homomorphic Cryptosystem |
| HCE | Host Card Emulation |
| HMM | Hidden Markov model |
| HOG | Histogram of Oriented Gradient |
| HSM | Hardware Security Module |
| IC | Integrated Circuit |
| ICCID | Integrated Circuit Card IDentifier |
| IMSI | International Mobile Subscriber Identity |
| I/O | Input and Output |
| IOT | Internet Of Thing |
| iSIM | integrated Subscriber Identity Module |
| JCRE | Java Card Runtime Environment |
| JCVM | Java Card Virtual Machine |
| k-NN | k-Nearest-Neighbor |
| LBP | Local Binary Pattern |
| LDA | Linear Discriminant Analysis |
| LFW | Labeled Faces in the Wild |
| LSTM | Long Short-Term Memory |
| LSVM | Linear Support Vector Machine |
| LR | Logistic Regression |
| MITM | Man In The Middle |
| MLP | MultiLayer Perceptron |
| MNO | Mobile Network Operator |
| MMOC | Mobile Match on Card |
| MOC | Match On Card |
| MOH | Match On Host |
| MTCNN | Multitask Cascaded Convolutional Network |
| NB | Naïve Bayes |
| NN | Nearest Neighbor |

| | |
|---|---|
| OEM | Original Equipment Manufacturer |
| OS | Operating System |
| OSN | Online Social Network |
| OTA | Over The Air |
| PIN | Personal Identification Number |
| PKI | Public Key Infrastructure |
| PRE | Precision |
| QIM | Quantization Index Modulation |
| QR | Quick Response |
| RAM | Read-Access Memory |
| RBF | Radial Basis Function |
| REC | Recall |
| ReLU | Rectified Linear Unit |
| ROC | Receiver Operating Characteristic |
| ROI | Region Of Interest |
| ROM | Read-Only Memory |
| RR | Recognition Rate |
| RSA | Rivest–Shamir–Adleman |
| RSP | Remote SIM Provisioning |
| SaaSE | SIM As A Secure Element |
| SBB | Social Behavior Biometric |
| SC | Smart Card |
| SE | Secure Element |
| SETA | Skin Elasticity Tolerant Algorithm |
| SHA | Secure Hash Algorithm |
| SIM | Subscriber Identity Module |
| SNA | Social Network Analysis |
| SOC | System-On-A-Chip |
| SPN | Sensor Pattern Noise |
| SR | Specular Reflection |
| SVDE | Support Vector Distribution Estimation |
| SVM | Support Vector Machine |
| SVM-GRBF | SVM with Gaussian radial basis function |
| TEE | Trusted Execution Environment |
| TPR | True Positive Rate |
| TRN | Tokenized Random Number |

| | |
|---|---|
| VAS | Value Added Service |
| VR | Virtual Reality |

# CHAPTER 1 INTRODUCTION

With the increasing usage of smartphones, a robust authentication system is required to prevent any unauthorized access to personal and sensitive information stored on the device. Traditional authentication schemes such PIN/password or graphical patterns suffer from two attacks namely *shoulder surfing attack* and *smudge attacks* [1]. Shoulder surfing attacks could happen in public places and public transportation where people can easily spy your PIN/password or your touch screen patterns. In smudge attacks, an attacker can take advantage of the residues left on the device after drawing a touch screen pattern. Moreover, recent studies show that many users do not take security issues seriously for accessing their smartphones. About 46.8% of users believe that unlocking their phones is time consuming and inconvenient and prefer to leave their mobile devices unprotected [2]. A study shows that 35.5% of users never lock their smartphones using PIN/password or lock patterns [3]. Another study shows that 67% of users use a simple PIN method to lock their devices [4]. About 34% of users in the united states do not use any authentication on their smartphones [5]. In recent years, biometric authentication has attracted attentions from academia and industry. Biometric authentication uses physiological characteristics such as fingerprint, face recognition, etc., or behavioral characteristics of users such as gait recognition, signature, gesture recognition on touchscreen devices, or a hybrid scheme that takes advantage of the both systems [6]. Moreover, new biometric authentication systems continuously and unobtrusively authenticate users to eliminate any attacks to the system even after a successful login.

Biometric templates are unique and long-lasting characteristics of their owners, and unlike traditional authentication methods such as PIN/password or graphical patterns, cannot be changed if compromised. Therefore, the storage and verification of biometric templates are big concerns in mobile authentication systems. Most of the mobile authentication systems store biometric templates on cloud storage or on smartphones both threaten the privacy of users' identities. Around 75% of users download applications from official repositories who believe that downloaded applications from these repositories are secure [7]. However, studies show that security control and application testing are not enabled in all official repositories or are inadequate [8]. Moreover, due to the increasing number of applications, validation techniques become more and more complex. On the other hand, not all users are knowledgeable and even are not aware of the consequences of installing a spyware or a trojan on their phones. Interestingly, only 36% of smart phone users consider themselves as responsible for the security of their devices and the sensitive information stored on them [9]. McAfee reports the increasing number of banking trojans that take advantage of Android vulnerabilities [10].

These statistics reveal the importance of a secure biometric authentication for mobile users.

*Software-based* privacy-preserving biometric authentication approaches such as fuzzy commitment [11] or homomorphic encryption [12] store biometric templates securely on mobile devices. In the first approach, the verification accuracy would degrade due to template transformation. In the latter method, computing the matching score on encrypted templates using homomorphic encryption affects the real-time response of the authentication system owing to the heavy computation on cipher text space. *Hardware-based* approaches used by Original Equipment Manufacturers (OEMs) or processor design companies use a Trusted Execution Environment (TEE) with supporting hardware (e.g., ARM TrustZone [13]) and a trusted OS (e.g., Trusty [14]) for isolation of user verification phase, and secure storage of biometric templates. However, this approach needs a special hardware design; moreover, the TEE is not available on all mobile devices, or is very costly to buy the required SDK.

In this thesis, we pursue a hardware-based secure biometric authentication system using a new concept Mobile Match on Card (MMOC) authentication that takes advantage of smart cards (SCs) in the form of Subscriber Identity Module (SIM) or the newly evolved technology, embedded SIM (eSIM) cards, available on all smartphones to design a privacy-preserving and secure authentication system for mobile users without relying on any specific hardware design.

The remainder of this chapter is organized as follows. Section 1.1 introduces basic concepts and definitions necessary to understand the conducted research in this thesis properly. Section 1.2 discusses secure biometric authentication approaches. In section 1.3, open problems in authentication systems are investigated. Then, we set our research objectives in section 1.4. After that, in section 1.5, the main contributions of the thesis is explained. Finally, the outline of the thesis is presented in section 1.6.

## 1.1 Basic Concepts and Definitions

### 1.1.1 Biometric user authentication

A biometric system uses one or more physiological characteristics (e.g., fingerprint, palmprint, face, iris, retina, face, voice) or behavioral traits of a user (e.g., signature, touchscreen, gait, key strokes) or hybrid schemes that take advantage of the both systems to authenticate the legitimate user (mostly known as verification).

User authentication problem is divided into two main domains. User identification and user verification. In user identification, when a probe (unknown biometric template) is presented to the system, the system should determine the identity of the user by comparing it with the

gallery set (known templates)[1] of legitimate users in the system to find the best match and determine if the input biometric template belongs to any of enrolled identities in the system. However, in user verification, a user claims an identity by presenting a probe and the system should validate the user's claim by comparing the probe with the galley set of the enrolled legitimate user in the system.

In general, a biometric user authentication system has two main phases. Fig. 1.1 shows a common authentication system.

- **Enrolment**: user enrolment of authentication systems can be studied in two categories. *Model-based* enrolment takes advantage of machine learning models to classify the legitimate user's templates from impostors' templates. Collected user's biometric templates are used for model training and model selection purposes that will be used in verification phase. In this approach, a user verification problem turns into a binary classification problem in machine learning era. *Template-based* enrolment uses similarity metrics such as distance metric to compare a presented probe to the system with the legitimate user's profile (gallery set). The user is a legitimate user if the similarity of her presented biometric template to the stored profile passes a pre-defined threshold in the system.

- **Verification**: in this phase, the system validates the user's claiming identity. Similar to the enrolment phase, this process starts with the feature extraction method. Depending on what approach is used in the enrolment phase, user verification can be *probability-based* approach or *similarity-based* approach. Probability-based approach is used when model-based method is employed in the enrolment phase where a squashing function produces a probability value that shows how likely the presented probe belongs to the legitimate user.

$$D = \begin{cases} P, & \text{if } \sigma(x) > T_C. \\ N, & \text{otherwise.} \end{cases} \tag{1.1}$$

  where, $x$ is the input, $\sigma$ is the squashing function, and $T_C$ is the classification threshold. $P$ means that the user is verified as a legitimate user (positive class) while $N$ means that the user is an impostor (negative class). On the other hand, the similarity-based approach is used when template-based approach is used in the enrolment phase. When a probe is presented to the system, the authentication system compares its similarity to the legitimate user's gallery. Distance metrics such as euclidean distance or cosine similarity are used to measure the similarity between templates. The probe belongs to

[1] also known as reference set or enrollment set

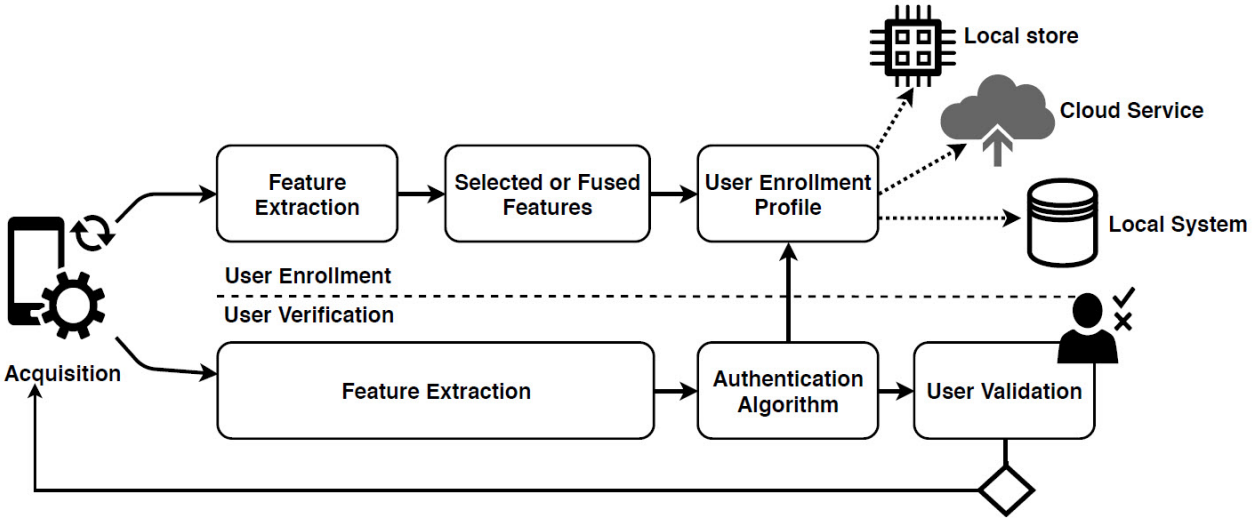



Figure 1.1 User biometric authentication systems have two phases. In user enrollment, biometric templates or the model's internals are stored for verification phase. In user verification, a verification algorithm based on the enrollment approach is employed to decide whether the presented template belongs to the legitimate user or not. The user can access the system upon the positive verification decision [15].

the legitimate user if its similarity metric is above a pre-defined threshold.

$$D = \begin{cases} P, & \text{if } d(x, y) > T_D. \\ N, & \text{otherwise.} \end{cases} \tag{1.2}$$

where, $d(x, y)$ is the similarity function, $x$ is the input template, $y$ is the legitimate template, and $T_D$ is the distance threshold.

A biometric system should satisfy the following requirements [16]:

- Universality: does everyone have the biometric characteristic?
- Distinctiveness: are two persons distinguishable using the biometric characteristic?
- Permanence: does not the biometric characteristic change over a long time?
- Performance: does the recognition system achieve fast acceptable accuracy by using low system resources?
- Acceptability: how willing are people to use the biometric characteristic?
- Circumvention: can the authentication system be easily fooled?

From the security point of view, we can divide authentication systems into *single point entry* authentication and *active authentication* (AA)[2]. Single point entry authentication systems validate the user's identity only once at the login time. It is likely that an impostor takes over the phone after a successful authentication. However, active authentication systems continuously and unobtrusively monitor the mobile user after the login while she is working with the device until she logs out of the system. This new emerging authentication scheme minimizes session hijacking attacks on smartphones. These systems mostly use behavioral biometrics such as gait, touchscreen biometric, motion sensors or facial attributes (using the smartphone's front-facing camera). Fig. 1.2 shows a face-based active authentication system.

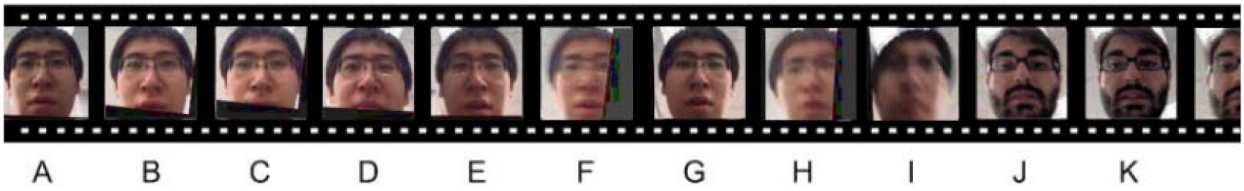



Figure 1.2 An active authentication system can detect intrusion on the phone even after login. Frames A-I show the legitimate user working with the phone. At frame J, an impostor takes the phone's control. Since the AA system periodically captures user's face images, it can detect the attack on the phone [17].

### Evaluation metrics

In order to evaluate authentication systems, several evaluation metrics are widely used. In the following we discuss these metrics.

- False Negative Rate (FNR): shows the percentage of legitimate users falsely identified as impostors.
- False Positive Rate (FPR): shows the percentage of impostors falsely identified as legitimate users.
- True Positive Rate (TPR), Recall (REC): is the probability of correctly identifying legitimate users, and defined as

$$REC = \frac{TP}{TP + FN} \tag{1.3}$$

[2]also known as continuous authentication (CA)

where, TP and FN are the number of true positive and false negative samples for a given threshold, respectively.

- Precision (PRE): identifies the frequency with which the authentication system was correct when predicting the legitimate user. That is

$$PRE = \frac{TP}{TP + FP} \tag{1.4}$$

where, FP is the number of false positive samples for a given threshold.

- Receiver operating characteristic (ROC): is a graph that shows the performance of a binary classification at all classification thresholds. The curve is plotted by showing the TPR of the system for different FPR values at a specific classification threshold.

- Area Under Curve (AUC): is the possibility of ranking a randomly chosen positive instance higher than a randomly chosen negative one by a classifier.

- Equal Error Rate (EER): represents a point where false positive rate (FPR) equals false negative rate (FNR). This point is obtained by intersecting the ROC curve with a diagonal of the unit square. In general, the lower the equal error rate, the higher the accuracy of a biometric system. Figure 1.3 illustrates the relation between ROC, AUC, and EER.

### 1.1.2 SIM/eSIM Overview

Smart Cards (SCs), devices in card format including an embedded integrated circuit (IC), are used to store information in a secure and yet flexible way. Smart card architecture consist of the following components:

- CPU: all smart card processors support 8-, and 16-bit computations. Moreover, some smart cards also support 32-bit calculations.

- I/O Interface.

- Crypto-processor: used to perform cryptographic operations or hash calculation such as RSA, ECC, SHA, faster on chip (not available on all smart cards).

- Memory

  - ROM: stores the OS

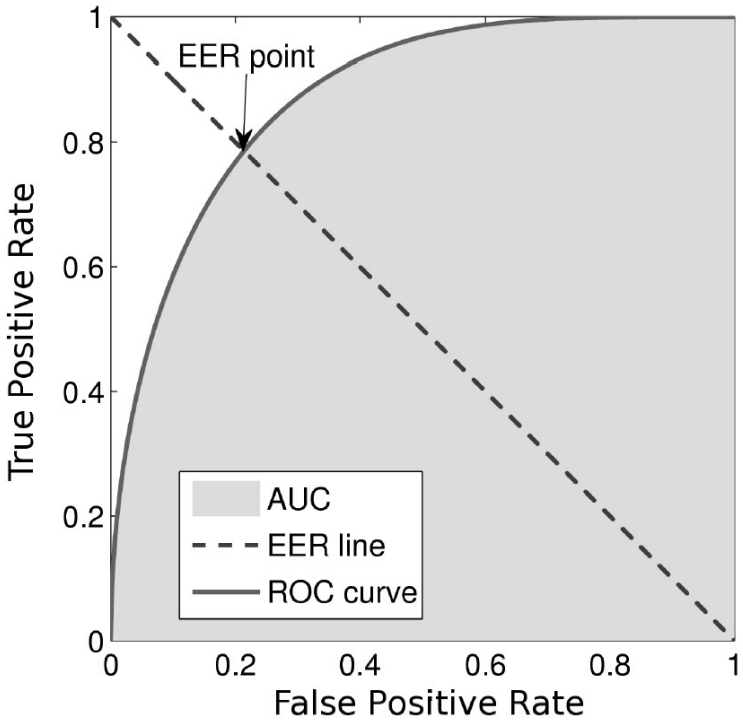



Figure 1.3 ROC, AUC, and EER of an authentication system [18]

– EEPROM: used to store system and application files, and to load applications.
– RAM: used during execution of applications to store computation values.

Smart cards demonstrate high security features that make them a good candidate to store sensitive information for authentication, cryptography, communication, personalization as payment cards, ePassport, public transport fare cards, SIM cards. Some of these security features are: 1) communication channel is a restricted path to the smart card that is controlled by the card OS. 2) smart cards support most of encryption algorithms even in hardware to offer higher security solutions. 3) smart cards are kept small and less complex to prevent system misfunctionality, or easy to find security flaws in the system [19]. 4) smart cards have a secure OS which is tamper resistant and robust against side channel attacks and fault injections [6], [20].

Smart cards communicate with the outside world using a packet mechanism called Application Protocol Data Units (APDUs). The communication model is a command-response model. The card receives a command APDU, performs the processing requested by the command, and returns a response APDU [21]. An APDU commnad is 255 bytes and consists of 5-byte header and 250 bytes of data. Fig. 1.4 shows the format of an APDU packet.

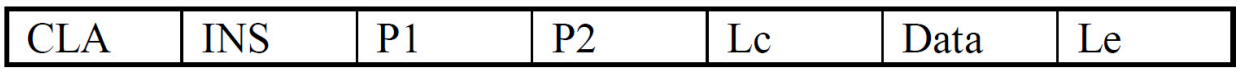



Figure 1.4 APDU packet format.

In Fig. 1.4:

- CLA indicates one of *ISO 7816-4* classes implemented on smart cards' OS.
- INS indicates the instruction code defined in an applet to run a specific function.
- P1 and P2 indicate additional parameters.
- Lc indicates the length of the data.
- Data is the input data for a specific function in an applet.
- Le indicates the length of the expected response.

However, not all smart card applets require data from the outside to operate. Therefore, the Data field is optional in the packet.

### Java card

Smart card applications or *applet*s are mostly developed by a simplified version on Java technology called *Java card* technology. Software components on the smart cards with Java Card technology are [21]:

- Card OS.
- Native services: performs the I/O, cryptographic, and memory allocation services of the card.
- Java Card Virtual Machine (JCVM): provides bytecode execution.
- Framework: the set of classes which implement the API.
- Application Programming Interface (API): used by applets to accesses the JCRE and native services.
- Java Card runtime environment (JCRE): provides the class libraries and other resources that a specific applet needs to run.

- Industry extensions: add-on classes that extend the applets installed on the card.
- Applets: application written in Java to run on smart cards.

### SIM card

Smart cards' security features made them a good candidate for secure storage of network subscribers' credentials and secure authentication of subscribers to the network. Smart cards in the form of Subscriber Identity Module (SIM) cards are used to securely store a *profile* on user devices. A profile contains the operator's and the subscriber's credentials that are used to authenticate the subscriber to the operator's network. It includes authentication keys `KI`, `KC`, `Opc Key`, International Mobile Subscriber Identity (IMSI) that uniquely identifies the subscriber in the mobile network, Integrated Circuit Card Identifier (ICCID) that is a serial number to identify the SIM card, among other useful information. Moreover, profiles can contain applets to offer value added services (VAS) to subscribers in order to take advantage of security features and available resources on the SIM cards.

### eSIM

Physical limitations of a SIM card such as its size, fragility, or physical security hinder it to be a successful player in Internet of Thing (IoT) era, despite its successful presence in telecommunication era for 25 years. With the explosion of IoT devices, the limitations of SIM cards are more evident. Embedded SIM (eSIM) is an evolution of SIM card designed to address the limitations of traditional SIMs. They incorporate new functionality that is needed to enable the world of IoT devices. eSIMs unlike SIMs are permanently soldered into the device and are a container of several mobile network operator (MNO) profiles. eSIMs are managed remotely with a platform called remote SIM provisioning (RSP) that enables storage and management of multiple MNO profiles.

Using traditional SIM cards, the user sets up a contract with an MNO and receives a SIM card that should be inserted into the device to connect to the MNO's network. If the user decides to change her operator, she should set up a new contract with a new MNO and physically swap the SIMs (new one with the old one) to connect to the new MNO's network. However, using eSIMs, there is no physical SIM available to the end user. When the user sets up a contract with an MNO, instead of a SIM card, she receives an instruction on how to connect to the MNO's RSP and download the required profile, probably in the form of Quick Response (QR) code. If the user wants to change her operator, she scans the new QR code to download the profile from the operator's RSP into the eSIM. The user can easily

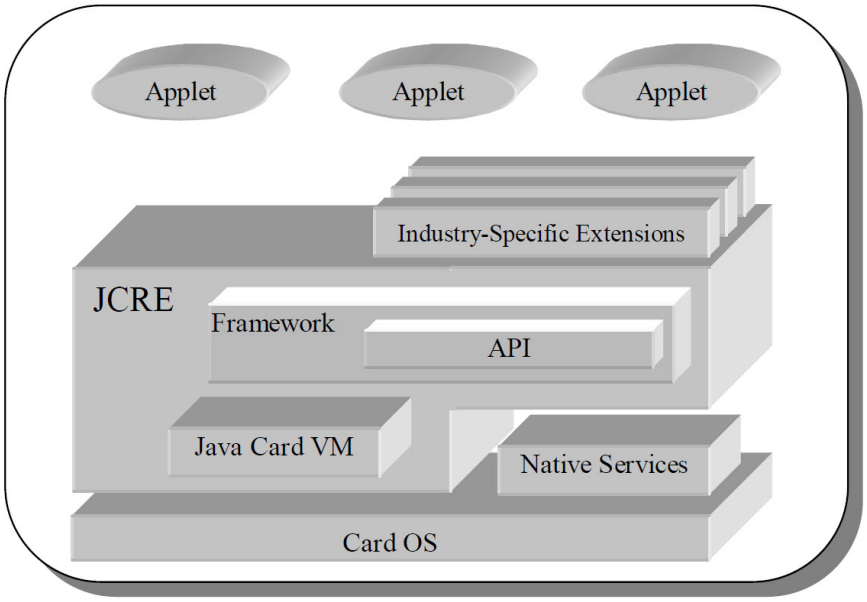



Figure 1.5 Smart card software components [21].

switch between the profiles available on the eSIM whenever she is not satisfied with the offered service of the current network [22]. Fig.1.6 illustrates the MNO profile installation and profile selection.

## 1.2 Privacy-preserving (secure) biometric authentication

Although biometric authentication systems provide a great usability and accuracy for users, they are the target of many security attacks. Despite their ease of use and accuracy compared to traditional authentication methods such as password, biometric templates are not kept secret and they are not easily revocable if compromised. Therefore, the consequences of being stolen are more severe such as identity theft, illegal access to personal records on electronic services such as e-health, e-ID, or unauthorized access to information on personal devices. Therefore, it is crucial to design a privacy-preserving biometric authentication system that is able to minimize the risk of aforementioned attacks. A privacy-preserving authentication system should meet the following requirements:

- resistance to possible attacks: a privacy-preserving authentication system should be robust against attacks on privacy, especially impersonation or spoofing attacks.
- Revocability: which indicates that in case of privacy compromise of biometric templates, the user should be able to revocate the previously enrolled templates and enroll

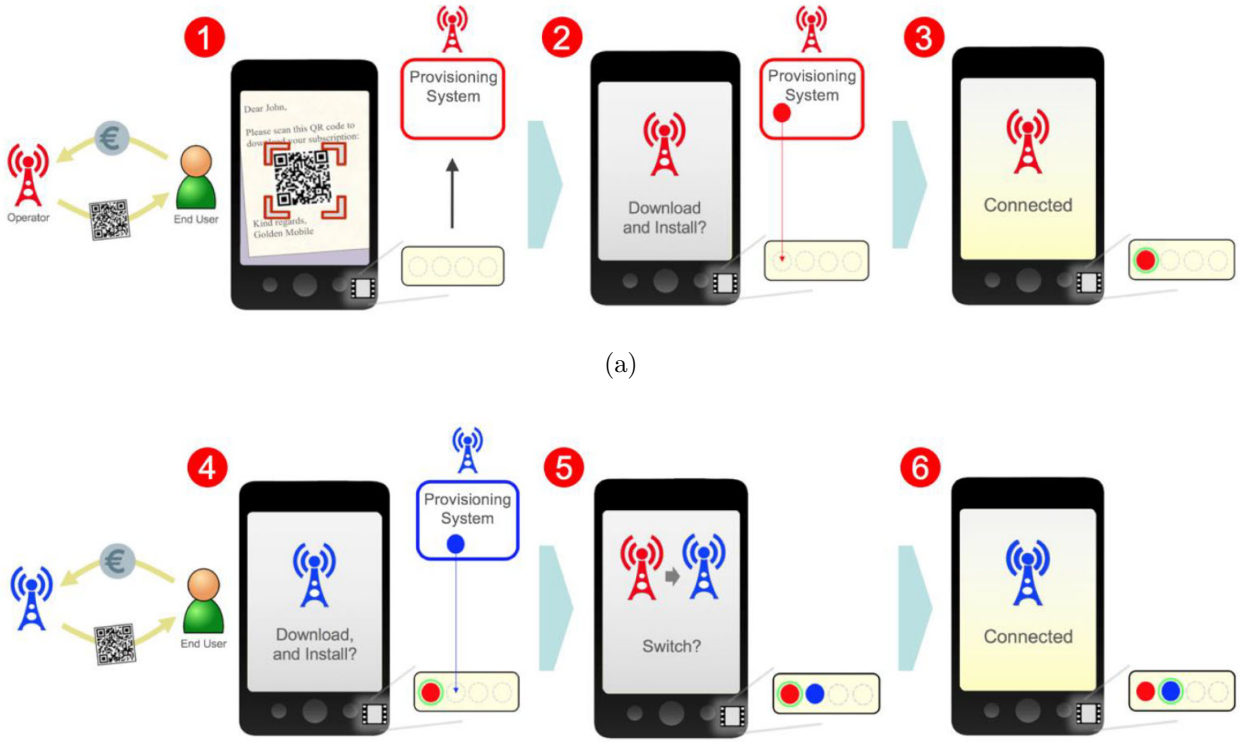



Figure 1.6 Remote SIM provisioning (RSP) operations. (a) shows the MNO profile download and installation. (b) shows profile selection on consumer device [22].

a new gallery set on the system.

- Noninvertibility: if biometric templates are transformed in order to protect the privacy of biometric information, this transformation should not be invertable. Otherwise attackers can easily retrieve the original biometric templates from the transformed ones.
- Unlinkability: access to the original bioemtric templates should be disconnected from the outside world. This way, we minimize the risk of network attacks to the biometric database.

The term privacy is about the safeguarding of the user's identity and the term security is about the safeguarding the user's data. In biometric-based authentication systems, since the data to protect is in fact the identity of the user, privacy-preserving authentication is also referred to as secure authentication. Therefore, the two terms are used interchangeably throughout the thesis.

### 1.3 Open problems in biometric authentication

Biometric authentication has become popular among mobile users due to its ease of use compared to the traditional authentication methods. However, there are several open issues in boimetric-based authentication that need to be addressed to achieve a higher level of acceptance among users.

- Performance: biometric authentication is supposed to grant access to valuable information on the system. Therefore, its performance accuracy and low resource consumption are crucial for a successful implementation. Finding a new biometric authentication solution that verifies legitimate users with high accuracy with zero false positive rate (FPR) and zero false negative rate (FNR) is a challenging task. Except well-known fingerprint biometric, other physiological biometrics have not shown that percentage of accuracy. For behavioral bimetrics this situation is even worse.

- Security: increasing the security of a biometric system requires collecting more information from the user (i.e., extract more features), increasing the frequency of data acquisition (i.e., active authentication), modality fusion (e.g., use fingerprint with face), or increasing the difficulty of verification phase (i.e., using deep recognition methods), all add more overhead on the mobile system that increases resource consumption. Therefore, design of an authentication system with high security, high performance and high efficiency is a challenging task and needs more attention from the academic community.

- Privacy protection: biometric templates can reveal the identity of their owners. Moreover, considering their enduring connection with owner, their privacy becomes very important. Leakage of biometric information may have many serious consequences for users, where it can be used to impersonate the legitimate user to access different services online or on the phone, or unauthorized access to personal or financial information.

- Usability enhancement: biometric systems need to demonstrate a high level of usability to be widely accepted by end users. The enrollment phase should be straightforward and fast, data acquisition should not be complicated, feature extraction needs to extract most distinctive features in an acceptable time, design of the user interface is important, and the verification phase should be in real-time, more importantly the system must have a high accuracy with almost zero false positive rate.

- Biometric for IOT: with the increasing use of wearable devices such as smartwatches, design of a lightweight and accurate biometric authentication system for constrained devices is a big challenge. Most of these devices do not support large data processing

tasks leading to a degrade in performance accuracy. Therefore, a trade-off between security, efficiency, and performance need to be considered for biometric authentication on resource constrained devices.

## 1.4 Research Objectives

The more the people trust on biometric authentication systems, especially on their personal devices such as smart phones, the more they reveal their identities to third parties. Considering the long-lasting characteristics of biometrics such as fingerprint, face, or behavioral traits, the increasing use of biometrcis will increase the risk of identity thefts. Therefore, secure and privacy-preserving authentication systems are required for biometric-based authentication on mobile devices.

### 1.4.1 Main objective

The main objective of this thesis is to design a lightweight and secure biometric-based system for real-time authentication on smartphones.

### 1.4.2 Specific objectives

The main objective of the thesis is divided into several specific objectives as follows.

1. Design an architecture for secure active authentication.
2. Propose a quantization scheme to reduce resource consumption on the device.
3. Design an optimization architecture to make real-time authentication decisions.
4. Present a generic model for an accurate and privacy-preserving authentication system.
5. Implement the proposed model on real smartphones.
6. Evaluate the performance of the system on different publicly available datasets.

## 1.5 Research Contributions

In this thesis, we design a novel privacy-preserving biometric-based authentication system for mobile users. The proposed system unlike other research efforts, takes advantage of hardware security of smartphones and demonstrates its potential for a secure authentication

on smartphones with a fast and higher accuracy performance and low resource consumption. This thesis makes the following contributions:

- **Mobile match-on-card system**: while most of the studies on privacy-preserving authentication concentrate on software-based solutions on smart phones, we consider SIM/eSIM cards on smart phones as a secure element (SE) to store biometrci templates and verify users securely isolated from the untrustworthy mobile environment. This way, we can benefit from the built-in and well-known security characteristics of SIM cards used for more than 25 years in telecommunication industry. Mobile match-on-card is a term coined within this thesis to indicate the match-on-card technique on smartphones.

- **Deep Neural Network (DNN) on smart cards**: DNN has shown higher accuracy in classification problems, and has been used in recent biometric authentication systems. Therefore, a novel quantization scheme is proposed to make a DNN-based verification system implementable on smart cards. Using the proposed scheme, we can implement a DNN model on off-the-shelf SIM cards without relying on any specific hardware design. To the best of our knowledge this work is the first work that implements a DNN model on smart cards.

- **Quantization scheme**: to this date, there is no smart card that support floating point operation. All operations are done in integer domain. Therefore, the models' internal (in DNN or in other models) and the inputs to the model (i.e., inputs to smart cards) should be converted to integers prior to sending to the smart card. A quantization scheme is proposed to convert floating point datatype to integer without losing a lot of accuracy in the system. Two quantization schemes are introduced. The first one maps data in a real valued range to a specific integer range readable for the smart card. The second scheme, goes further by moving to log domain where it can significantly reduce the execution time in the DNN model while showing low quantization error as well.

- **On-card optimization**: an authentication system is expected to work in real-time. However, smart cards have limited resources and even a simple computation on smart phones is an expensive computation on smart cards. To mitigate the effect of smart card resource constraints on the authentication system, an optimization architecture is proposed to speed-up the verification process on the card. The effect of the proposed technique is more noticeable in DNN models where many multiply-accumulate calculations (MACs) are needed to obtain the verification decision. The proposed solution

helps us to gain 44.3× speed-up over the original architecture on a DNN model showing the feasibility of on-card real-time authentication using a DNN inference.

- **A generic model for smart card-specific feature extraction**: deep feature extraction is widely used to extract features from biometric templates. They use multiple layers to extract more discriminative features of the samples in a domain. These models need a giant dataset for their high accuracy which may not be available on a specific problem domain. Moreover, the extracted feature vectors can be large and not suitable for smart cards. Using transfer learning concept, we fine-tune a model on our target dataset, and produce features that are suitable for our proposed privacy-preserving authentication system. In addition, a quantization layer is added on top of the feature extraction network to make extracted features readable for smart cards.

- **Implementation on real devices**: an MMOC face-based active authentication system is developed for Android devices. The system has three components: 1) a component on a cloud server for training purpose, where it trains a user specific model for each legitimate user by using a reference dataset and *one-vs-all* protocol. 2) an Android application for data acquisition and pre-processing that captures user face images using front-facing camera every sampling interval, pre-process and extract features for enrollment or verification. 3) a SIM card applet for secure storage of biometric templates during the enrollment phase or secure matching during the user verification phase. In enrollment, after a pre-defined number of samples are stored in the SIM, the samples are sent to server for training. In the verification, a newly captured face image is compared to the threshold to grant access to the system or not. To the best of our knowledge this is the first implementation of an MMOC system on real devices.

## 1.6 Thesis Outline

In this chapter, we introduced basic concepts needed to pursue the research done in this thesis. We talked about biometric authentication systems, and open issues in biometric systems. Moreover, we discussed about SIM/eSIM card, their functionalities and security features as a solution for privacy-preserving biometric authentication. Then, we defined the research objectives of the thesis, followed by the research contributions. The remainder of the thesis is organized as follows. Chapter 2 discusses the related works in one-shot or active biometric authentications especially recent works on face-based and touchscreen-based active authentication systems. Moreover, we also study the research works that considered match-on-card (MOC) for secure and privacy-preserving biometric authentication. Chapter

3 describes the relation between the submitted/published articles obtained from this research and the objectives of the thesis set in section 1.4.

Chapter 4 presents the full text of an article entitled *«Mobile Match on Card Active Authentication Using Touchscreen Biometric»*, accepted for publication in the *IEEE Transactions on Consumer Electronics* journal. This article proposes a secure touchscreen active authentication system based on MMOC technique. As the user interacts with the touchscreen, the system in the background extract features and store them privately on the SIM/eSIM card. A DNN classifier is trained on a cloud server and the inference phase, simplified with a quantization scheme, is migrated to the card for active authentication. The active authentication engine, every sampling interval, captures new touchscreen sample, extract features, pre-process and send them to the SIM for verification. In order to increase the authentication accuracy *multi-stroke* authentication is applied. Moreover, a speed-up technique is employed to decrease the verification time on the card.

Chapter 5 presents the full text of an article entitled *«Lightweight and Secure Face-based Active Authentication for Mobile Users»*, submitted to the *IEEE Transactions on Mobile Computing* journal. In this article, a face-based active authentication system is proposed that uses SIM cards as a secure element. A highly accurate deep feature extraction method that is implementable on smart cards is employed to extract the most distinctive features from each face image. Two architectures are proposed for user authentication. First architecture that is model-based authentication uses cloud resources for training and model selection, while the second architecture that is a template-based authentication only relies on card resources for enrolment and verification; therefore, showing higher privacy compared to the first architecture. A numerical method is employed to implement euclidean distance computation on the card for comparison with the classification threshold. An on-card active authentication model is proposed that controls the security and efficiency of the system. An analysis is done to prevent any malfunctioning of the system when the card is isolated from the outside world (i.e., unlinkability). Moreover, the article reports platform evaluation results obtained by an implementation on real devices.

Chapter 6 presents the full text of an article entitled *«A Generic Model for Privacy-Preserving Authentication on Smartphones»*, accepted for publication in Proceedings of the *15th Annual IEEE International Systems Conference*. This article presents a generic model for an MMOC-based privacy-preserving authentication system. It uses transfer learning technique. The feature extraction network is modified to meet the requirements for a successful on-card implementation, then the model is fine-tuned on a target dataset. The customized feature extraction network is used afterwards for user verification on the card. Since the

classification sub-network of transfer learning has several layers with multiple nodes, its implementation is cumbersome on smart cards. A novel log quantization scheme and an on-card optimization architecture are proposed to decrease the execution time of on-card classification sub-networks. The model is evaluated on face biometrics; however, since transfer learning is also employed on other biometric systems, the proposed model has the potential to be used for other recognition systems.

Chapter 7 presents a general discussion about the proposed authentication system and the obtained results. Finally, chapter 8 concludes the thesis by highlighting the contributions of the research, limitations, and showing the path for future research works.

# CHAPTER 2 LITERATURE REVIEW

User authentication is a central part of many security services. After user verification phase, the user can utilize resources or services offered to legitimate users [39]. In this chapter, we review biometric-base authentication (single point entry or continuous) systems. More precisely, we study two main types of biometric systems, that is physiological or behavioral biometrics, and privacy-preserving authentication systems that is the main objective of this research as well.

Biometric authentication systems use physiological or behavioral characteristics of a user to validate her identity. Upon authentication decision, the access is granted to the user or her access is denied. On the other hand, biometric systems can operate in a single point authentication where the user is verified only once at the beginning of a session, or in an active authentication in which the user is continuously and transparently is authenticated during the session by the system in the background.

## 2.1 Physiological biometrics

Physiological biometrics are physical characteristics of a user that are not supposed to change in a long-time. Some of well-known physiological biometrics are discussed in the following:

### 2.1.1 Fingerprint recognition

With no doubts, fingerprint recognition is the most successful and acceptable biometric authentication for smartphones. Matching accuracy of fingerprint is shown to be very high; as a consequence many manufacturers equip their devices with fingerprint authentication. Fingerprint systems mostly use *minutia* or *patterns* techniques to extract most distinctive features of a fingerprint.

Liu *et al.* [23] proposed a real-time embedded finger-vein recognition system for smartphone authentication. The system is implemented on a DSP platform. A novel finger-vein recognition algorithm is proposed that helped the system to take only about 0.8 seconds to verify one finger-vein template with an Equal Error Rate (EER) of 0.07% on a database of 100 subjects.

Derawi *et al.* [24] introduced the first effort towards employing cell phone cameras capturing fingerprint images as biometric templates. They evaluated the proposed approach using about 1300 fingerprint images from each embedded capturing device. Fingerprints were collected

by a Nokia N95 phone and a HTC Desire. The results of the proposed method showed a performance with an EER of 4.5% by applying a commercial extractor.

### 2.1.2 Palmprint recognition

Users' palms contain pattern of ridges and valleys that are different among people. Palms have larger area compared to fingerprint; therefore, they tend to output more accurate authentication results. On the other hand, the size of palms hinders them to be acceptable on mobile phones. Moreover, since palmprint scanners scan larger areas, they are more expensive. There are several research works on palmprint recognition on smartphones. For example, Han *et al.* [25] proposed a real-time palmprint authentication system for mobile phones. They used sum-difference ordinal filter to extract disticitive features of palmprint using only +/- operations on image intensities. They claimed that the proposed algorithm verifies a user palmprint in about 200 ms while having $\frac{1}{10}$ of other methods' complexity.

Use of Palmprint for biometric authentication on smartphones is studied in [26]. They showed that using the main camera in smartphones, good quality pictures of palmprint are captured that can be used as an alternative biometric authentication on smartphones. Applying image processing techniques, they extracted the palmprint Region Of Interest (ROI) in the captured image. They used Competitive Code (CompCode) algorithms for feature extraction of palmprint ROIs. A Collaborative Representation Classifier with Regularized Least Squares (CRC-RLS) was used for model training and verification of legitimate user. This classifier has a high performance and needs light computational resources that is a good selection for constrained resource devices. It showed an average EER of 7.4% and Recognition Rate (RR) of 86.06%.

### 2.1.3 Iris recognition

The iris is elastic region of the eye bounded by the pupil and the sclera on either side. The iris has unique patterns that are different from a person to person. Moreover, the iris recognition systems show a promising accuracy and efficiency that make the iris recognition systems good candidates for large-scale identification systems [27]. The trabecular meshwork, which is an area of tissue in the eye located around the base of the cornea through which aqueous humor flows out of the eye, is used a the main characteristic for iris recognition systems [28].

Park *et al.* [29] proposed an iris authentication system for mobile users based on corneal Specular Reflections (SRs). The proposed method demonstrated promising results on face images with and without eye glasses. Experimental results with 400 face images captured

from 100 persons with a mobile phone camera showed that the rate of correct iris detection was 99.5% (for images without glasses) and 98.9% (for images with glasses or contact lenses). The accuracy of iris authentication was 0.05% of the EER using the proposed method.

### 2.1.4 Face recognition

This authentication system uses face images or face videos captured by the front-facing camera of the phone to validate the identity of the user. With advances in development of inexpensive and high quality camera sensors, and deep learning-based recognition approaches, face recognition accuracy has boosted drastically, and is used by several smart phone manufacturers.

Face features are extracted in different ways:

- Geometrical features: size, position, and shape of facial attribute are used for face recognition.
- Three dimensional features: special cameras are used to capture three dimensional face images.
- Skin texture: uses details in the skin of each user and extract features directly from pixel representation without considering geometric features.
- deep features: deep learning techniques are employed to extract the most distinctive face features by training a complicated network on giant datasets. To the date, this technique is the most accurate feature extraction method for face recognition.

Deep learning method for feature extraction begun when AlexNet won the ImageNet competition by a large margin in 2012 [30].Deep learning methods (i.e., convolutional neural networks), use several layers of processing units for feature extraction and transformation. They learn multiple levels of representations that correspond to different levels of abstraction. The levels form a hierarchy of concepts that are robust to the face pose, lighting, and expression changes. In deep feature extraction networks, the first layers learn the features designed for years or even decades, such as Gabor filter, and the later layers learn more detailed features. Finally, the combination of these higher level abstraction represents facial identity with unprecedented stability [31].

Well-known Convolutional Neural Network (CNN) architectures such as ResNet, VGGNet, SENet are used as the baseline models for face recognition [32–34]. After a CNN network is trained with a giant dataset and a right loss function, the feature extraction sub-network

is used to extract features on a traget dataset, and test images to obtain a deep feature representation. Once the deep features are extracted, similarity metric methods are used to calculate the similarity between two features using euclidean distance or cosine similarity metric. The Nearest Neighbor (NN) or thresholding techniques are used for verification task. Moreover, some methods use Deep Neural Network (DNN), metric learning or Sparse-Representation-based Classifier (SRC) to perform an efficient and accurate face matching.

The most successful architecture in deep face recognition starts with AlexNet in 2012. AlexNet has an augmentation layer with 5 convolutional layers followed by relu and dropout layers and 3 fully connected layers [30]. In 2015, VGGNet used an architecture with very small convolutional filters ($3 \times 3$). They also showed that increasing the network depth to 16-19 weight layers significantly improves the system performance [32]. In 2016, ResNet reformulated the layers as learning residual functions with reference to the layer inputs. They showed that these residual networks are easier to optimize, and can achieve higher accuracy from considerably increased depth. They evaluated ResNet on ImageNet dataset with network depth of 152 layers and achieved error rate of 3.57% with lower complexity [33].

These mainstream architectures are used in face recognition era afterwards. In 2014, DeepFace employed explicit three-dimensional face modeling, and derived a face representation from a nine-layer DNN with several locally connected layers without weight sharing. The proposed method reached an accuracy of 97.35% on the Labeled Faces in the Wild (LFW) dataset [35]. In 2015, FaceNet used a large dataset to map the face image $x$ into a compact feature space $\mathbb{R}^d$. FaceNet architecture consists of a deep CNN followed by $L_2$ normalization that produces face embedding $f(x) \in \mathbb{R}^d$ which embeds a face image into a $d$-dimensional Euclidean space. Triplet loss is introduced during training that minimizes the Euclidean distance between the *anchor* ($x_i^a$) and the images of same identity ($x_i^p$), and maximizes the distance between the *anchor* and the images of different identity ($x_i^n$), that is $L = \sum_i^N \left[\|f(x_i^a) - f(x_i^p)\|_2^2 - \|f(x_i^a) - f(x_i^n)\|_2^2 + \alpha\right]_+$ where, $N$ is the number of samples in the training set and $\alpha$ is a threshold between positive and negative samples. FaceNet achieves accuracy of 99.63% on LFW dataset and 95.12% on YouTube Faces DB [36]. In the same year, VGGFace collected about 2.6 M images of around 2.6 K people from the Internet. Then, VGGNet architecture was trained on the obtained dataset using the same loss function as FaceNet. It achieved an accuracy of 98.95% on the dataset [37]. In the late 2017, VGGFace2 introduce a large dataset containing 3.31 M face images from 9.1 K people. The dataset has large variations in pose, age, illumination, ethnicity and profession. For evaluation, ResNet-50 is trained on VGGFace2, and MS-Celeb-1M [38], and showed that training on VGGFace2 leads to improved recognition performance over pose and age [39]. Fig. 2.1 shows a general framework of deep face recognition systems.

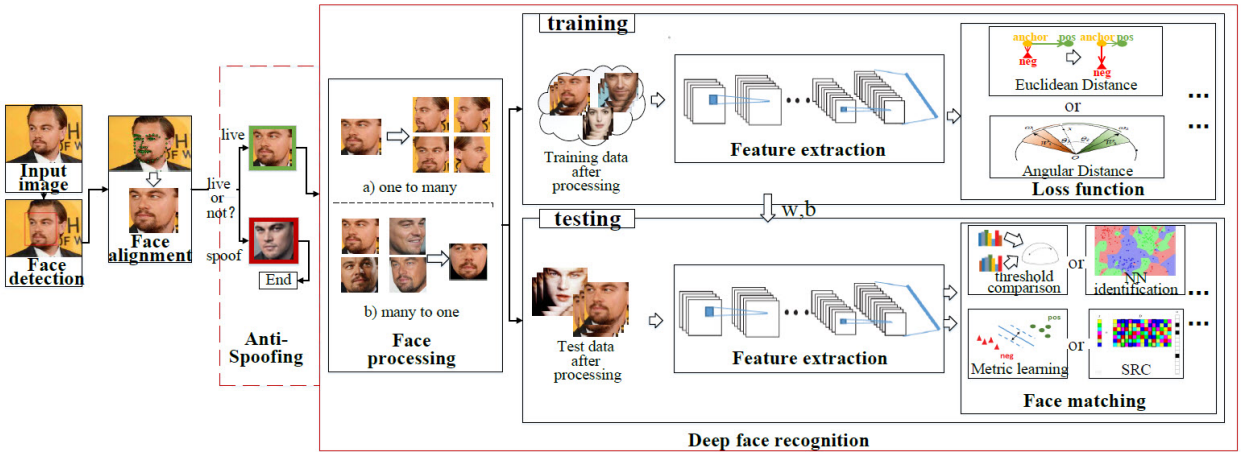



Figure 2.1 After face detection and alignment, an anti spoofing method is employed to verify liveliness of the face. In the training phase, data augmentation methods are used to generate more face image samples for CNN-based feature extraction, while in testing phase, methods are used to generate one canonical face image from different non-frontal face images. Feature extraction network outputs the most distinctive features using a defined loss function with the augmented dataset. The trained model is used in testing (verification) as well, and a matching function classifies a test image face as a legitimate or impostors [31].

With high quality front-facing cameras on recent smartphones, face recognition also shows the potential to be use as a continuous authentication system where the front-facing camera in static or dynamic time intervals captures face images of the user working with the phone. The authentication engine in the background validates the identity of the user transparently without interrupting the user's interactions. The system only interrupts the user if it detects an illegitimate usage of the device. The frequency of user verification, the feature extraction, and data preprocessing should be considered carefully in order to keep low energy consumption of the face-based active authentication on smartphones.

Fathy *et al.* [40] studied face-based active authentication on smartphones using a face videos dataset captured by the device's front camera in different illumination conditions. They evaluated recognition rate of different still-image-based methods and image-set-based methods on different scenarios. Their study showed a significant drop in recognition rate when user is enrolled in one session and verification is done on the other sessions.

Mahbub *et al.* [41] evaluated face-based active authentication on UMADD-02 dataset which contains data collected form three sensors on the phone: front camera, touch sensors, and location sensors. They compared different feature extraction methods for face and different similarity metrics for face verification. They reported the best EER of 18.44% using DCNN and Cosine similarity metric.

Samangouei *et al.* [42] presented a method for face-based active authentication using facial attributes. For each facial attribute, a set of features is extracted and a classifier is trained on it. For user verification, extracted features of facial attributes are compared with those of the legitimate user's enrolled attributes. They evaluated their proposed method with Local Binary Pattern (LBP)-based method using two publicly available datasets, and reported that the fusion of LBP and attributes results better performance in terms of EER. They also evaluated platform performance by implementing a simplified version of their method on a real device.

McCool *et al.* [43] introduced a valuable publicly available audio-visual dataset for mobile phones. For face verification, they calculated an average histogram of LBP features over all frames with detected faces in the enrollment videos and the test video and a similarity score is obtained. This score is compared to a threshold obtained from the validation set to decide whether this sample comes from the identity it claims or not. They showed that fusion of face and speaker improves the performance more than 25%.

Crouse *et al.* [44] proposed a face-based continuous authentication system. They fused captured face images with sensory data such as gyroscope, accelerometer, and magnetometer data to correct face orientation. They extracted face features using Biologically Inspired Model (BIM) and trained a Support Vector Machine (SVM) classifier on a person-specific training dataset. Moreover, they introduced login score $s_{login}$ which is updated periodically as a new face image is captured every $t_{sample}$, and the user is logged out of the system if $s_{login}$ is below a predefined threshold $T_{login}$.

Perera *et al.* [45] presented a method for face-based multiple user AA systems based on Extremal Openset Rejection (EOR). For evaluation, they compared the proposed EOR-based verification with different methods on three publicly- available datasets. They concluded that EOR method results superior performance where number of users on the device varies.

## 2.2 Behavioral biometrics

Behavioral biometrics are good candidates for continuous authentication on mobile devices in which the user is transparently and unobtrusively is authenticated in the background. This technique increases the security of mobile users against possible attacks after the first login and during an active session.

Most of active (continuous) authentication systems take advantage of the various sensors available on smartphones to monitor the behavior of the user such as motion patterns [46–48], touchscreen gestures, gait, keystroke dynamics, or profiling on the phone. These sensors

include motion sensors (such as accelerometer, gravity, magnetometer, and gyroscope), position sensors (such as compass or GPS), or environmental sensors (such as temperature, light, barometer, and proximity). We discuss active authentication using behavioral biometrics in the following categories:

### 2.2.1 Motion-based authentication

Motion sensors on smartphones give us enough information to study the behavior of mobile users. Sensors such as accelerometer and gyroscope are available on modern smartphones. Accelerometers in samrtphones are used to detect the orientation of the phone. The gyroscope adds an additional dimension to the information supplied by the accelerometer by tracking rotation or twist.

Ehatisham-ul-Haq *et al.* [46] presented a novel active authentication for smartphones which recognizes mobile users based on their physical activity patterns using accelerometer, gyroscope, and magnetometer sensors. They also analyzed the system performance when the user places the smartphone at different body locations. Among different machine learning algorithms, SVM achieved the best results for user recognition with average accuracy of 97.95%.

Amini *et al.* [49] presented *DeepAuth* as a framework for re-authenticating mobile users. In the proposed system, they used time and frequency domain features extracted from motion sensors and a Long Short-Term Memory (LSTM) model with negative sampling to build the framework. It is able to verify a user with 96.70% accuracy in 20-second authentication window for a dataset of 47 subjects.

Regarding motion-based authentication, some researchers also studies eye movements for continuous authentication. Zhang *et al.* [50] proposed to use eye movement to continuously authenticate the current wearer of a Virtual Reality (VR) headset. They used an implicit visual stimuli to evoke eye movements of the wearer without distracting her from the normal activities. They evaluated their appraoch on a dataset of 30 subjects. They evaluated the time stability of the proposed method by collecting eye movement data on two different days that are two weeks apart. Their method achieved an EER of 6.9% if data for testing was collected from the same day and 9.7% if data was collected from two weeks apart.

In another study, Zhang *et al.* [57] investigated a biometric method based on the saccadic eye movement. Saccadic eye movement and gaze fixation between them can be applied to identify users. They claimed that large saccades can reveal better differences between individuals. For classifying saccades, they applied different classifiers namely, Linear Discriminant Analysis

(LDA), SVM with linear, polynomial of the third degree and RBF, and Multilayer perceptron (MLP) networks trained with eight nodes of one hidden layer. Their results showed TPR of 80-90% with polynomial SVM showing the best performance.

### 2.2.2 Gait-based authentication

Gait-based authentication verifies smartphone users by the manner of walking. In most of gait recognition systems, the data is collected using floor sensors or wearable sensors. In the first approach, especial sensors are located on the floor which learn the walking behavior of the user by stepping on them. The latter approach, the smartphone sensors are used to collect information about the way the user walks. Especially accelerometers sensor available on mobile phones have been used for acceleration based gait authentication [51]. Gait-based authentication consists of four phases. 1) data collection phase where the device is placed at a right body location to collect walking information. 2) data preprocessing in which methods are used to remove the noise added during data collection phase due to environmental and gravitational, the user's wearable materials, or other factors. 3) walk detection phase where cycle-based methods or machine learning algorithms are used to detect walk. Machine learning techniques have shown more accuracy in walk detection where a model is developed to detect walk. 4) analysis phase, time intervals or frequencies are studied. For acceleration based gait authentication, Dynamic Time Warping (DTW) as distance metric between two time series is usually used. For frequency analysis, wavelet transformations have been used with non-cycle-based acceleration gait data [52].

Yeh *et al.* [53] proposed plantar biometrics for continuous authentication in IoT-based environments with wearables. Raw bio-data are extracted by plantar pressure sensors embedded into slippers. A Raspberry PI II platform collects the data and forwards them to the back-end authentication server in a transparent manner. Machine learning techniques such as Naïve Bayes (NB) and SVM with Gaussian radial basis function (SVM-GRBF) are used for classification and individual verification.

Muaaz *et al.* [54] studied an approach to deal with recognition errors that come from the continuously changing sensor orientation under a realistic scenario by using the magnitude data of tri-axes accelerometer and wavelet based noise elimination modules. They obtained the best performance result of EER 7.05% in a same-day scenario where reference and probe cycles are a same session walk, with the phone placed in trouser pocket.

Gafurov *et al.* [55] attached sensors to the person's body to collect motion activities of the user. They analyzed acceleration signals from the foot, hip, pocket and arm. They used several methods (features) on acceleration signals, the best EER obtained for foot-, pocket-,

arm- and hip- based user authentication were 5%, 7.3%, 10% and 13%, respectively.

### 2.2.3 Touchscreen gesture-based authentication

Touchscreen gesture or *stroke* is a finger action on the screen from the time it touches the screen until it is lifted. These actions can be swipes, taps [56], flicks [57, 58] and slides [59]. Using built-in smartphone sensors many discriminative features such as acceleration, velocity, pressure, touch area, angle of the stroke, *x-y* coordinates can be extracted from each stroke. These features show touchscreen behavior of the user and can be used to verify the legitimate user with a high accuracy.

Shahzad *et al.* [1] proposed a behavior based authentication scheme for touch screen devices. They used gestures and signature actions for user authentication. Then, they extracted several types of features like velocity magnitude, device acceleration, stroke time, inter-stroke time from the samples. The features of consistent values are selected, and a Support Vector Distribution Estimation (SVDE) model is trained to verify legitimate users from impostors.

*ShakeIn* [60] is a user authentication scheme by shaking the smartphone. The fact behind the shakeIn is that every person has consistent and distinguishing physiological and behavioral characteristics when shaking. They extracted unique biometric features from the embedded sensors in the devices and used them to train a SVM classifier. Later, the SVM classifier is used to verify legitimate user form imposters.

Fierrez *et al.* [61] proposed a swipe gesture scheme for continuous user authentication. They captured the stroke features during a normal activity of a user. Using two different classifiers, i.e., SVM and Gaussian Mixture Model (GMM), similarity score is computed comparing it to the selected templates. Using different datasets, they showed several interesting findings about swipe gestures such as horizontal swipes are faster than vertical independently of the device orientation, landscape orientation is more stable, and horizontal gestures are more discriminative than vertical ones. Moreover, they also studied intra-session (a user is enrolled and authenticated within the same day), inter-session (a user is enrolled in one day and authenticated in another day two weeks apart), and mixed-session scenarios (the data from both sessions are combined). The results showed that best EER is achieved in the intra-session scenario.

Frank *et al.* [62] proposed a continuous authentication scheme relying on the way users interact with the touchscreen device. They considered 30 different behavioral features such as mid-stroke area covered, mid-stroke pressure, average velocity, average direction, stroke duration, phone orientation among others to be extracted while the user is interacting with

the device. During the enrollment phase, the classifier learns to create a profile of legitimate users. In the continuous authentication phase, the classifier, based on the user's interaction with the touchscreen, decides whether he is a legitimate user or an imposter. Two different classifiers k-nearest-neighbors (kNN) and SVM are used in their work. The proposed system achieves EER of 0% in intra-session scenario, 2%-3% in inter-session scenario, and below 4% in a mixed-session scenario. Moreover, they also introduced multi-stroke authentication where several strokes are used to verify the user. It helps to increase the accuracy of the system. For a single stroke the EER is about 13% while at a level of 11 to 12 strokes, the EER drops to a range between 2% and 3% and stays there up to 20 strokes.

Antal *et al.* [63] showed that users' gestures on touch screen devices can be used to classify users' identity, gender, and experience level in using the device. They extracted several user related features from each stroke, and by using three different machine learning algorithms, i.e., SVM, k-NN, random forests, they depicted that identity, gender and experience level prediction reach 95% accuracy from 10 or more strokes.

Serwadda *et al.* [64] evaluated ten classification algorithms using the same dataset they gathered from 190 subjects, each subject participating in two sessions that were at least one day apart. Among the algorithms, i.e., SVM, Naïve Bayes, Random Forest, Neural Network, Logistic Regression, J48 tree, etc., Logistic Regression and J48 tree had the lowest and highest mean EER respectively. Moreover, they used "failure to enroll" concept to improve the system performance by preventing users whose mean EERs are above a certain threshold.

Shen *et al.* in [65] investigated the reliability and applicability of user's behavior on touchscreen to be used for continuous authentication on mobile devices. They analyzed users' touchscreen interactions on a collected dataset of 71 participants with 134900 touchscreen operations. Different scenarios, different touch operations, various application tasks, and various touch operation length, along with different classification techniques ( nearest neighbor, neural network, support vector machine) are studied in their work. Their results revealed that touchscreen gestures are discriminative, reliable and stable among mobile users, and achieved EER of 1.72% with touch operation length of 11. Accuracy improves with long touch operations and small time intervals between operations. Moreover, the accuracy is higher in a specific task rather than in free tasks.

Miguel-Hurtado *et al.* [66] proposed the use of touchscreen gesture data for the prediction of soft-biometrics, particularly the user's sex. They evaluated the performance of different classification techniques (naïve bayes, logistic regression, support vector machine and decision tree). Their results showed that using a fusion of swipe direction-based decision with two different swipe directions, the user's sex can be predicted using her touchscreen interaction

pattern by accuracy of 78%.

### 2.2.4 Behavior-based profiling authentication

This authentication technique aims to identify users based on the way they interact with the services on their smartphones. During the authentication, the activities of the user such as application usage, dialing numbers, activities on social networks, location, screen time, and other user specific behaviors are collected and compared to the legitimate user's profile through a machine learning method.

Anjomshoa *et al.* [67] used social behavioral biometric for continuous authentication of users. They extracted features from users' interactions with five online social network services and built-in sensors of a mobile device such as location of users, number and duration of interactions with the social networks. Using cloud-based machine learning techniques a verification model is trained, which is used for verifying the authenticity of the users.

Li *et al.* [68] proposed a novel behaviour-based profiling technique by using mobile application usage pattern of the current user and detect abnormal mobile activities. MIT Reality dataset was used for performance evaluation. They divided the dataset to three subsets; two intra-application datasets compiled with telephone and message data; and an inter-application dataset containing the rest of the mobile applications. A user's profile was generated using static and dynamic profiles. Using three sets of applications i.e., telephone call, text message, and application-level applications, they reached the EER of 5.4%, 2.2% and 13.5%, respectively.

Acien *et al.* [69] proposed an authentication system based on an ensemble of biometrics and behavior-based profiling signals. They considered the fusion of seven different biometric data: Touch dynamics (touch gestures and keystroking), accelerometer, gyroscope, WiFi, GPS location and app usage. The biometric data were collected during the user interaction with the smartphone. Moreover, they also studied both one-time authentication and continuous authentication. Their results showed that the multimodal system increases the system performance with accuracy ranging from 82.2% to 97.1% depending on the authentication scenario.

## 2.3 Multimodal authentication

Multimodal authentication incorporate two or more biometric modalities in order to built an authentication system with higher accuracy and security. Considering various sensors in modern mobile devices and the sufficient processing resources on them, this technique tends

to be a promising authentication method on smartphones.

Galdi *et al.* [70] combined recognition of the human face and smartphone fingerprint, with the image processing capabilities of new smartphones, both the distinctive characteristics of the face and of the device that captures the face image can be extracted from a single photo or video frame and used for a double check of user identity. They proposed a method to identify smartphones based on camera fingerprint. They claimed that sensors of different cameras could be distinguished by analyzing the Sensor Pattern Noise (SPN). This method combined with face recognition techniques using Histogram of Oriented Gradient (HOG) were used for user authentication on a smartphone. Authentication based on smartphone identity resulted in an EER of 0.3; however, combined with face recognition authentication their proposed scheme reached an EER of 0.06.

Sultana *et al.* [71] proposed a multimodal biometric authentication scheme based on the fusion of Social Behavior Biometric (SBB) with face and ear biometrics. Their study showed that human social interaction with Online Social Networks (OSN) has a distinguishing pattern that can be used to authenticate users. They used Twitter as the source for their social data. Social behavior patterns extracted from user interaction with Twitter were fused with face and ear biometrics using post-matching parallel score fusion of face, ear, and SBB information. They showed that their scheme obtained a more reliable result compared to a unimodal method using one of the three fused biometrics and increased the correct match accuracy to around 99%.

Monwar *et al.* [72] introduced a multimodal biometric system based on the fusion of different individual biometric matchers for face, ear, and signature. They presented an effective fusion scheme that combines information presented by multiple domain experts based on the rank-level fusion integration method. They showed that the fusion of individual modalities could improve the overall performance of the biometric system. They used different rank-level fusion methods namely, Highest rank, Borda count, and logistic regression. Based on the ROC curve they showed that logistic regression method performs better than other methods, also face recognition achieved higher accuracy compared to ear and signature.

Fox *et al.* [73] introduced a multi expert biometric system by combining the information from face, mouth, and speech. Since the contribution from each source of information to the final decision must be taken into account based on the reliability of the expert, they chose a score level fusion based on the weighted sum rule. Using a subset of XM2VTS, their experiment showed that the proposed multi expert system outperforms the individual experts by at least 19.9%.

Paul *et al.* [74] proposed a multimodal biometric system using Social Network Analysis (SNA)

for improving the overall performance of three individual face, ear, and signature matchers at different fusion levels. At the first level, SNA was used to improve the confidence score of each classifier. At the second level, it was fused with the outcomes of the other individual classifiers to obtain the final decision. Their result revealed that using the combination of face, ear, signature, and SNA improves the performance of the biometric system by reaching Genuine Acceptance Rate (GAR) of 100% with 5% FAR while other methods achieve it with 12% FAR.

Zhu *et al.* [75] introduced an authentication system called *RiskCog* that authenticates the user transparently and in real-time manner using inertial sensors. RiskCog does not require any user input and no requirement on the device placement and the user's motion state. RiskCog collects data from the acceleration sensor, gyroscope sensor and gravity sensor, and uses SVM as a classifier. They evaluated their proposed system on a large dataset of 1,513 users. Their result showed an average system accuracy of 93.8% and 95.6% for steady and moving users, respectively.

## 2.4 Privacy-preserving biometric authentication

The proliferation of biometric usage brings serious security and privacy concerns in user authentication. Storing enrollment templates on the device or on cloud servers has a serious security flaw that an attacker can steal the enrollment templates of the legitimate user which allows him to have unauthorized access to services, sensitive personal or banking information; moreover, it may cause identity theft due the long-lasting connection of biometric data to the user's identity. In traditional authentication systems, password are stored in cryptographic hashes; however, since biometric information are noisy in nature, this method is not suitable for privacy-preserving in biometric-based authentication. In this section, we present an overview of privacy-preserving biometric-based authentication systems, also referred to as *secure biometrics*. We can study privacy-preserving biometric systems in four categories: *fuzzy commitment*, *secure multiparty computation*, *cancelable biometrics*, and *match-on-card*.

### 2.4.1 Fuzzy commitment

In a conventional *bit commitment* scheme, one player seals bit $b$ for the second player by encrypting bit $b$ as $y$ which is called *blob*. It is infeasible for the second player to find the real value ($b$) for $y$, and only the first player can open the encrypted value in one-way. Moreover, the first player cannot change value $b$ while it is in hands of the second player. A bit commitment scheme is said to be *concealing* if it is infeasible for the second player

to guess $b$. Also, it is said to be *binding* if it is infeasible for the first player to decommit the $y$ with the incorrect bit, for example *1-b*. In bit commitment, the first player can open blob $y$ with a witness value $x$. However, in *fuzzy commitment* blob $y$ can be opened with any witness $x'$ that is close to $x$, but not necessarily identical to $x$. Putting this definition in biometric authentication context, in enrollment phase, the user presents biometric template $x$ to the authentication system $S$. $S$ selects a code $c$ for the user and computes the commitment $y = F(c, x)$ and stores it on the device or on a cloud storage. In authentication phase, the user presents $x'$ to $S$. The system checks whether $x'$ can perform a successful decommitment [11].

Several methods are proposed in literature to bind code word $c$ to biometric templates. One method is to use Error Correcting Code (ECC) where the error correction codes is used to mitigate the inherently noisy nature of biometric traits. Error correction would decode small perturbation of a template into the template itself, solving the problem of noisy data. In this way, the systems can get error-free biometric templates that will not affect the matching biometric process [76], another method is to use Quantization Index Modulation (QIM) [77]. However, it has been shown that fuzzy commitment techniques leak private information [78].

### 2.4.2 Secure computation

This approach tries to compute the distance between enrollment templates and the probe in an encrypted domain [79]. Public key Homomorphic Cryptosystems (HC) are mostly employed in this architecture. In homomorphic cryptosystems, operations in plaintext domain can be carried out in ciphertext domain. In enrollment phase, feature vectors are encrypted element wise using an additively homomorphic cryptosystem, and the result is stored locally or on a remote database. In authentication phase, the user presents a feature vector probe, and the system encrypts it with the public key. Then, the system computes the euclidean distance between the enrollment template and the probe template in ciphertext domain [80]. The user is verified as a legitimate user if the distance is below the classification threshold. Some studies also considered nearest neighbor computation [81, 82]. Fig. 2.2 illustrates a secure biometric system using the homomorphic cryptosystem method. Another method is Garbled Circuits (GC) that is a cryptographic technique that enables two entities to compute a function and only reveal the output of the function. This technique show a high potential for privacy-preserving authentication. To the best of our knowledge, garbled circuits is the most promising cryptographic tool to prevent template recovery attacks in biometric systems [83].

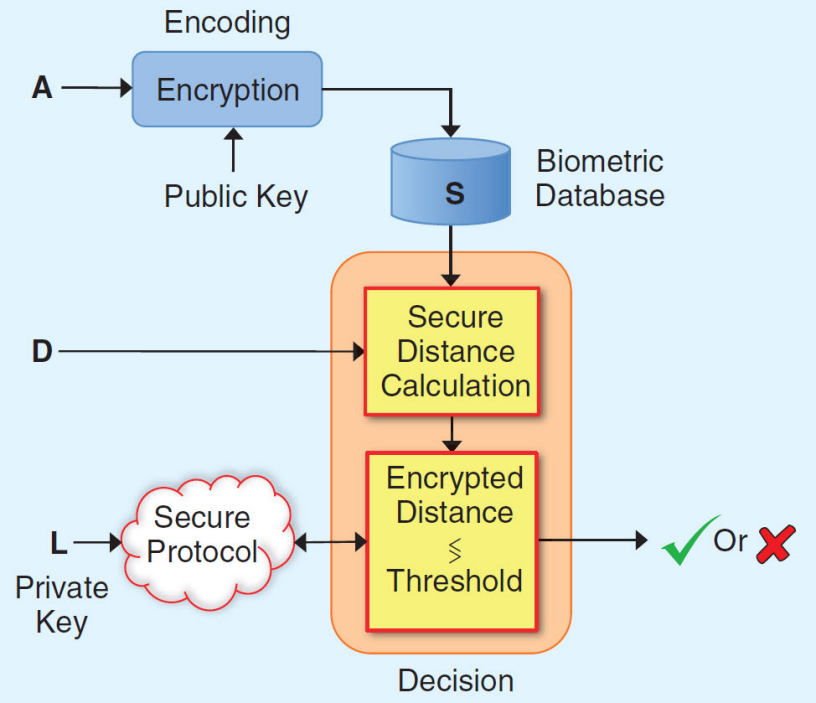
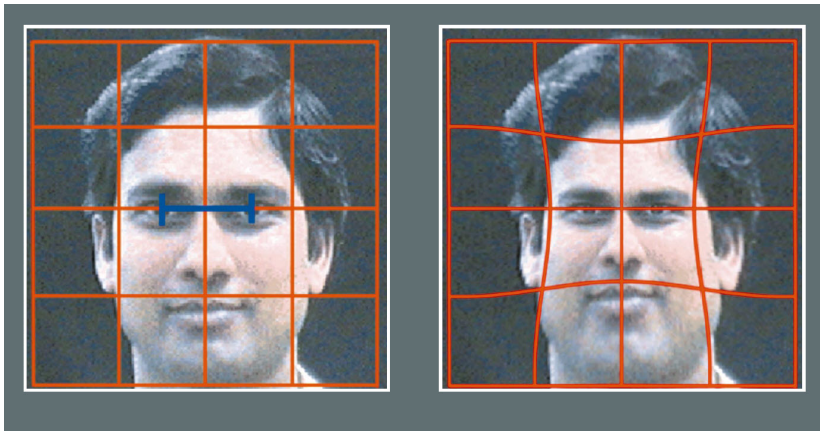



Figure 2.2 Homomorphic cryptosystems. Enrollment templates are encrypted with the public-key and stored encrypted in a database. For authentication, the distance in the encrypted domain is computed and compared with the threshold. The legitimate user has the right private key to decrypt and process the decision result [79].

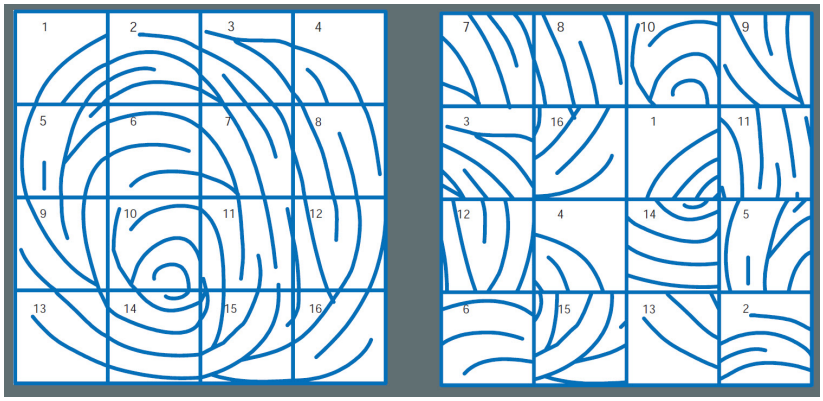
### 2.4.3 Cancelable biometrics

In order to protect the privacy of biometric templates, in cancelable biometric systems, biometric signals are distorted before being stored on local or remote databases [84]. The distorting function should be noninvertible and revocable which means that the transformed enrollment templates can be revoked if any suspicious activity is detected. Distortion transforms can be used in the signal domain or in the feature domain. Distortion methods in the signal domain include gird morphing and block permutation. Fig. 2.3 shows different distortion transforms in the signal domain. An example of a distortion transform in the feature domain is a consequence of random perturbations of feature points. This can be done within the same physical space as the original, or in a higher dimension space. The second case provides more brute force strength [85]. Some research works also considered biohashing to produce cancelable biometrics [86]. BioHashing is a new method for secure biometrics that combines biometric features and a Tokenized Random Number (TRN) to hide the original biometric features. It has four steps: 1) a set of pseudo-random vectors are generated based on the seed. 2) the Gram–Schmidt process is applied to the pseudo-random vectors and

obtain TRN, a set of orthonormal vectors. 3) the dot product of the feature vector and each orthonormal vector is computed. 4) a biocode $b$ is obtained by comparison with a defined threshold in which $b_i$ is 0 if the dot product of step 3 is less than or equal to the threshold, and is 1 otherwise, and computing quantized random projections of biometric feature vectors [87]. There some works that have shown the vulnerability of canclable biometrics where a one-way transform is analyzed and inverted successfully [88].



(a)




(b)

Figure 2.3 Distortion transform in cancelable biometrics [85]. (a) image morphing technique. (b) block permutation technique.

### 2.4.4 Match-on-card authentication

Smart cards have many security characteristics that make them suitable for security services that need protection of sensitive and confidential information such as financial information (bank cards), mobile network information (SIM cards), identity information (e-ID,

e-Passport), among many other applications. These applications started with e-purse systems that were used in mid 1990s. Introduction of smart cards in telecommunication systems in 1990s for secure storage of GSM profiles was a turning point in the proliferation of smart card usage. Among other applications, citizen cards, driver's licences, and patient cards are widespread. Contact-less smart cards are also successful in several applications such as contact-less transit cards, monetary transactions that are used widely around the world. Smart cards have all elements of a real computer in small scales, they embedded microprocessor, ROM, EEPROM, and RAM. However, this technology is growing fast and today's smart cards have acceptable resources that make them good candidates for secure systems even with high resource consumption demands. For example, most recent smart cards have a 32-bit CPU, with 2MB of EEPROM, around 40 KB of RAM, and a crypto co-processor for a fast hardware computation of cryptographic operations such as RSA, ECC, AES, or other algorithms. The communication bandwidth with the outside world is 115 kbit/s in ISO 7816 contact mode, and up to 424 kbit/s in ISO 14443-B contact-less (RF) mode [89].

Smart cards not only can be used for secure storage of biometric templates, but also using their processing resources, these devices can be used for secure user verification. Smart card-based authentication systems can be studied in two approaches:

- Template-On-Card (TOC): in which only biometric templates are stored on the smart card, and the verification phase is done outside the smart card, on the device or on a remote server.
- Match-On-Card (MOC): in this architecture, not only biometric templates are stored securely on the smart card, but also the user verification phase is performed on the smart card as well. No biometric template is transferred to the outside world.

Most of the research efforts in this field, consider smart cards for secure biometric storage, i.e., TOC approach. However, there are some few research works that have considered MOC as a secure biometric authentication solution. We discuss them in the following sections.

**Fingerprint-based match-on-card**

In 2000, Noore [90] proposed a secure and reilable on-card biometric authentication by using dual on-chip biometric fingerprint sensors that are integrated with the smart card architecture. The fingerprint sensor on the front of the card is based on DC capacitive sensor technology and the biometric fingerprint sensor on the back of the card is based on AC electric field sensor technology. When a user holds the proposed smart card two fingerprints

are simultaneously captured. Both sensors are capable of producing images with resolutions greater than 500 dpi. Since the system uses two fingerprints and process them simultaneously, the reliability of identifying the "true" owner of the card during its use is enhanced.

Seto [91] proposed two-step method for authetication over the Internet in which the smart card itself is authenticated based on a Public Key Infrastructure (PKI), and the user is authenticated using the fingerprint template stored in the smart card where the user is verified using match-on-card technique. For on-card user authentication, the probe fingerprint is legitimate if the number of the chip images (small images around the feature points) on the fingerprint is above a threshold. To embed the chip matching function into a smart card, the memory limitation on the card does not allow the storing of the entire fingerprint image in the memory at one time. The on-card matching scheme uses a partial image of the captured fingerprint that is transmitted to the smart card then matched to the chip image in turn.

Kumar and Ganesh [92] proposed the idea of integrating smart card and Gabor Filter method for fingerprint with matching on card technique. Their system uses Gabor filters to capture details in a fingerprint and present it as a compact fixed length FingerCode that will be stored in the smart card. In authentication phase, when user inserts the card into the system, it will ask for the user ID and a fingerprint probe. The matching-on-card module will be successful only if the Euclidean distance between the FingerCode of the probe and stored FingerCode of the smart card is equal to zero. They claimed that the system achieves greater accuracy, faster verification, and is highly fool proof.

Bistarelli *et al.* [93] proposed a novel matching algorithm for fingerprint on smart cards. The main feature of the algorithm is its asymmetric behaviour to the execution time, between correct positive and negative matches. The matching algorithm calculates how similar is the neighborhood of a minutiae in the probe template is similar to the neighborhood of each minutiae in the galley set. Two templates are matched if the similarity score is above a threshold. Their matching algorithm achieved the best EER of 0.48% on a Hybrid Database. For a maximum minutiae occupation of 40 bytes, the on-card matching time is reported about 1-8 seconds for nearly all of the matches.

Nedjah *et al.* [94] presented an efficient user authentication using finger print minutiae. Their method is based on Skin Elasticity Tolerant Algorithm (SETA) for fingerprint comparison. SETA requires large space for storing the results of translations and rotations. Therefore, in order to implement their proposed method on smart cards, they subdivided the search space into small sub-spaces.

**Face-based match-on-card**

The first work on face-based authentication on smart cards was proposed in Li's PhD thesis [95] in 2000. He proposed Client Specific Linear Discriminant Analysis (CS-LDA) method for an on-card face recognition system, and extensively studied the effect of system's limited capacity on the performance of the system.

Kittler *et al.* [96] proposed a one dimensional client specific fisher face representation for personal identity verification. A distance to the client template, and a distance to the mean of impostors is computed to find legitimate users from impostors. Using the public dataset XM2VTS, they showed the simplicity of the training phase, and the possibility of enrollment insulation which is suitable for smart cards. Moreover, the result showed that the speed of verification is more than 2× faster than that achieved by conventional PCA and LDA methods.

Czyz *et al.* [97] evaluated a face verification system based on Fisherfaces (Principal Component Analysis and Linear Discriminant Analysis) for implementation on computationally constrained devices such as smart cards. Their results on XM2VTS dataset showed that reducing the image quality to 256 pixel gray images and model size to 20 real numbers (i.e., dimension of LDA subspace) do not degrade the performance of the verification system, and reaches EER of 3.83%. For 64 pixel images, they suggested to use the model with another biometric modality.

Lee and Byun [98] used SVM as a classifier for face authentication on memory-constrained devices such as smart cards. They used Genetic algorithm (GA) to select most discriminating features to achieve best classification performance and to have small size to be suitable for implementation on smart cards. Finally, they tested their proposed MOC-based face authentication system on different datasets and showed GA+SVM outperforms SVM in terms of FPR/FNR.

Bourlai et al. [99] employed the CS-LDA technique for face verification on smart cards. After image normalization and feature extraction, the distance between the probe image and the user's template is computed on the smart card and compared to the threshold to verify whether the probe belongs to the legitimate user or an impostor. They also optimized the verification system by reducing the spatial and grey-scale resolution of images.

Findling *et al.* [19] presented a MOC authentication approach using face biometrics. They trained and simplified an offline model on a sever and the simplified model is migrated to the smart card where authentication is done using the stored model on the card. On one hand, performance is improved slightly using 32-bit cards compared to 16-bit cards. On the other

hand, 32-bit cards perform more expensive computations that increases execution time on the card. They reported an EER of 2.4%-5.4% for face biometrics.

### Other biometrics for match-on-card

Except fingerprint and face bioemtrics that are mainly studied in the literature, other biometrics are rarely considered for on-card implementation. In the following, we review some of the key research works that have considered other biometrics with matching on card. Among them only one of them studied match-on-card using behavioral biometrics.

Nedjah *et al.* [100] proposed an implementation of iris texture verification on smart cards using MOC technique. False Positive Rate (FPR) and False Negative Rate (FNR) are improved using circular translations of the matched iris codes. They used segmentation, normalization and binary code formation for iris feature extraction and, Hamming distance for comparison of iris codes. Iris code is an array of $8 \times 256$ bits; however, Java card does not support multi-dimensional array. Therefore, they stored the iris codes row by row. In order to reduce the execution time on the card, they proposed acceptance threshold that is a predefined proximity value from which the comparisons should be considered correct. Using acceptance threshold FPR is about 0.42% while the FNR is about 15.95% and the average execution time for authentic comparisons is 1210 ms, while the average time for false comparisons is 2430 ms.

Sabri *et al.* [101] proposed a multimodal biometric verification framework consisting of two MOC fingerprints and one match-on-host (MOH) face system. They used a dynamic sequential score fusion algorithm to improve the accuracy of their authentication system. If the quality of the current biometric trait is not good enough, the next classifier is used while using the score of the first.

Choi *et al.* [102] proposed a speaker verification system based on SVM. They reduced the number of features and implemented it on a 32-bit smart card. The proposed method $100\times$ reduces the required memory and can be executed in real-time. SVM shows higher accuracy compared to DTW and Hidden Markov model (HMM) with TER of 1.76% ($TER = FPR + FNR$). Moreover, they proposed a hardware design for the algorithm on FPGA platforms.

Nedjah *et al.* [100] implemented a palmprint verification on smart cards. They extracted binary code (palm-code) from each palm-print image, and used Hamming distance for comparison. Upward, downward, leftward and rightward translations of the matched palm-codes are proposed to improve systems' FPR and FNR. They extracted an area of interest (ROI) and used Gabor 2D filter for the extraction of the main palm-print features, then generated

accurate binary codes. In order to reduce the execution time on smart cards, they introduced an acceptance threshold to decrease the number of comparison on the card. They achieved EER of 0% for comparisons with 2-bit translation, with the average execution time of 3725 ms and no acceptance threshold while it reduces to 1217 ms when the acceptance threshold is imposed.

Findling *et al.* [103] presented an MMOC approach that uses offline training to obtain simplified authentication models to enable their usage on smart cards. The obtained model was used on the card without requiring retraining when enrolling individual users. They used the proposed approach to acceleration based mobile gait recognition using 16 bit smart cards, and evaluate authentication performance and computation time on the smart card using a publicly available dataset. Their results showed that the approach is feasible with an equal error rate of 11.4% and an execution time below 2 seconds on the smart card, including data transmissions and computations.

# CHAPTER 3 APPROACH OF THE ENTIRE RESEARCH PROJECT

This thesis aims to design a biometric-based authentication system for smartphones to enhance the security and the privacy of users' biometrics while reducing the resource consumption. In order to achieve this main objective, three specific objectives have been defined in section 1.4. These objectives were broken down into three main phases, each of which was the subject of a scientific publication article that will be discussed in the next following chapters. In this chapter, we present the approach of the entire research project by highlighting the link between the defined objectives and the published or submitted articles obtained from this research.

## 3.1 Phase 1: A secure architecture for active authentication

Recently, active authentication systems have attracted a considerable attention from the academic community and the industry owing to their ability to protect users' confidential information more by continuously monitoring the current user in the system. However, these systems suffer from: 1) well-known issues in biometric-based authentication such as security and privacy of biometric templates, and 2) their specific issue of resource consumption. Therefore, an active authentication system that addresses these issues is missing in the literature, and was the motivation for the first published article entitled «Mobile Match on Card Active Authentication Using Touchscreen Biometric» presented in Chapter 4.

### 3.1.1 Cloud-assisted secure active authentication

While many research works in the literature, consider software solutions such as homomorphic encryption or cancelable biometrics to increase the security and privacy of the system, we approach the security and privacy issue in biometric systems from hardware security perspective. The core of our architecture is mobile mach-on-card (MMOC) technique. This in a new term used in this research work, where it uses SIM/eSIM card as a secure element (SE) to protect users' biometric information and verify users in a secure environment.

With extensive research efforts placed in neural networks and the high performance accuracy gained from these algorithms, DNN-based solutions are replacing other machine learning algorithms in many different applications. One of these applications that requires a higher accuracy is the authentication system. These approaches need a large dataset for training that requires high computation resources. However, these resources are not available on

SIM/eSIM cards, nor on smartphones. Therefore, a cloud-assisted architecture is proposed that takes advantage of cloud resources for model selection and training purposes.

### 3.1.2 DNN quantization scheme

Although the model training and the model selection are done using the cloud processing power; however, the verification phase (network inference) should be implemented on SIM/eSIM cards for high security and privacy. Therefore, a quantization scheme is proposed to reduce the model size and make it implementable on resource constrained devices such as SIM cards. This scheme is applied to model inputs and model internals as well. This quantization scheme consists of off-card and on-card quantizations. The off-card quantization tries to map floating-point real values of the model to a specific integer range supported by the SIM card, while the on-card quantization tries to return these values to their closet real valued integers. This scheme helps us to reduce the quantization error and keep the system's accuracy close to the original model. Moreover, we replace the ReLU activation function with a clipped-ReLU function in order to prevent overflows caused by dot products in the neural network layers.

### 3.1.3 Performance evaluation

Touchscreen biometrics demonstrate the behavior of the user while interacting with the touchscreen through the finger movements on the screen such as swipes, taps, flicks, and slides. This biometric authentication scheme shows high distinctiveness and high accuracy reaching to 99% [15]. Moreover, most of the user's interactions with the mobile device is through the touch screen. Therefore, we concentrate our solution on an active authentication using touchscreen biometrics. We employed two publicly available datasets and evaluated our proposed system in terms of its robustness against *spoofing attacks*, its performance from different angles (i.e., EER, AUC, PRE, REC, F1), and its execution time on SIM cards. We show that our system is robust against the main attack vector in biometric systems that is spoofing attack. Moreover, it is also robust against adversarial attacks on machine learning algorithms. In terms of accuracy, our system reaches EER of 2.6% (real-valued EER is 1.1%) with real-time response of 650 ms. The results show the feasibility of implementing a secure active authentication system using touchscreen on SIM cards even when a DNN model is employed.

## 3.2 Phase 2: Fully secure active authentication for biometrics with big templates

In section 3.1, we considered touchscreen biometric for active user authentication. Touchscreen biometrcis are intrinsically lightweight bometric data with few number of features (i.e., around 30 features) and consequently the verification can be done in real-time on the card. However, this situation is not always true for other biometrics in which they extract more distinctive features and produce large feature vectors that makes their implementations a tedious task on SIM cards. One of these biometrics that is popular in the academic community and the industry is face biometric that generally extracts many features from a face image for a higher accuracy. With the high quality front-facing cameras available on almost all modern smartphones, this biometric can also be used as an active authentication system. However, this biometric due to its big template size brings more challenges for MMOC implementation. Thanks to *transfer learning*, the face biometric can be modified for MMOC implementation with low quantization error and even higher security. In the article entitled «Lightweight and Secure Face-based Active Authentication for Mobile Users», discussed in chapter 5, a highly secure and accurate face-based active authentication system is presented, and implemented on real smartphones as well.

### 3.2.1 Full mobile match on card

Deep Feature extraction methods use DCNN with many layers to extract deep representation of the face that is robust against face pose and illumination changes. On the other hand, deep feature extraction methods require giant dataset for training that is not available in every problem domains. It is where transfer learning comes to play an important role. In transfer learning, DCNN-based models can be trained on a large dataset in one domain and the learned knowledge can be transferred to another domain. In the era of face recognition, transfer learning has boosted the recognition quality drastically, and several high accurate architectures are already developed, that can be modified for other problem domains. One of these successful architectures is *FaceNet* that has achieved accuracy of 99.63% on LFW dataset with feature vector size of 128. Another interesting characteristic of transfer learning is that the verification phase of the model can be reduced to comparison with the classification threshold. These features pave the path for a full mobile match-on-card architecture where both enrollment and verification are done off-line, and on the SIM card without relying on any outside resources.

### 3.2.2 Performance evaluation

An extensive evaluation study has been conducted to assess the proposed system from different aspects such as performance accuracy in terms of EER, AUC and FNR @ FPR. Moreover, we evaluated our system under different verification scenarios such as single-platform and cross-platform. Effect of model size on the system accuracy is studied as well. The system shows EER of 0.1% under the single-platform scenario and EER of 0.2% under cross-platform scenario. Results of full MMOC architecture are also comparable with the results obtained by cloud-assisted MMOC, that shows a highly secure active authentication system using MMOC technique is feasible. To show the effect of the model's size on accuracy, we quantized our model with different bit-widths ranging from 8 bits to 2 bits. Applying 2-bit quantization, we gain about 93.75% reduction in memory footprint while keeping the system's EER less than 5% (i.e., about 1.5%) that is the acceptable EER for a reliable system. Moreover, the effect of quntization is more sensible for FPR$<$ 0.1% where more precision is needed to satisfy these strict FPRs. However, for FPR$>$ 1% this quantization error is less than 0.2%. The main bottle neck in the implantation of the full MMOC architecture is the enrollment phase, where an appropriate classification threshold that satisfies a predefined FPR in the system should be computed. Our results show that using 100 templates for thresholding, user enrollment takes about 15.8 seconds on the card. However, since the enrollment is only done once in the device setup phase, this delay will not affect the real-time verification response of the proposed system.

### 3.2.3 Platform evaluation

An android application and two SIM applets are developed to realize the proposed MMOC architecture on smartphones. By default, access to the SIM card is not granted to third party applications by the Android OS. However, a special applet called Access Rule Applet (ARA) can be developed on the SIM card with a list of privileged Android applications. The Android OS reads this list and grants access to a specific Android application defined in ARA. Therefore, we developed two SIM applets, one for enrollment and verification and another one for ARA. Moreover, a back-end python code is developed for model training in cloud-assisted MMOC. The Android application consists of MTCNN module for face recognition and a frozen model of *FaceNet*. The application in the background every 10 seconds captures an image and sends it to the SIM for active authentication. A phone-based version of the system is also developed that encrypts model's internals on the phone using AES algorithm. Our test bed is a Samsung galaxy A20. The result shows a slight improvement in resource consumption by using SIM-based solution. For instance, we gain 2% and 0.1% of CPU and

battery usage reduction, respectively.

## 3.3 Phase 3: A generic model for mobile secure authentication

In section 3.2, we described high accuracy of authentication systems using transfer learning. However, not all pre-trained models produce outputs that are implementable on constrained devices. In general, the transfer learning are suitable for platform with enough processing power. Therefore, in order to take advantage of the accuracy brought by this technique, we need to modify the network architecture of the model to output features that are suitable in size for SIM cards. Moreover, the classification sub-network of transfer learning consists of one or two fully connected layers that initially are not implementable on SIM cards; therefore, an on-card optimization technique is needed to compute the output of this sub-network in real-time. How to resolve these issues are discussed in the third article entitled «A Generic Model for Privacy-preserving Authentication on Smartphones», presented in chapter 6.

### 3.3.1 Transfer learning for MMOC authentication

A general model based on transfer learning is proposed to address the resource limitations on SIM cards. This architecture adds a dimensionality reduction layer on the top of the feature extraction network of transfer learning architecture. the reduction size is a hyperparameter in the system and can be defined using the validation set, also is a trade-off between the system's accuracy and the system's efficiency. Moreover, a quantization layer is added to the model's output in order to convert the real valued features to integers in a specific range, suitable for processing on the card. The obtained model is fine-tuned on the target dataset and can be frozen and used as a specialized feature extraction sub-network for MMOC authentication.

### 3.3.2 Optimization architecture

As mentioned in the beginning of section 3.3, classification sub-networks consist of one or two fully connected networks. More precisely, the verification phase of the system is the forward pass of the frozen fully connected network. However, this simple and fast computation on desktop computers, is a tedious task on smart cards that leads to a non real-time response of the authentication system. Therefore, an optimization architecture using the optimization techniques in modern compiler design, is employed to reduce the forward pass time of the classification sub-network on the card. Another, innovative solution is to use *log quantization* in which real valued variables are transformed to log domain. This transformation helps us to replace multiply-accumulate in vector dot product operation with bit-shift-accumulate that

is faster in hardware. Applying these two techniques for computation of the forward pass, a considerable speed-up gain is achieved that leads to a real-time secure authentication system.

### 3.3.3 Performance evaluation

We fine-tuned Resnet50 [33] trained on VGGFace2 [39]. The feature extraction sub-network of the model generates 2048 deep face features from each image. This vector size leads to a non real-time verification response on the card; therefore, in order to make it suitable for on-card implementation, we reduce its dimension to 64. Our optimization architecture contains 4 loop blocks and 16 weight vectors are fed into each block. Each loop is unrolled with unroll factor of 64. We also use ReLU-10 as layer activation function. We report the performance of the proposed system in terms of AUC, EER, REC, PER, and more importantly, verification time on the card for different classification sub-network size. The promising result comes from the optimization architecture, where we gain about $\frac{60.71}{1.37} \approx 44.3\times$ speed-up over the original architecture using the optimized architecture with log quantization. In general, quantization scheme reduces the performance accuracy of the system. For instance, in the worst case, we see about 0.7% increase in EER. Moreover, reducing the classification sub-network size decreases the accuracy of the system as well. In the worst case, it increases EER slightly for 0.8%. On the other hand, this reduction in the number of hidden layer nodes reduces the verification time considerably by 500 ms.

# CHAPTER 4 ARTICLE 1: MOBILE MATCH ON CARD ACTIVE AUTHENTICATION USING TOUCHSCREEN BIOMETRIC

Sepehr Keykhaie and Samuel Pierre

Mobile Computing and Network Research Laboratory (LARIM)
Department of Computer and Software Engineering, Ecole Polytechnique de Montréal,
Montreal, QC H3T 1J4, Canada
E-mail: sepehr.keykhaie@polymtl.ca; samuel.pierre@polymtl.ca.

Status: accepted for publication in IEEE Transactions on Consumer Electronics, October 2020.

**Abstract**

With the wide use of personal consumer electronics devices such as smartphones, people store sensitive and confidential information more on their devices. Active authentication (AA) systems continuously authenticate users to reduce possible attacks after a successful login on the device. In this paper, we propose match-on-card (MOC) approach for a secure active authentication scheme using touchscreen for smartphones to enhance the security and privacy and decrease the performance overhead on the consumer device. We train a Deep Neural Network (DNN) model, and store the model on the smart card available on the device for user authentication. To implement the user verification on smart cards, we quantize inputs to the model and the model's parameters. A speed-up technique is added to the verification phase to improve the execution time. Evaluation results show that with a well configured DNN model, our on-card authentication reaches an Equal Error Rate (EER) of 2.6% for 15 strokes and verification time of 0.65 second for each stroke. Considering the average user's stroke frequency of 1 stroke/s, our proposed scheme shows the potential for mobile MOC active authentication using touchscreen gestures on consumer devices.

**Keywords :** Biometrics, Authentication, Touchscreen gesture recognition, Smart cards, Neural networks.

## 4.1 Introduction

Increasing growth of personal consumer electronics (CE) devices usage such as smartphones, these devices store many sensitive and confidential information and run payment or banking applications. Therefore, an authentication scheme is required that only legitimate users can

access the sensitive information stored on devices. Traditional authentication techniques such as PIN/password or graphical patterns are prone to two types of security attacks known as *shoulder surfing attacks* and *smudge attacks* [1].

Recently, a new category of user authentication scheme based on user biometrics has received much attention. Biometric authentication uses physiological characteristics such as fingerprint [104], face [37], iris [105], palmprint [26], among many others or behavioral characteristics of user such as gait recognition [106], signature [1], gesture recognition on touchscreen devices [62], to name a few or hybrid schemes that take advantage of the both systems. Single entry point authentication systems only authenticate users at the beginning of a session (i.e., time until the next unlock) and do not authenticate the user during the active session. Therefore, an impostor may take control of the device during the session and access the user's sensitive information stored on the device. On the other hand, Active Authentication (AA) systems continuously and transparently authenticate users during the session using biometric traits. Most of the user's interactions with smartphones are through a touchscreen. Moreover, people have consistent and distinguishing behavior of performing gestures on touchscreens [1], [65]. These reasons plus the fact that no extra sensors are needed for user authentication on the device, make touchscreen biometric a promising behavioral biometric for active authentication on smartphones. Most mobile user authentication systems store biometric data on cloud servers or on smartphones, both threaten the privacy and security of users' identities. Adding to this, many consumers are unaware of attacks against mobile platforms, make the security and privacy of biometric data big concerns in biometric authentication systems [107]. Therefore, the implementation of an authentication system should be isolated from the mobile OS. Device manufacturers use a Trusted Execution Environment (TEE) for isolation of user verification. However, the TEE is not available on almost all CE devices. Smart cards (SC) in the form of SIM (Subscriber Identity Module) or eSIM (embedded SIM) cards available on almost all smartphones can be considered as a *dedicated* TEE to implement a secure biometric authentication system. Using SIM cards, we can securely store biometric templates, and do the user verification isolated from the mobile OS. Storing biometric templates on smart cards, we can take advantage of Match-On-Card (MOC) technique to enhance the security of the authentication system where *matching* (verification) is performed on-card and templates do not leave the card, which reduces the risk of biometric data leakage. Moreover, low resource consumption of smart cards makes them a good candidate for continuous authentication. However, smart card resource constraints make the implementation of a MOC authentication system a big challenge.

To overcome this challenge, we take advantage of processing resources available off-device. We train a Deep Neural Network (DNN) model on a cloud server and send the model's

parameters to the smart card on the consumer device through a secure channel. Moreover, in order to reduce the memory footprint of verification on SC's, we perform quantization on biometric templates and the model's parameters as well.

The main contribution of this work is to show the feasibility of implementing MOC active authentication systems on smartphones. The proposed system uses touchscreen sensors and SIM cards available on almost all smartphones while taking advantage of the security and the privacy features and low resource consumption of the latter. Therefore, the system is readily implementable on smartphones with no need of extra hardware on the device. The verification engine on the SIM card consists of a simplified DNN model to reduce the memory footprint, with a speed-up technique to improve the execution time on the SIM card. Most DNN models are implemented on specially designed hardware for Neural Networks. However, the proposed system implements a DNN model on off-the-shelf smart cards. Moreover, we also consider the implementation of the system on the emerging eSIM technology which are available on recent smartphones and smartwatches. We evaluate our proposed system on two touchscreen datasets to show its reliability and real-time performance.

The remainder of this paper is organized as follows. Section 4.2 describes our trust and threat models. Section 4.3 summarizes the related works on MOC authentication and touchscreen active authentication. In section 4.4, we describe the proposed system. Section 4.5 evaluates the system's performance. Section 4.6 briefly discusses eSIM implementation. Finally, we conclude our work in section 4.7.

## 4.2 Trust and Threat Models

### 4.2.1 Trust Model

In the proposed MOC authentication scheme, we trust the storage of biometric templates on the SIM card and on the cloud server, and the data path between the entities on the system. These assumptions are feasible in real implementations: 1) Access to SIM cards is restricted by an Access Control Applet (ARA) that contains applications' signatures that are allowed to access a specific file or applet on the SIM [108]. Moreover, recent SIM cards are tamper resistant and robust against side channel attacks. 2) A secure channel between on-card entities and off-card entities can be established using the protocol proposed in [109]. This channel can prevent Man In The Middle (MITM) and replay attacks in data transmission to the SIM card. 3) A secure channel between the on-device application and the cloud server is established to block possible attacks on the communication channel. The protocol design of a secure channel is out of the scope of the present study; however, an efficient protocol

is that both parties exchange their public key's certificates to authenticate each other, then using Elliptic Curve Diffie-Hellman (ECDH) protocol, an ephemeral session key is generated that is later used by Advanced Encryption Standard (AES) to confidentially exchange data between the application and the server. Moreover, the training dataset is stored encrypted on the cloud server. Hardware Security Module (HSM) on the cloud server can be used to protect the encryption key.

### 4.2.2 Threat Model

The main goal of the proposed system is to block any illegitimate usage of the consumer device as soon as possible. Therefore, our threat scenario consists of an attacker who uses the smartphone illegally in order to steal the owner's confidential, sensitive, or personal information stored on the device. The main attack we consider in this study is the spoofing attack. We divide it into several attack vectors, and describe how we can defend them in our system in section 4.5.

- *Shoulder surfing attack*: an attacker observes the device owner's pattern while interacting with the touchscreen.
- *Physical access to the device*: an attacker illegally accesses the biometric data stored on the owner's smartphone in order to impersonate the legitimate user.
- *Software attack*: an attacker runs a skilled software (e.g., a malware or a trojan) on the owner's device to capture user's behavior on the device.
- *Attacks on machine learning models*: an attacker uses *white box* attacks where an attacker has a knowledge of the model internals or its training data, or *black box* attacks where an attacker only can observe the model's output (i.e., predicted labels or the scores) to generate adversarial samples and fool the classifier.

## 4.3 Related Work

### 4.3.1 Match-On-Card authentication

Most studies on MOC authentication are based on fingerprint biometric with minutiae-based matching [93, 110]. Recently, Nedjah *et al.* [94] presented an efficient user authentication using fingerprint. Their method is based on Skin Elasticity Tolerant Algorithm (SETA) for fingerprint comparison. Some researchers, however, considered other physiological or behavioral biometrics for MOC authentication. Lee and Byun [98] used Support Vector Machine

(SVM) as a classifier for face authentication on memory-constrained devices such as smart cards. They used Genetic Algorithm (GA) to select most discriminative features for implementation on smart cards. Bourlai *et al.* [99] used Client Specific Linear Discriminant Analysis (CSLDA) technique for feature extraction. They also proposed several techniques to optimize their on-card verification system. Findling *et al.* [19] presented a MOC authentication approach using gait and face biometrics. They trained and simplified an offline model on a server and the simplified model is migrated to the smart card. Nedjah *et al.* [100] proposed an implementation of iris texture verification on smart cards using MOC technique. They used segmentation, normalization and binary code formation for iris feature extraction and, Hamming distance for comparison of iris codes. Choi *et al.* [102] proposed a speaker verification system based on SVM. They proposed a hardware design for the algorithm on FPGA platforms. Sabri *et al.* [101] proposed a multimodal biometric verification framework consisting of two MOC fingerprints and one match-on-host (MOH) face systems. Nedjah *et al.* [111] implemented a palm-print verification on smart cards. They extracted binary code (palm-code) from each palm-print image, and used Hamming distance for comparison. Table 4.1 compares recent studies on MOC-based authentication systems.

These works confirm the potential of MOC technique for secure authentication systems. This study differs from the current research efforts: (1) the existing works do not consider the integration of MOC technique on smartphones, whereas we propose a secure solution for smartphone authentication using SIM/eSIM cards; (2) the existing works investigate MOC technique for single point entry authentication; however, this work studies MOC potential for continuous authentication systems on smartphones; (3) the existing works use template-based or a simple linear model-based authentication that are not applicable to other biometrics, except the study by Findling *et al.* [19]. While their work uses some approaches similar to ours such as off-device model training and model simplification; however, their simplification scheme is not applicable on DNN model, whereas we implement a DNN model for user verification on cards with a quantization scheme that is also applicable to other machine learning models. Moreover, their proposed system is a single point authentication with gait biometric for authentication; however, we propose an active authentication system using touchscreen biometric.

Since smartphones are equipped with touchscreen sensors, this evolving biometric is considered as a primary method for active authentication. We review recent works in touchscreen active authentication in the next section.

Table 4.1 Comparison with MOC-based authentication studies

| Study | Year | Biometric | Type | Matching Algorithm | Best execution time (ms) | Card type (bit-length) | Best result |
|---|---|---|---|---|---|---|---|
| Pan et al. [110] | 2003 | Fingerprint | Physiological | Distance metric | 1600 | 32 | - |
| Lee and Byun [98] | 2003 | Face | Physiological | GA+SVM | - | - | FAR = 2.7% FRR = 9.4% |
| Choi et al. [102] | 2006 | Voice | Physiological | SVM | 58.7 | 32 | TER = 1.76% |
| Bistarelli et al. [93] | 2006 | Fingerprint | Physiological | Distance metric | 300-8000 | 16 | FAR = 0.1% |
| Bourlai et al. [99] | 2010 | Face | Physiological | CSLDA | 397.3 | - | HTER = 1.8% |
| Nedjah et al. [111] | 2017 | Palm-print | Physiological | Hamming distance | 1217 | 16 | EER = 0% |
| Nedjah et al. [100] | 2017 | Iris | Physiological | Hamming distance | 1210 | 16 | EER = 2.39% |
| Findling et al. [19] | 2018 | Gait | Behavioral | CSLDA | 824 | 16/32 | EER = 11.4% |
| Nedjah et al. [94] | 2019 | Fingerprint | Physiological | SETA | 2849 | 16 | EER = 21.11% |
| Sabri et al. [101] | 2019 | Fingerprint+Face | Physiological | SVM | 2640 | 16 | EER = 0.7% |
| Our system | 2020 | Touchscreen | Behavioral | DNN | 650 | 32 | EER = 2.6% |

### 4.3.2 touchscreen gesture active authentication

Frank *et al.* [62] proposed a continuous authentication scheme based on touchscreen gestures. They extracted 30 different behavioral features while the user is interacting with the device. Two different classifiers, k-nearest-neighbors (kNN) and SVM, are used in their work. They reached an average equal error rate below 4% depending on the scenario and the classifier used. Fierrez *et al.* [61] proposed a swipe gesture scheme for continuous user authentication. They captured strokes' features during a normal activity of a user. With SVM and Gaussian Mixture Model (GMM), similarity scores are computed and using different datasets, they evaluated the impact of different parameters such as number of swipes or number of training samples. Shen *et al.* [65] analyzed user's touch-interaction behavior for different touch operation types, operation lengths, application tasks, and different usage scenarios. The results showed that operations performed in small area on the screen are more reliable, increasing touch operation length improves authentication accuracy and error rate converges at 11 operation length . Antal *et al.* [63] showed that users' gestures on touchscreen devices can be used to classify users' identity, gender, and experience level in using the device. They depicted that identity, gender and experience level prediction reaches 95% accuracy with 10 or more strokes. Serwadda *et al.* [64] evaluated ten classification algorithms using the same dataset they gathered from 190 subjects. Among the algorithms, Logistic Regression and

J48 tree had the lowest and highest mean EER, respectively.

These studies show touchscreen biometric as a promising technique for active authentication on smartphones. On the other hand, MOC technique has shown the potential for secure user authentication. Therefore, integration of touchscreen biometric with MOC technique could arise as a secure active authentication system for smartphones. In the proposed MOC active authentication, computationally expensive operations are done off-card; therefore, same approaches for preprocessing, feature extraction and model training proposed in the related studies [62], [61] can be used in the MOC implementation. However, the verification phase of the model should be modified to make it suitable for offline (no communication with the outside world) real-time and accurate on-card user verification.

To the best of our knowledge this is the first work on MOC active authentication scheme with DNN model for smartphones.

## 4.4 System Description

A *stroke* or gesture is the movement of finger when it touches the screen until it is lifted. Using the available touchscreen device's APIs, we can capture useful data from user interaction with the touchscreen. Afterwards, we can exploit these data to extract distinguishing features for training a classifier. The classifier, then, is used for active authentication of the logged in user on SIM/eSIM cards.

### 4.4.1 Enrollment

During the enrollment phase, the authentication system collects enough touchscreen data from the legitimate user, and extracts specific features from each stroke. Features such as start and stop coordinates, stroke duration, average velocity, mid-stroke pressure, and etc., are extracted (see section 4.4.2). In order to increase the security of the proposed system, each template is quantized using the quantization scheme (section 4.4.3), and stored on the card. After sufficient number of strokes from the legitimate user is collected, templates are retrieved from the card and are sent to the cloud server for model training through a secure channel. To increase the security, during the enrollment, primary authentication method such as fingerprint or password is active on the SIM. Fig. 4.1 shows the architecture of the proposed biometric system.

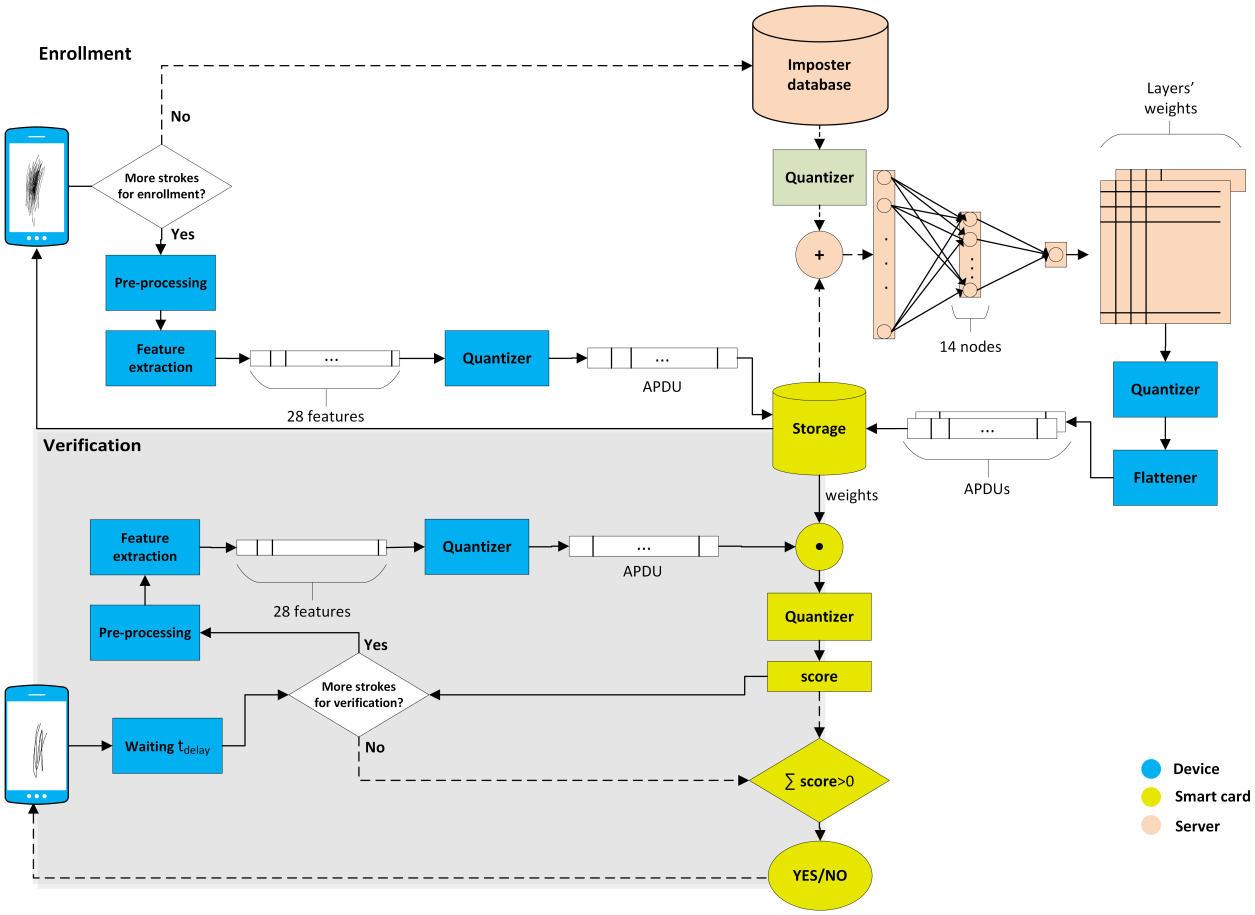



Figure 4.1 Architecture of the proposed MOC active authentication using touchscreen biometric with a DNN model for smartphones. In the enrollment phase, adequate strokes are collected, quantized and stored on the smart card. These templates are sent to the server for training along with the impostor database. The obtained model's parameters are sent to the device for quantization. The flattener module on the device flattens the weights' matrices to APDU format and sends them to the smart card. In verification phase, after $t_{delay}$, sampling starts, and the required number of strokes are sent to the smart card. For each sample, vector dot product is computed, quantized, and the score is obtained. Sum rule is used to compute the final verification decision. False conditions are shown by dashed lines in the figure.

### 4.4.2 Feature Extraction

Using the method described in [61], the most discriminative features of a stroke, to distinguish the legitimate user, are extracted that build up a 28-dimensional feature vector for each touch stroke. For instance, first-quartile, second-quartile, and third-quartile for velocity, acceleration, pressure on the screen, and the covered area vectors are computed. Moreover, start and stop coordinates, direct end-to-end distance, deviation form the striaght line, stroke duration are calculated. Table 4.2 shows the list of extracted features. Features are normalized using

tanh-estimators method [27].

### 4.4.3 Quantization scheme

As stated in section 1, smart cards have limited resources. Several challenges are:

- Communication between an off-card application and an on-card applet is a half-duplex operation through Application Protocol Data Unit (APDU) that restricts data transfer to 255-byte blocks (5 bytes for header and 250 bytes for data)
- Smart cards do not have Floating Point Unit (FPU). One-byte signed integer (int8) and 2-byte signed integer (int16) are supported in all smart cards. Some smart cards, however, support 4-byte signed integer (int32).
- Restricted resources on smart cards. For instance most smart cards' processors have efficiency below 2 DMIPS/Mhz[1]. Moreover, available storage on modern smart cards is around 40 KB of RAM, and around 1MB of EEPROM/Flash.

Considering the above-mentioned restrictions of smart cards, especially the lack of floating and fixed point arithmetic on smart cards, we quantize data that need to be transferred to the smart card. In order to respect the limitations on the card, avoid overflow, and improve inference (verification) time, real value data are quantized to 8-bit signed integer (int8) [112]. Assume that a floating point variable $r$ in range ($r_{min}$, $r_{max}$) needs to be quantized to int8 which has 256 quantization levels, i.e., $Q_{levels} = 2^k$, for k=8 we have $Q_{levels} = 256$. Real value $r$ is quantized to $r_q$ as follows.

$$r_q = Clip\Big\{round\Big(\Delta r + \Gamma\Big), -\Lambda, \Lambda - 1\Big\} \tag{4.1}$$

where $\Delta$, $\Lambda$, and $\Gamma$ are defined as

$$\begin{aligned} \Delta &= \frac{Q_{levels} - 1}{r_{max} - r_{min}}, \\ \Lambda &= Q_{levels}/2, \\ \Gamma &= -(\Lambda + \Delta r_{min}) \end{aligned} \tag{4.2}$$

[1]Dhrystone MIPS (Million Instructions per Second)

Table 4.2 Extracted features

| **Extracted features** |
|---|
| $Q_1$, $Q_2$, $Q_3$ area covered |
| Mean area covered |
| Standard deviation area covered |
| $Q_1$, $Q_2$, $Q_3$ pressure |
| Mean pressure |
| Standard deviation pressure |
| $Q_1$, $Q_2$, $Q_3$ pairwise velocity |
| Mean pairwise velocity |
| Standard deviation pairwise velocity |
| $Q_1$, $Q_2$, $Q_3$ pairwise acceleration |
| Mean pairwise acceleration |
| Standard deviation pairwise acceleration |
| $x$ coordinates (start and stop) |
| $y$ coordinates (start and stop) |
| Direct end-to-end distance |
| Angle of the stroke |
| Stroke duration |
| Length of trajectory |

*round(x)* stochastically rounds $x$ to $\lfloor x \rfloor$

$$round(x) = \begin{cases} \lfloor x \rfloor & \text{w.p } 1 - (x - \lfloor x \rfloor) \\ \lfloor x \rfloor + 1 & \text{w.p } x - \lfloor x \rfloor \end{cases} \tag{4.3}$$

*Clip* function is defined as

$$Clip(x, a, b) = max(a, min(b, x)) \tag{4.4}$$

Stochastic rounding is an unbiased rounding and has a nice property that $\mathbb{E}[round(x)] = x$ [113]. Suppose $r = -0.39$, in range [-0.73, 0.55], needs to be quantized to int8. We have, $\Delta = 199.2$, $\Lambda = 128$, $\Gamma = 17.4$, and $\Delta r + \Gamma = -60.3$. With probability of 0.7 it would round to -60 and with probability of 0.3 it would round to -61, that is already in int8 range; therefore, no clipping is applied. Although smart cards support signed *short* and *int* data types, data are transferred to smart cards in int8. Transferring int16 or int32 data incurs the casting cost on smart cards. Therefore, the extracted features of a stroke and the model's weights are quantized to int8 using Eq. (4.1). However, the model's bias is converted to 16-bit signed integer to retain the classification precision as much as possible in verification phase.

It causes the casting cost on the smart card; but, it happens once for transmission of the model's bias, and does not have a big impact on the whole authentication phase. Moreover, since the sign of the weights and the bias are important to determine that a probe belongs to the positive class (legitimate) or the negative class (impostor), they are not shifted, i.e., $\Gamma = 0$ in Eq. (4.1).

### 4.4.4 Classifier

After feature quantization, we send the extracted feature vectors, collected during the specific time interval, to the classifier on the cloud server through a secure channel. Among classifiers we select DNN model.

Deep Neural Network iteratively adjusts the weights to minimize the network error function in three steps. In *feed-forward*, the overall network function is computed. The input $x$ is fed into the network, and the primitive functions and their derivatives are evaluated at each node. In *back propagation* step, the network is run backwards in order to evaluate the derivatives of network function w.r.t $x$. *Parameter update* step computes gradient w.r.t each layer's parameters and makes adjustment to the parameters.

#### Model training

A verification system is a binary classification problem. The network function of a DNN model to compute the prediction score, $s_{pred}$, can be written as

$$y(\mathbf{x}) = \sigma\left(\phi(\mathbf{x})^{(L+1)}\right) \tag{4.5}$$

where $\sigma$ is the output activation function which squashes the prediction scores to [0,1], $\phi(\mathbf{x})$ is the basis function that transforms the feature-space, and $L$ is the total number of hidden layers. In DNN, the basis function is a nonlinear function of linear combination of the inputs [114]. That is

$$\phi(\mathbf{x})^{(i)} = h^{(i)}\left(\mathbf{w}^{(i)} \cdot h^{(i-1)}(\mathbf{x}) + b^{(i)}\right) \tag{4.6}$$

where $h(.)$ is the layer activation function, $h^{(0)}(\mathbf{x}) = \mathbf{x}$, $(i)$ defines the hidden layer number, and $i = 1...L+1$. Rectified Linear Unit (ReLU) is used as the layer activation function. For binary classification, *soft* or *hard sigmoid* [115] is used as the output activation function. In the proposed model, however, since the scores are later fused to reach the final verification decision, we consider $\phi(\mathbf{x})^{(L+1)}$ as prediction scores, $s_{pred}$, for score fusion, and no output activation function is applied. On the server, an impostor database is used along with the

legitimate user's samples to train a user specific model. Finally, the model's parameters for the user is sent back to the smart card.

### Model quantization

In the proposed scheme, the model is trained to obtain the full precision model's parameters that are quantized afterwards for the on-card inference. The inference of the network consists of the feed-forward phase with several convolution layers; therefore, bit-width overflow should be considered in intermediate computations. We quantize DNN model based on the following rules:

- The feature vectors are quantized to int8 using Eq. (4.1). $\Delta_{features}$ in (4.2) is computed using minimum and maximum values among all feature vectors, and $\Gamma_{features}$ is found accordingly. These values are calculated off-device and imported to the quantizer module on the smartphone for precise quantization.

- the model's weights and the bias are quantized in the same manner. However, since the weights' signs have key impacts on calculating matching scores, the model's weights and the bias are scaled but not shifted, that is, in Eq. (4.1) $\Gamma = 0$. The weights are quantized to int8, $Q_{levels} = 2^8$, maximum and minimum of the weights are considered as $r_{min}$ and $r_{max}$ to calculate $\Delta_{weights}$ in (4.2). The precision of the model's bias has a great impact on the final decision. Therefore, in order to retain the model's accuracy close to the full precision model, we quantize the bias to int16, $Q_{levels} = 2^{16}$ in (4.1), and scale it in the range of the weight vector, that is $\Delta_{bias} = \Delta_{weights}$.

- In the on-card inference module, the output of each layer is divided by $\Delta_{weights}$ and rounded down to the nearest integer in order to keep the quantized outputs close to the real value outputs. Moreover, to avoid overflow of the intermediate multiply-addition calculations in deeper layers, the output of layer activation function is clipped from the first layer. Therefore, the next layer is calculated using the clipped output that prevents overflowing. We use ReLU-$n$ as the layer activation function, defined as

$$h(x) = max(min(x, 0), n) \tag{4.7}$$

  where $-Q_{levels}/2 \leq n \leq Q_{levels}/2 - 1$ is obtained using validation set.

### 4.4.5 Authentication

As the user interacts with the touchscreen, the gesture detection application in background every sampling interval, $t_{delay}$, (e.g., every 30 seconds [44]) collects each stroke's features, quantizes them, and sends them to the smart card for authentication. In *single stroke* authentication, only one stroke is used for user verification, and the user is legitimate if $s_{pred}$ is greater than a classification threshold. Here, zero is considered as the baseline classification threshold; therefore, $s_{pred} > 0$ indicates the legitimate user.

It is shown that *multi-stroke* authentication improves the performance of a touchscreen authentication system remarkably [62], [65], [63]. Therefore, the score of each individual stroke is calculated, and a score fusion method is applied to reach the final authentication decision [116]. At each $t_{delay}$ interval, the application captures, quantizes and sends the stroke's features to the SC for authentication. In the authentication phase on the SC, the prediction score, $s_{pred}$, for the given template is calculated and stored. When all required strokes are presented to the smart card, using sum rule, the final prediction score is computed. The multi-stroke input belongs to the legitimate user if

$$\sum_{k=1}^{K} s_{pred_k} > 0 \tag{4.8}$$

where $K$ is the total number of required strokes. if the total score is less than zero, the authentication system consider the user as an impostor, and the primary authentication scheme (e.g., knowledge-based authentication) on the SIM is activated to check the authenticity of the user.

### 4.4.6 Verification speed-up

The most computationally expensive operation of the proposed method on the smart card is the vector multiplication to compute the prediction (matching) score. Consider the multiplication of two $d$-dimensional vectors $\mathbf{a}$ and $\mathbf{b}$ with entries $a_i$, $b_i$, $0 \leq i \leq d-1$, the inner product of the two vectors is defined as

$$c = \sum_{i=0}^{d-1} a_i b_i \tag{4.9}$$

A well-known method to improve the execution time of a loop is *loop unrolling*. A loop that its body is replicated $r$ times is called an unrolled loop with an unroll factor of $r$. Loop unrolling reduces execution time by reducing the number of loop termination test and

modifying the index variable fewer times. Moreover, the compiler can take advantage of operation pipelining [117]. Applying loop unrolling, we can rewrite (4.9) as

$$c = \underbrace{\sum_{i=0}^{\lfloor \frac{d}{r} \rfloor - 1} \sum_{j=0}^{r-1} a_{ri+j} b_{ri+j}}_{\text{Main iterations}} + \underbrace{\sum_{k=\lfloor \frac{d}{r} \rfloor * r}^{d-1} a_k b_k}_{\text{Leftover iterations}} \tag{4.10}$$

For example, suppose a dot product of two 100-dimensional vectors, and $r = 30$. Using loop unrolling, instead of checking loop termination and index modification $100 + 1$ times, the compiler interferes only 4 times, $\lfloor \frac{100}{30} \rfloor + 1 = 4$. We will have 30 main iterations ($j = 0...29$), and 10 leftover iterations ($k = \lfloor \frac{100}{30} \rfloor * 30 = 90...99$). Fig. 4.2 shows the effect of loop unrolling on the execution time for dot product of two 100-dimensional vectors. As seen in the figure, by increasing the unroll factor the execution time decreases. Using unroll factor of 50 , we achieve over 15× speedup on the vector dot product which is the main bottleneck to implement verification phase of biometric systems with large template size on cards. However, loop unrolling may have an adverse impact on the execution performance if the instructions of the unrolled loop overflow the instruction cache. Therefore, unroll factor should be selected carefully [118].

## 4.5 Performance evaluation

For evaluation, we use a SIM card with 1.5 MB of secure flash memory and 53 KB of RAM running Java Card version 3.0.2 that supports 32-bit integers. It has a secure processor and a secure OS which is tamper resistant and robust against side channel attacks and fault injections [20]. We use T=0 protocol to communicate with the card through a contact interface. We develop an applet on the SIM card for storing feature vectors, the model's parameters, and doing the verification (inference).

### 4.5.1 Evaluation datasets

We used two public touchscreen datasets available at TouchDB benchmark [61].

#### Frank dataset

*Frank* dataset [62] consists of touch data from 41 users collected in two sessions, one week apart. An application designed for reading documents and viewing images. Using phone's API, raw features are recorded during the user's interaction with the touchscreen (e.g., read-

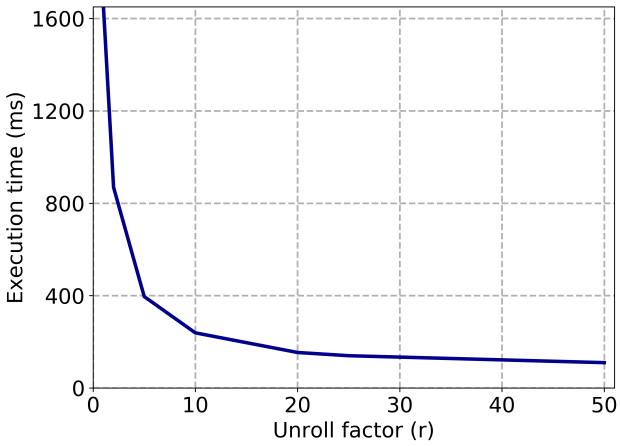



Figure 4.2 Execution time improvement using loop unrolling for dot product of two 100-dimensional vectors on smart cards.

ing the document or viewing images). Features such as time in *ms* of recorded action, where an action could be *touch down*, *touch up*, or *finger move*, $x$- and $y$-coordinates of each action, phone orientation (landscape or portrait), finger orientation, area covered by finger, and finger pressure on the touchscreen are extracted from each stroke. From the dataset analysis, we see that only 22.5% of the subjects used the phone in landscape orientation. All the subjects, however, used portrait orientation.

**Serwadda dataset**

*Serwadda* dataset [64] consists of touch data from 190 subjects in two session, one day apart. Two mobile applications were developed to collect touch data while users should answer to questions by scrolling/swiping between different screens. Vertical and horizontal strokes are considered and for each stroke raw touch data are recorded: $x$- and $y$-coordinates of points in the stroke, finger pressure on the screen, area covered by the finger, the time when finger touched or left the screen, and phone orientation. Similar to Frank dataset, landscape orientation is not used by all the subjects, and only 28% of the users interacted with the phone in landscape orientation. However, about 70% of the subjects used the phone in portrait mode.

### 4.5.2 Evaluation configuration

Features are extracted as stated in section 4.4.2. The datasets have data captured in two separate sessions. However, *intra-session* data are used for experiments in this work, where data from one session are used for model training and testing. Moreover, as discussed in

section 4.5.1, portrait orientation is dominating in both datasets; therefore, portrait mode is used for evaluation. In addition, since horizontal strokes are more discriminative than vertical strokes [61], only horizontal swipes are considered in our experiments. The dataset is split to 80%-20% training and test sets. In verification systems, we are interested in distinguishing a specific user (i.e., a legitimate user) from the other users (i.e., impostors). Therefore, using *one-vs-all* classification [119], a separate person-specific binary classifier is trained to distinguish the legitimate user form impostors. All stroke samples form the specific user are considered as "one" class while the other samples in the dataset are considered as "all" class. The person-specific model's parameters are then transferred to the SIM card for class prediction of unseen data in the test set. Applying one-vs-all method on the dataset, an imbalanced dataset is generated that will affect the overall accuracy of the biometric system [120]. In order to mitigate the effect of imbalanced dataset more weights are given to the "one" class that causes the model to pay more attention to the legitimate user's class; therefore, we assign weights proportional to the size of the class, i.e., $weight_P = \frac{P+N}{P}$ and $weight_N = \frac{P+N}{N}$ where $P$ and $N$ are positive and negative classes, respectively. Table 4.3 shows the average rate of legitimate users in Frank and Serwadda datasets. Our DNN model consists of one hidden layer of size 14 followed by a relu activation function, and one-node output layer. The binary cross entropy loss is minimized by ADAM with learning rate $\eta$ of $10^{-3}$. Moreover, 20% of the training set is used as the validation set for early stopping and model selection. We initialize the bias of the output layer such that it initially predicts reflecting the *legitimate:total* sample ratio. Therefore, we set the bias of last layer as $b^{(2)}_{init} = -\log_e(N/P)$. The batch size is 100, and to control bit-width overflow, using validation set, we defined $n = 10$ in (4.7). Our evaluation metric consists of Equal Error Rate (EER), and Area Under the Receiver Operating Characteristics Curve (AUC). Moreover, we also report well-known *threshold* metrics, namely recall (REC=$\frac{TP}{TP+FN}$), precision (PRE=$\frac{TP}{TP+FP}$), and F1 score (F1=2.$\frac{PRE.REC}{PRE+REC}$) where TP, FP, and FN are the number of true positive, false positive, and false negative samples for a given threshold, respectively.

### 4.5.3 Security analysis

In this section we describe how our proposed system is robust against the attacks we defined in our threat model.

- *Shoulder surfing attack*: an attacker may conduct a shoulder surfing attack by capturing a video while the user is working with the smartphone. It is possible that the attacker learns the coordinates of strokes; however, many of the extracted features such as pressure, acceleration, or velocity are hard to reproduce.

Table 4.3 Average legitimate users in dataset. Standard deviation is shown in parentheses.

| **Dataset** | **Average legitimate users (%)** |
|---|---|
| Frank | 2.4% (0.95) |
| Serwadda | 0.7% (0.3) |

- *Physical access to the device*: an attacker can access the stored legitimate biometric templates on the device and later replay them to spoof the active authentication system. However, our system defends this attack by using the MOC technique. Biometric templates are securely stored on the SIM card, and no biometric template leaves the SIM card for verification; therefore, the attacker does not have access to the biometric templates.

- *Software attack*: an attacker running a malware can monitor the memory or the device's sensors to steal the biometric information in the enrollment or authentication phase and replay them. Although the attacker obtains the legitimate user's biometric templates, he cannot impersonate the legitimate user in our system, because it is revealed that the software is unauthentic in channel establishment and even by the on-card access controller. Therefore, the attacker's malware is not able to reach our authentication module.

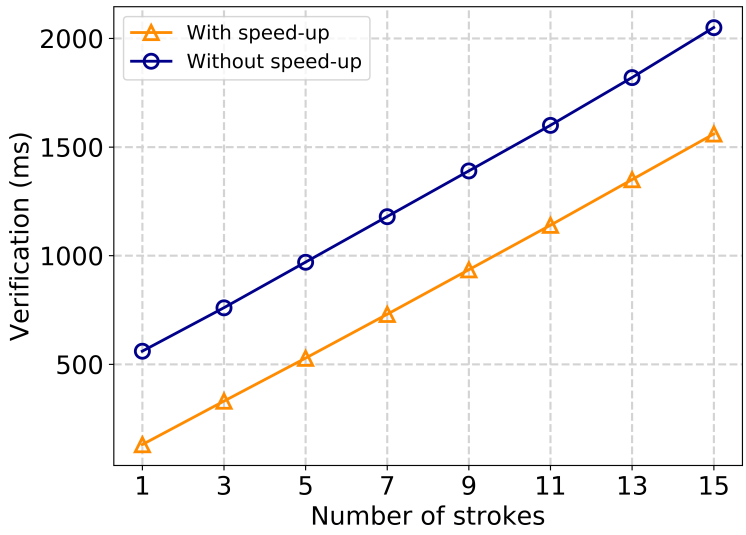
- *Attacks on machine learning models*: an attacker crafts adversarial samples in order to fool the model. White box attacks are defendable in our system since our training dataset and the model internals are securely stored. In case of the black box attacks, the attacker may be able to iteratively modify his samples towards the legitimate samples by reading the model's output from the running application. However, since the model is protected by an ARA on the SIM card, untrustworthy entities are not allowed to send inputs to the model; therefore, this attack needs code tampering or code injection techniques to modify the samples inside the trusted entity on the phone. In the first case, modifying codes requires application re-signing that ARA on the SIM could immediately detect it. However, in the latter case, the attacker injects the code directly into the process memory to craft his biometric samples before transmission to the SIM card. This highly sophisticated attack is feasible since modifying the memory content does not change the application's signature. Defence against black box attacks is out of the scope of our current work. However, there are some obstacles in our system that make this attack a hard task to accomplish: 1) the authentication module on the SIM

does not output prediction score to assist the attacker; 2) after $K$ strokes, if the score is below the threshold, the module activates the primary authentication method on the SIM (e.g., password or fingerprint). If the attacker succeeds in these steps, a simple yet efficient defence against this attack in our system is to revoke the application's certificate on the ARA using over the air (OTA) channel of the network operator. This way, we can block any communications of the suspected application with the authentication module on the SIM.

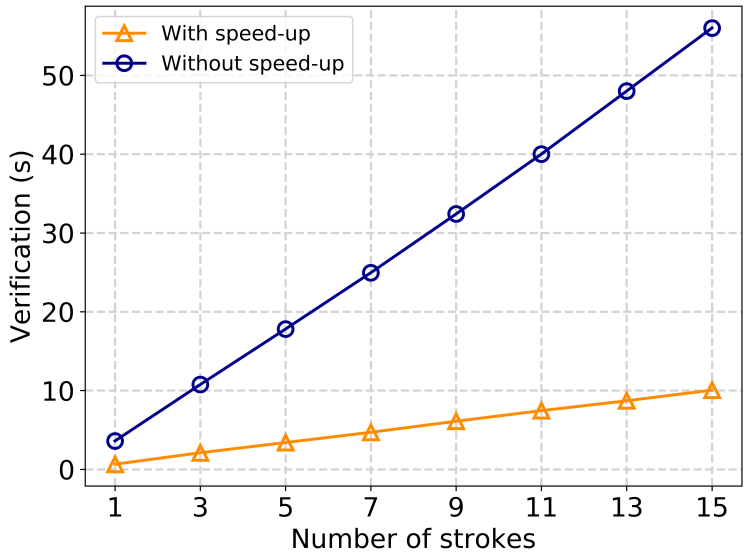
### 4.5.4 Experimental results

We compare our DNN model with SVM classifier as used by Frank *et al.* [62] and Fierrez *et al.* [61]. However, considering resource constraints of smart cards, we use SVM with linear kernel (L-SVM) to compare its performance on smart cards with the proposed DNN model. Using the validation set, the hyperparameter $C$ is set to $10^{-5}$, and same quantization scheme is applied for migrating the L-SVM model to the smart card.

Fig. 4.3 shows the verification time on the 32-bit SIM card for DNN and L-SVM models. As can be seen, loop unrolling decreases the execution time on SIM for both models. In Java Card, two dimensional arrays are not supported. In order to implement the matrix multiplication needed for DNN inference, weight matrix is stored column-wise in the SIM's storage; then, vector dot product of the input vector with each column vector of the weight matrix is computed and added together to obtain the final result. The unroll factor is set to the length of vectors in dot product of each layer. Since we have 15 vector dot products for each stroke (14 for the hidden layer and 1 for the output layer) in DNN, we observe a considerable increase in computation time as the number of strokes increases comapred to L-SVM. For example, using 15 strokes execution time goes to 56.2 s (std = 10.5 s), whereas it is about 2.0 s (std = 0.2 s) in L-SVM. Using loop unrolling technique, however, the execution time decreases by 82% where the execution time for 15 strokes drops to 10.1 s (std = 1.6 s) in DNN. Please note that all strokes are not ready for verification at once. A median user makes a stroke at least every 1.0 second [62]. Using the speed-up technique, an individual stroke's score is computed in about 650 ms (std = 25 ms) before the next stroke arrives, and the final score is obtained using Eq. (4.8). Therefore, the inference execution time on SIM does not affect the overall verification response. This duration includes transmission of quantized stroke's features using APDU communication, calculation of matching score for each stroke, layer output quantization, and computing the verification result. In Java Card, we have RAM and EEPROM/FLASH for storing data. EEPROM has physical write limit, and the write time on EEPROM is more than thirty times slower than writing on




(a) L-SVM

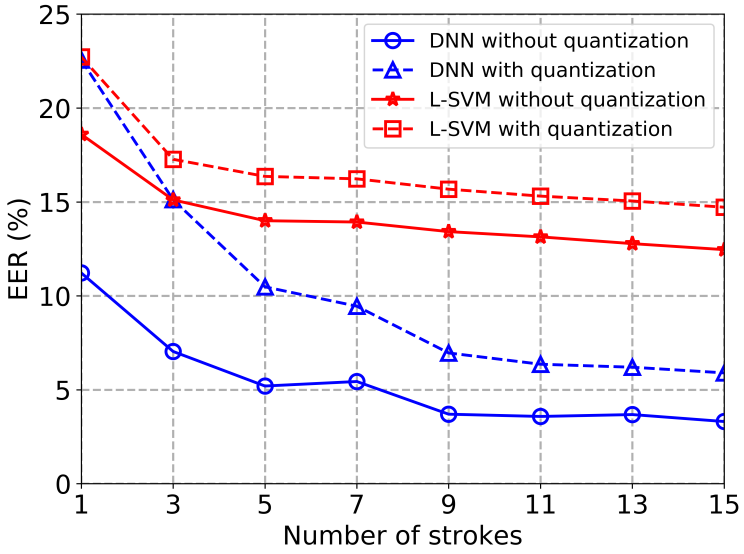



(b) DNN

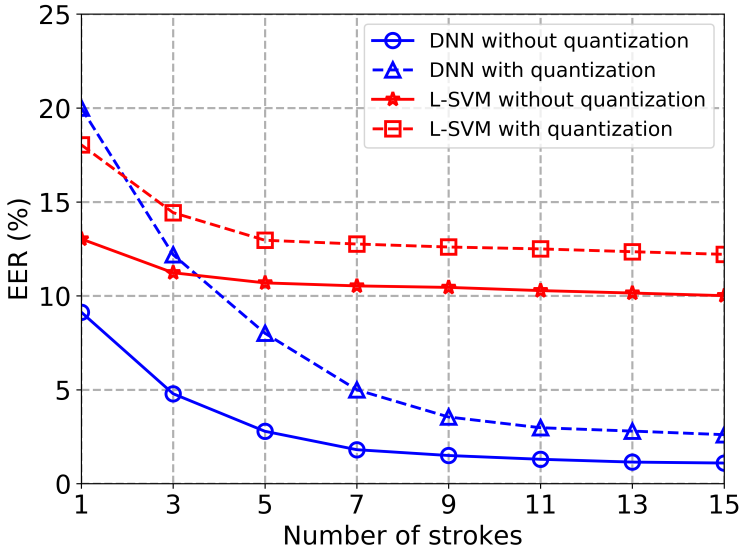
Figure 4.3 Verification time on SIM card with and without speed-up for different number of strokes.

RAM [121]. Therefore, in order to make a good use of the memory on chip and extending the card's lifetime, model's parameters are stored in EEPROM once received form the server; however, the probe's template and prediction scores are stored in RAM.

Fig. 4.4 shows the effect of model quantization on EER for Frank and Serwadda datasets with multiple strokes for authentication. In Frank dataset, using a single stroke, EER is around 11.2% for DNN and 18.6% for L-SVM. Using more strokes for authentication the




(a) Frank dataset




(b) Serwadda dataset

Figure 4.4 Effect of multi-stroke and quantization on EER using DNN and L-SVM classifiers.

EER decreases. After 10 strokes, the system is stabilized, and it reaches EER of around 3.5% afterwards. The best EER achieved with 15 strokes. Our DNN model performs better on Serwadda dataset where it is stable from stroke 13, and mean EER decreases to 1.1% for the real model, and to 2.6% for the quantized model, whereas it is about 10.1% for real L-SVM and 12.3% for the quantized one. Moreover, multi-stroke improves system performance more on Serwadda dataset than on Frank dataset, it can be seen in the figure where

lines' slopes connecting strokes are steeper on Serwadda dataset. The figure shows that the quantization scheme affects our system's accuracy; however, adding more strokes reduces the quantization effect on model accuracy. The reason is that the fusion technique sums several scores that results a robust final score (i.e., farther form the threshold); therefore, quantization ruins the systems' accuracy less in multi-stroke mode. As can be seen in the figure, L-SVM performs worse than DNN due to the fact that the linear kernel degrades the classifier accuracy compared to the RBF kernel proposed in [62] and [61]. Table 4.4 compares the systems' EER and AUC on different datasets.

Table 4.4 also reports the system's performance in terms of PRE, REC, and F1 score for different number of strokes. In authentication systems, identifying an impostor as a legitimate user, i.e., false positive rate (FPR), is more vital than falsely classifying the smartphone's owner as an impostor, i.e., false negative rate (FNR). However, since we do not have many of the legitimate user's samples compared to the impostors' samples in our dataset, we give more weights to the positive class to mitigate the effect of imbalanced dataset. Our model reports higher PER in the system while letting some positive samples falsely identified as an impostor, i.e., lower REC. In both datasets, increasing the number of strokes increases REC and PER accordingly. Moreover, in both dataset, the FPR is lower than FNR and we see higher value for PER compared to REC. For the baseline threshold as 0 (see section 4.4.5), in Frank dataset, the best results are achieved by 15 strokes with mean REC and PRE of 55.6 % and 93%, respectively. Likewise, in Serwadda dataset, the best results belong to 15 strokes with REC and PRE of 52.4% and 95.8 %, respectively. The table also reports the corresponding F1 score for REC-PRE pairs in the dataset.

## 4.6 eSIM implementation

Embedded SIM cards (eSIMs) are the most recent evolution of SIM cards with a new form factor. These devices unlike SIM cards are not transferable and are permanently soldered directly into the device. ESIM acts as a container of several Mobile Network Operator's (MNO's) profiles that contains information to authenticate subscribers on the network. They may also contain applet for value-added services (VAS). This specification allows users to change their network operator if they are not satisfied with the MNO's services, and maintain connection with a SIM profile that has been added to the eSIM, without having to remove the SIM from the field. Since eSIMs support Java Card technology [122], our active authentication applet executable (.cap) file can be added to a SIM profile and downloaded to the eSIM. Moreover, eSIM's remote provisioning system enables us to transfer the MNO's profile and its content to a new device. However, if the user decides to switch to another

Table 4.4 Evaluation results of the proposed MOC active authentication for Frank and Serwadda datasets. Real values are depicted in square brackets. No quantization is applied to obtain the real value results.

| Dataset | Strokes | EER (%) | AUC (%) | PRE (%) | REC (%) | F1 (%) |
|---|---|---|---|---|---|---|
| Frank | 1 | 22.5 [11.2] | 84.2 [93.4] | 74.2 [83.5] | 16.7 [25.3] | 27.3 [38.8] |
| | 3 | 15.1 [7.0] | 91.2 [95.8] | 85.7 [92.2] | 30.5 [41.9] | 44.9 [57.6] |
| | 5 | 10.4 [5.2] | 93.0 [96.6] | 89.1 [94.0] | 37.1 [48.9] | 52.4 [64.3] |
| | 7 | 9.4 [5.4] | 94.1 [96.5] | 89.9 [94.6] | 42.3 [52.2] | 57.5 [67.3] |
| | 9 | 6.9 [3.7] | 95.8 [97.1] | 91.2 [95.2] | 46.9 [55.5] | 61.9 [70.1] |
| | 11 | 6.3 [3.5] | 96.0 [97.5] | 92.3 [96.1] | 51.9 [58.5] | 66.4 [72.7] |
| | 13 | 6.2 [3.5] | 96.0 [97.5] | 92.5 [96.3] | 53.7 [60.3] | 68.0 [74.2] |
| | 15 | 5.9 [3.4] | 96.1 [97.5] | 93.0 [96.3] | 55.6 [60.9] | 69.6 [74.6] |
| Serwadda | 1 | 20.0 [9.1] | 86.2 [96.2] | 74.5 [83.1] | 12.3 [20.6] | 21.1 [33.0] |
| | 3 | 12.1 [4.7] | 92.7 [98.5] | 85.9 [93.9] | 24.0 [32.6] | 37.5 [48.4] |
| | 5 | 8.0 [2.7] | 95.2 [99.2] | 90.3 [97.0] | 36.2 [43.0] | 51.6 [59.6] |
| | 7 | 5.1 [1.8] | 96.6 [99.8] | 91.7 [97.8] | 46.0 [50.5] | 61.3 [66.6] |
| | 9 | 3.5 [1.5] | 97.5 [99.8] | 93.1 [97.2] | 50.1 [55.2] | 65.1 [70.4] |
| | 11 | 2.9 [1.3] | 98.1 [99.8] | 93.4 [97.1] | 51.4 [55.7] | 66.3 [70.7] |
| | 13 | 2.8 [1.1] | 98.2 [99.9] | 95.3 [97.5] | 52.2 [56.8] | 67.5 [71.7] |
| | 15 | 2.6 [1.1] | 98.6 [99.9] | 95.8 [97.5] | 52.4 [56.9] | 67.7 [71.8] |

operator by activating the profile, the authentication system does not respond if the module is not accessible on the new profile when switching the profile, which requires a solution to transfer applets and their contents between different profiles on eSIM.

## 4.7 Conclusion

In this paper, we studied a MOC active authentication scheme using touchscreen biometric for smartphones. We trained a DNN model on a cloud server and migrated the model internals to the SIM card. In order to make it feasible to implement the verification on the card and to reduce the memory footprint, we applied a quantiaztion scheme. Using the quantized DNN model, we reached the best EER of 2.6% on the card. A speed-up technique helped us to gain up to 5.5× speed-up over the original computation. Furthermore, the results revealed that single stroke verification time (0.65 second) is less than median user's stroke frequency (1 stroke/s); therefore, we can sequentially compute each stroke's score and apply the sum rule when all strokes are presented to the card, which shows that SIM execution does not hinder active authentication process.

In conclusion, our results show that we can implement a lightweight and secure touchscreen

active authentication on off-the-shelf SIM/eSIM cards. Furthermore, our study is the first work that shows a quantized DNN model can be used in match-on-card biometric authentication to improve the accuracy of the system where DNN performs better. In future, we are going to improve the proposed model to implement deeper on-card neural networks with lower execution time and lower quantization error.

# CHAPTER 5 ARTICLE 2: LIGHTWEIGHT AND SECURE FACE-BASED ACTIVE AUTHENTICATION FOR MOBILE USERS

Sepehr Keykhaie and Samuel Pierre

Mobile Computing and Network Research Laboratory (LARIM)
Department of Computer and Software Engineering, Ecole Polytechnique de Montréal,
Montreal, QC H3T 1J4, Canada
E-mail: sepehr.keykhaie@polymtl.ca; samuel.pierre@polymtl.ca.

Status: submitted to the IEEE Transactions on Mobile Computing journal, June 2020.

**Abstract**

Active Authentication (AA) systems continuously authenticate users on smartphones. With high quality front-facing cameras available on recent smartphones, face-based active authentication emerges as a good candidate for AA systems. On the other hand, secure authentication of mobile users is a big concern in biometric systems. Mobile match-on-card (MMOC) technique takes advantage of SIM/eSIM card as a secure element (SE) to protect biometric templates and verify users isolated from the smartphone's environment. However, resource limitations of smart cards make MMOC authentication hard to implement. In this paper, we propose two system architectures for MMOC face-based AA systems. In Cloud-assisted MMOC architecture (CA-MMOC), we use cloud resources for model selection and training. Full MMOC architecture (F-MMOC) relies only on SIM/eSIM card's resources for enrollment and verification. A quantization scheme is proposed to make the authentication system implementable on SIM cards, plus a speed-up technique to reduce on-card execution time. Using a public mobile video dataset, we evaluate the proposed system. Our evaluation results show that the proposed MMOC authentication achieves high accuracy in real-time with a small memory footprint on SIM, and is suitable for cross-platform authentication. We also implement the CA-MMOC system on a real smartphone and evaluate the system's performance overhead in terms of power consumption, CPU and memory usage.

**Keywords :** Active authentication, Secure authentication, Face Biometric, SIM/eSIM, Mobile device.

## 5.1 Introduction

With the growing use of mobile devices, people use their smartphones for different services such as social apps, shopping, or banking applications. Therefore, they store their credential and banking information on their phones that require a secure authentication technique. Biometric authentication is a new authentication scheme that uses physiological or behavioral characteristics of a user for authentication. Traditional Authentication systems unlock the device at the beginning of a session and lock the device as user closes the session or if the device is idle for a certain amount of time. This gives an impostor an opportunity to take control of the phone by stealing the phone and continuing the active session. Active Authentication (AA) systems, on the other hand, continuously and unobtrusively authenticate users during an active session that reduces the chance of unauthorized access to the phone after a successful authentication. AA systems mostly rely on face ( [42], [45], [41], [43], [44]), touchscreen gestures ( [123], [61], [62], [124]), or inertial sensors such as accelerometer, gyroscope, and gravity sensor ( [46], [75], [47]) for transparent user authentication.

Privacy of biometric templates and secure verification of users' identities are big concerns in mobile platforms. Around 75% of users download applications from official repositories who believe that downloaded applications from these repositories are secure [7]. However, studies show that security control and application testing are not enabled in all official repositories or are inadequate [8]. Moreover, due to the increasing number of applications, validation techniques become more and more complex. On the other hand, not all users are knowledgeable and even not aware of the consequences of installing a spyware or a trojan on their phones. Interestingly, only 36% of smartphone users consider themselves as responsible for the security of their devices and the sensitive information stored on them [7], [9]. McAfee reports the increasing number of banking trojans that take advantage of Android vulnerabilities [10]. Since Android OS dominates the market share with more than 80%, most of cyberattacks target Android OS [125]. These studies and many other studies show the untrustworthiness of mobile platforms, which implies the necessity of a secure authentication scheme for mobile users.

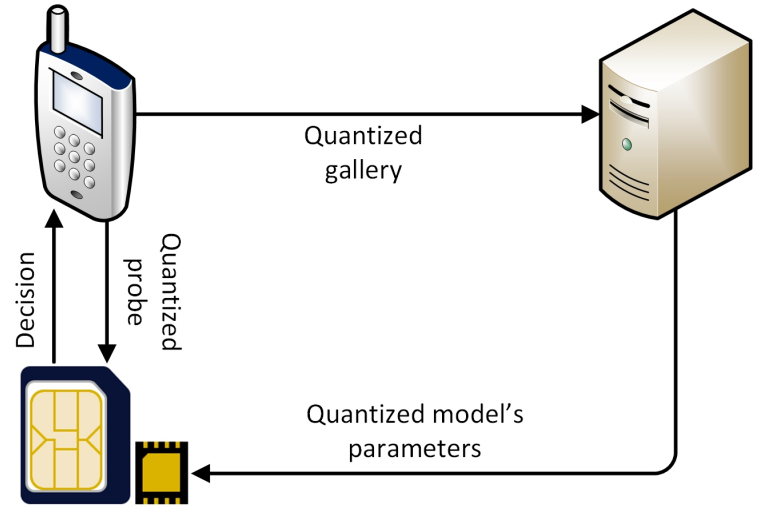
Most studies have considered software-based approaches to design a secure authentication system. However, these approaches have shown limitations in secure authentication systems. Cancelable biometrics [84] suffer from low verification accuracy, due to the template transformations; fuzzy commitment [11] schemes have information leakage; and homomorphic encryption [12] generates non real-time verification response, due to the expensive computations. Smart cards (available on smartphones) have good security characteristics that make them suitable for security services. They can be used to implement an authentication system

isolated from the mobile OS in a Secure Element (SE). A SE is a chip with a secure micro controller running a secure OS. It ensures that data are stored and processed in a safe place and only authorized applications or users are allowed to access it. There are three types of SE on smartphones: 1) Embedded Secure Element (eSE) are integrated into smartphones by device manufacturers and are used for storing payment information and key generation in payment wallets such as Google Pay or Apple Pay, or for storing biometric data and user verification. 2) SIM/eSIM as a secure element (SaaSE) takes advantage of SIM/eSIM cards available on all smartphones (see section 5.2). This type of SE, in the hands of Mobile Network Operators (MNOs), can be used for payment wallets, and also for biometric template storage and matching on the card, which needs more attention with the growing use of wearable devices connected to mobile networks using eSIM technology. Throughout the paper, if a specific consideration is needed, we use the specific term; otherwise, the term SIM is used to refer to both SIM and eSIM technologies. 3) Host Card Emulation (HCE) is considered as a secure cloud storage for mobile users, which is of interest to Cloud Operators (COs). They are used for storing users' payment information with an assumption of permanent secure channel between client and server. The fact that eSE is not available on all smartphones, and a secure communication channel is not permanently available for HCE, plus nice properties of SIM/eSIM cards such as security and privacy, availability, profile transferability, low performance overhead, and inexpensiveness make the SaaSE a promising technology for secure authentication on smartphones. Using SIM/eSIM cards, we can securely store biometric templates, and verify users isolated from the mobile environment. Only the verification response is sent to the mobile OS. Access to the SIM/eSIM is very restricted, and only mobile applications with right privileges can access it [14].

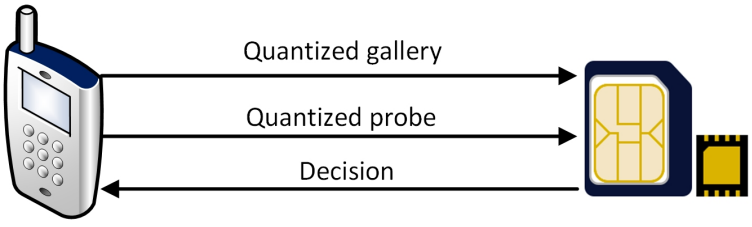
Using SIM cards for biometrics, a less secure solution known as Mobile Template-On-Card (MTOC), is to use the SIM card only for secure storage of biometric templates, while matching (user verification) is done outside the card by transferring the biometric templates outside the SaaSE. In a more secure solution known as Mobile Match-On-Card (MMOC), template storage and matching are done on the card; therefore, no biometric information leaks into the untrustworthy mobile environment. However, SIM cards have restricted resources in terms of processing power and memory, which make the MMOC implementation a challenging task.

In this paper, we propose an MMOC face-based active authentication system. A DCNN model for distance metric learning is outsourced for training, and is used to extract the most discriminative features of captured images on the device. These features are then quantized for enrollment and verification on the card using our quantization scheme. We propose two architectures for user authentication. In the first system, we train a machine learning model on a cloud server with the quantized features, then store the quantized model

on the SIM card for verification. In the second architecture, we use distance metrics to learn the best classification threshold to distinguish legitimate users from impostors on the card. No biometric template leaves the SIM card in the second model (see Fig. 5.1). With the profile transferability feature of SIM/eSIM cards, the proposed system is suitable for cross-platform authentication, where it can be trained on one device and used for verification on other devices without requiring to retrain the model. In order to implement a real-time verification on the card, we also introduce a speed-up technique to the authentication system. The authentication system continuously captures images of the user interacting with the device and sends them to the card for verification. On the card, a verification window is defined which stores each image's score and slides over the scores to compute the final decision score. Average score fusion is applied to obtain the final score. If this score is below the decision threshold, the current user is a legitimate user; otherwise, she is an impostor, which triggers the primary authentication scheme on the card (e.g., PIN/password).




(a) Cloud-assisted MMOC architecture




(b) Full MMOC architecture

Figure 5.1 Overview of the proposed secure authentication systems.

This paper makes the following contributions:

- We propose two architectures for MMOC face-based active authentication system using SIM/eSIM cards available on smartphones: 1) an architecture that uses cloud resources for enrollment and SIM/eSIM for verification; 2) an architecture that only relies on SIM/eSIM card resources for enrollment and verification.
- We introduce a quantization scheme to reduce the model size up to 93.75% on the SIM and a speed-up technique for real-time on-card responses that 8.5× improves the enrollment time.
- We evaluate the performance of the proposed architectures with different classifiers and different distance metrics on a publicly available mobile face dataset.
- We implement the MMOC face-based AA system on a real smartphone and evaluate the platform overhead of the system. To the best of our knowledge, this is the first implementation of MMOC face-based active authentication on real devices.

The remainder of this paper is organized as follows. Section 5.2 briefly introduces SIM and eSIM technologies. Section 5.3 reviews MOC-based authentication systems. In section 5.4, we describe two proposed MMOC face-based active authentication systems, the quanitzation scheme, and the speed-up technique. Section 5.5 evaluates the proposed active authentication system in terms of accuracy and execution time. Section 5.6 implements the proposed authentication scheme on an Android device and evaluates its performance overhead. We discuss the advantages and disadvantages of the two proposed systems in section 5.7. Finally, we conclude our work in section 5.8.

## 5.2 Overview of SIM/eSIM cards

A Subscriber Identity Module (SIM) card is used to securely store an MNO's *profile* that contains information to authenticate subscribers to the network, who are identified by international mobile subscriber identity (IMSI). These modules resemble computers in small scales; they have CPU, RAM, ROM, and EEPROM/Flash running a lightweight yet secure OS. They may also contain applets for value-added services (VAS). These modules are transferable between different mobile devices. This feature of SIM cards is promising in designing a secure and cross-platform authentication system. If a user decides to transfer her operator's profile to a new device, she easily inserts the SIM card into the new device. The profile and all personal information are transferred to the new device.

An Embedded SIM (eSIM), the most recent evolution of SIM cards with a new form factor, is permanently soldered directly into the device. eSIM acts as a container of several SIM

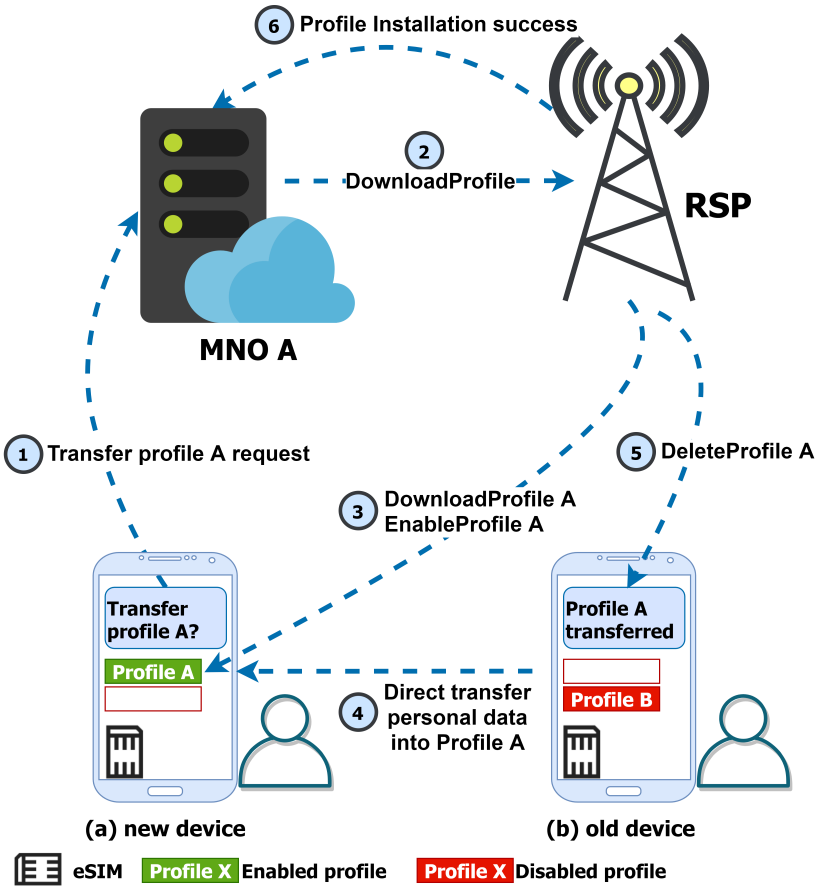



Figure 5.2 eSIM profile transfer. The user decides to change her device. She needs to transfer MNO A's profile to the new device's eSIM. Using a mobile application, she sends the request to MNO A (step 1). MNO A sends DownloadProfile command to RSP (step 2). RSP downloads the required information, installs, and enables MNO A's profile on the device (step 3). Personal data is directly transferred into the profile (step 4). It also deletes profile A from the old device and notifies MNO A of the successful operation (steps 5 and 6).

profiles. They are managed remotely by a platform called remote SIM provisioning (RSP), which enables storage and management of multiple MNO profiles [126]. This specification allows users to change their network operators if an outage occurs in the network, or if the network bandwidth is not satisfactory, and maintain the connection with a SIM profile that has been added to the eSIM without having to remove the SIM from the field. Transferring an eSIM profile and its content, in order to have a cross-platform authentication, is not an easy manual task as in SIM cards, and it needs attention from the RSP to complete the profile transfer. Fig. 5.2 shows the profile transfer process in eSIM technology.

Since an MNO's profile information does not occupy a lot of space on SIM/eSIM cards, we can take advantage of this opportunity to use these devices for secure and privacy-preserving

authentication on smartphones.

### 5.3 Related Work

Most of studies have considered fingerprint for secure authentication on smart cards since fingerprint templates are smaller and less information is processed by the card [90–94]. In 2000, Li in his Ph.D thesis proposed the first face-based authentication system on smart cards. He proposed Client Specific Linear Discriminant Analysis (CS-LDA) method for an on-card face recognition system. He showed the simplicity of the training phase, and the possibility of enrollment insulation which is suitable for smart cards. Afterwards, researchers paid more attention to low memory footprint matching algorithms and on-card real-time verification. In 2003, Lee and Byun [98] proposed to use Genetic Algorithm (GA) for feature extraction to reduce the amount of memory by storing less number of features while having low error rate. Several year later, Bourlai et al. [99] studied the system optimization for face authentication on smart cards. They evaluated the effect of image resolution reduction and image compression on the system's accuracy and real-time card response. Apparently, reducing the image resolution drops the system's accuracy.

The above-mentioned works were domain specific and a proposed method in one work is not applicable to other biometrics. Findling et al. [19], in 2018, addressed this issue in MOC authentication systems and proposed a generic approach for biometric MOC authentication. A CSLDA model was trained off-line, and the simplified model was migrated to the smart card for authentication. They applied the proposed method to face authentication and acceleration based gait authentication. They achieved 11.4% and 2.4-5.4% EER for gait respectively face authentication, with 2 s respectively 1 s for computation on SCs. However, they did not evaluate their method on a real challenging mobile face dataset with varying poses and illuminations.

Biometrics other than fingerprint and face have been less investigated in the literature. In 2017, Nedjah et al. [100] implemented an iris texture verification on smart cards. They used Hamming distance for iris code comparison. To decrease the execution time on smart cards, they proposed an acceptance threshold that terminates the execution when the comparison reaches the threshold. In another work, they implemented an efficient palm-print verification on smart cards [111]. They extracted binary code representing each palm-print image, then using Hamming distance, they compared the palm-print probe with the templates enrolled in the system.

Face-based Active authentication has been investigated in several works. In 2012, McCool

et al. [43] introduced a valuable publicly available audio-visual dataset for mobile phones. Later in 2015, Crouse et al. [44] proposed a face-based continuous authentication system. They fused captured face images with sensory data such as gyroscope, accelerometer, and magnetometer. They introduced login score $s_{login}$ which is updated periodically as a new face image is captured every $t_{sample}$. In the same year, Fathy et al. [40] studied face-based active authentication on smartphones using a face video dataset captured by the device's front camera. Their study showed a significant drop in recognition rate when user is enrolled in one session and verification is done on other sessions. In the next year, Mahbub et al. [41] evaluated face-based active authentication on UMADD-02 dataset which contains data collected from three sensors on the phone: front camera, touch sensors, and location sensors. They reported the best Equal Error Rate (EER) of 18.44%. In 2017, Samangouei et al. [42] presented a method for face-based active authentication using facial attributes. For each facial attribute, a set of features is extracted and a classifier is trained on it. Recently, Perera et al. [45] presented a method for face-based multiple user AA systems based on Extremal Openset Rejection (EOR).

Securing active authentication has not been investigated in the literature. These systems require real-time response, in a way that verification processing does not affect the verification frequency. Moreover, energy consumption management is crucial in active authentication. The software-based solutions due to their own limitations (discussed earlier in section 5.1) do not show the feasibility for securing active authentication systems. In 2016, Findling et al. [103] presented an MOC approach for acceleration based mobile gait recognition using 16-bit smart cards. The obtained model can be used on smart cards without requiring retraining. Although gait biometric can be used for active authentication, the authors did not proposed a model for it and only used it in single-point authentication. Recently, Keykhaie and Pierre [6] showed the feasibility of MOC-based secure active authentication using touchscreen biometric with small template size. In their work, they employed a simplified DNN model for on-card active authentication.

## 5.4 MMOC face-based active authentication

### 5.4.1 Image preprocessing and feature extraction

The key point in a good face verification system is the feature extraction technique. Considering the resource constraints of SIM cards, we should offload the heavy computations from SIM cards, and use this SE for secure verification and template storage. Therefore, we extract features off the card and on the device. Moreover, in order to have an accurate

face authentication system, we need to preprocess captured face images before extracting features. We detect the user face (if multiple faces are detected, the closet one is selected as the user face), and crop it to the right size before the feature extraction.

Deep feature extraction methods such as *OpenFace* [127] and *FaceNet* [36] using Deep convolutional neural networks (DCNN) with multiple layers, output a deep representation of the face that is robust against face pose and illumination changes. A loss function such as *Euclidean-distance-based* or *cosine-margin-based* loss is introduced in the network to make features more discriminative. In general, the network is trained on a giant dataset with many identities that tries to learn deep feature vectors using the loss function. Since people have similar face shapes and colors, the representation learned on the network can be applied for real world face recognition [128]. In this paper, we use Facenet to extract deep features for on-card verification.

*FaceNet* maps the face image $x$ into a compact feature space $\mathbb{R}^d$. FaceNet architecture consists of a deep CNN followed by $L_2$ normalization that produces face embedding $f(x) \in \mathbb{R}^d$, which embeds a face image into a $d$-dimensional Euclidean space. Triplet loss is introduced during training that minimizes the Euclidean distance between the *anchor* ($x_i^a$) and the images of same identity ($x_i^p$), and maximizes the distance between the anchor and the images of different identity ($x_i^n$). It is defined as

$$L = \sum_{i}^{N} \Big[ \|f(x_i^a) - f(x_i^p)\|_2^2 - \|f(x_i^a) - f(x_i^n)\|_2^2 + \alpha \Big]_+ \tag{5.1}$$

where $N$ is the number of samples in the training set and $\alpha$ is a threshold between positive and negative samples. In [36], $d$ (Face embedding dimension) is 128. FaceNet achieves accuracy of 99.63% on Labeled Faces in the Wild (LFW) dataset and 95.12% on YouTube Faces DB.

The verification can be done using linear classifiers or threshold comparison, which promises the feasibility of an accurate, real-time, and secure mobile authentication using MMOC technique even without relying on the outside resources.

### 5.4.2 Quantization

Smart cards have resource constraints that make MOC based authentication systems hard to implement:

- Data type: smart cards do not support floating point arithmetic. Only signed integers are supported on smart cards. All smart cards support 8-bit (int8) and 16-bit signed integers (int16). Int32 is also supported on some platforms.

- Communication channel: communication between the smart card and the outside world is restricted to a half-duplex serial communication through Application Protocol Data Unit (APDU) protocol transferring data to smart cards in 255-byte blocks ( 5 bytes for header and 250 bytes for data). This limits the maximum transfer rate to 115 kbps [99].
- Resources on smart cards: smart cards have restricted resources. New ARM secure core has CPU clock frequency up to 25MHz [13]. Moreover, available storage on modern smart cards is around 40kB of RAM, and around 1MB of EEPROM/Flash [129].

Considering the aforementioned restrictions on smart cards, especially the supported data types, we need to quantize data before transferring them to smart cards. Suppose that a floating point variable $r$ in range $[r_{min}, r_{max}]$ needs to be quantized to a $k$-bit signed integer (int$k$) ranging from $-2^{(k-1)}$ to $2^{(k-1)}-1$ with $2^k$ quantization levels, $Q_{levels} = 2^k$. For example, int8 is in range [-128,127] with $Q_{levels} = 256$. However, in order to save more bit space, we consider symmetric range in our quantization scheme, e.g., [-127,127] for int8. Real-valued $r$ is quantized to $r_q$ as follows [6].

$$r_q = Clip\Big\{round(\Delta r), -(\Lambda - 1), \Lambda - 1\Big\} \tag{5.2}$$

where $\Delta$ and $\Lambda$ are defined as

$$\Delta = \frac{Q_{levels} - 1}{r_{max} - r_{min}}, \qquad \Lambda = \frac{Q_{levels}}{2} \tag{5.3}$$

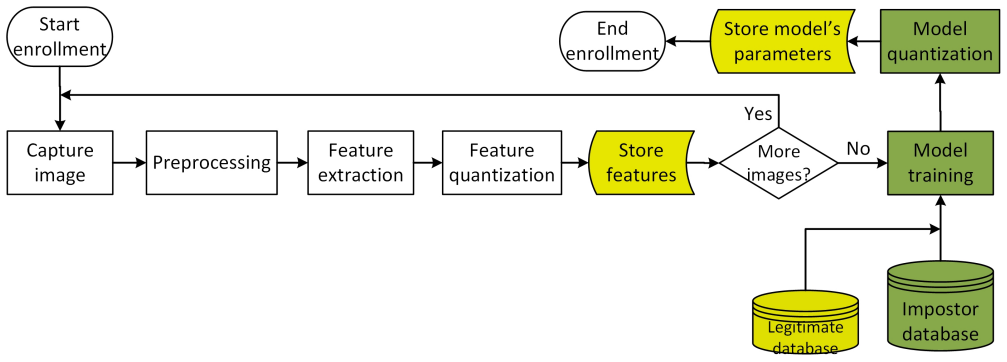
*round(x)* stochastically rounds $x$ to $\lfloor x \rfloor$. It has the desired property that the expected rounding error is zero.

$$round(x) = \begin{cases} \lfloor x \rfloor & \text{w.p } 1 - (x - \lfloor x \rfloor) \\ \lfloor x \rfloor + 1 & \text{w.p } x - \lfloor x \rfloor \end{cases} \tag{5.4}$$

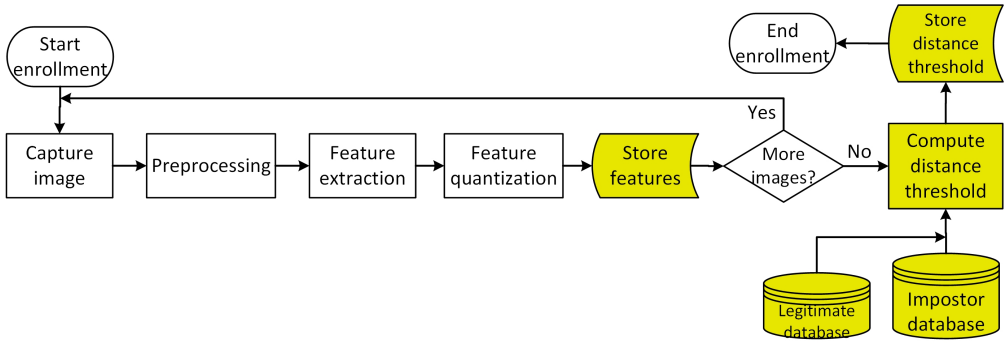
For instance, 1.3 is rounded to 1 with probability of 0.7 and is rounded to 2 with probability of 0.3.

*Clip* is a saturation function that clips unbounded values to $[-(\Lambda - 1), \Lambda - 1]$. It is defined as

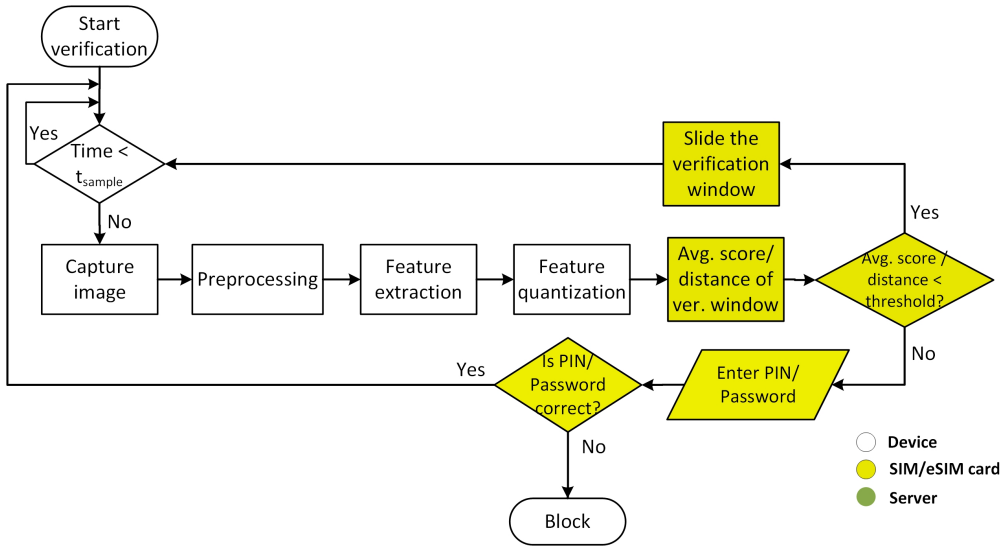
$$Clip(x, a, b) = max(a, min(b, x)) \tag{5.5}$$




(a)




(b)




(c)

Figure 5.3 Flow chart of the proposed architectures. (a) shows the enrollment process in Cloud-assisted MMOC architecture. (b) shows the enrollment process in Full MMOC architecture. A subset of a public face image dataset is used to build an impostor dataset in both architectures. (c) shows the on-card active authentication process. Every $t_{sample}$ a face image is captured and is sent to the SIM card. The average of the verification window is compared against the classification threshold, and the primary authentication method is triggered if the average goes above the threshold.

### 5.4.3 User authentication

We propose two user authentication schemes for our MMOC face authentication system.

#### Cloud-assisted MMOC authentication

In this scheme, which is a *model-based authentication* scheme, we capture the legitimate user's face images, preprocess, extract features, quantize, and store them on the card. This way, we take advantage of a secure and portable biometric storage. After required number of templates are stored on the card, templates are sent to a cloud server for training. On the server, an impostor dataset is used along with the legitimate user's templates to train a person-specific model. A subset of a public face image dataset is used to build an impostor dataset. The person-specific model's parameters are then quantized and transferred to the SIM card for user verification (see Fig. 5.3a). Considering resource limitations on smart cards, we select linear classifiers among available classifiers. For a linear classifier, the decision function is defined as

$$y(\mathbf{x}) = \mathbf{w}^{\mathrm{T}}\mathbf{x} + b \tag{5.6}$$

where $\mathbf{x}$ is an input vector (i.e., the extracted features), $\mathbf{w}$ is the weight vector, and $b$ is the bias. The input vector $\mathbf{x}$ is assigned to class $C_1$ if $y(\mathbf{x}) \geq threshold$ and to class $C_2$ otherwise [130]. In verification systems, we have legitimate user (positive) class and impostor (negative) class. In the proposed model, Eq. (5.6) (user verification) is done on the card; therefore, we need to quantize all the equation's variables (i.e., $\mathbf{x}$, $\mathbf{w}$ and $b$). We compare the performance of three well-known linear classifiers, namely Linear Support Vector Machine (L-SVM), Linear Discriminant Analysis (LDA), and Logistic Regression (LR):

- Support Vector Machine [131]: SVM separates data points belonging to two different classes with an optimal hyperplane that maximizes the margin (i.e., the distance between the nearest data points of either classes and the hyperplane). For non-linear separable data points, using kernel functions, SVM projects data points to a higher-dimension space where they are linearly separable. Regarding the smart cards resource limitations, non-linear kernel functions such as radial basis function (RBF) can not be applied to the model. Therefore, a linear SVM is used for training the model on the server and authentication on the card. The hyper-parameter $C > 0$ controls the trade-off between misclassification penalty and the margin. We use a 10-fold cross validation on the training set for tuning the $C$ parameter.

- Linear Discriminant Analysis [132]: LDA is a linear classification by using dimensionality reduction. It projects the data points to a lower dimension. In order to avoid

overlapping in one dimension, by adjusting the component of the weight vector, it selects the projection that maximizes the separation between the projected class means while minimizing the intra-class variance.

- Logistic Regression [130]: LR, a member of Generalized Linear Models (GLM), predicts the probability of certain class in binary classification; however, it can be extended to multi-class problems as well. In fact, it uses the *logistic sigmoid* function to calculate the posterior probability of class $C_1$:

$$P(C_1|\mathbf{x}) = y(\mathbf{x}) = \sigma(\mathbf{w}^{\mathrm{T}}\mathbf{x} + b) \tag{5.7}$$

  where $\sigma(.)$ is the logistic sigmoid function defined as

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{5.8}$$

In order to migrate the verification process to the SIM card, we need to quantize the model to make it implementable on the card. Most of the quantization is done off the card; however, a post-quantization process is done on the card to retain the precision of on-card model close to the real model.
Off-card quantization: we apply the following rules to the input and the model's parameters, i.e., $\mathbf{x}$, $\mathbf{w}$, and $b$ in Eq. (5.6), before sending them to the SIM card.

- Input (template) are quantized to int8 using Eq. (5.2). Using the training set, we find the minimum and maximum values to calculate $\Delta_{template}$. We train our model with the quantized templates. Therefore, the decision function of the trained model can be written as

$$y_q(\mathbf{x}_Q) \simeq \mathbf{w}^{\mathrm{T}}\Delta_{template}\mathbf{x} + b \tag{5.9}$$

  where $\mathbf{x}_Q$ is the quantized input vector and $y_q(.)$ is the intermediate quantized decision function.

- The model's weights are quantized to int8 using Eq. (5.2). Minimum and maximum values of the weight vector, obtained by training the model on the quantized templates, are used to calculate $\Delta_{weight}$.

- The model's bias is quantized using Eq. (5.2) with $\Delta_{weight}$. Since the model's bias has a larger magnitude than the weights, quantizing it to lower number of bits drops the accuracy of the system; therefore, we assign a wider range to the model's bias compared to the input and the model's weights in order to clip fewer values at lower and upper

bounds. It helps us to keep the accuracy close to the real-valued model. Moreover, it does not affect the execution time on the SIM noticeably.

On-card quantization: after applying the off-card rules, we store the model on the card and compute the decision function $y_Q(.)$. It is

$$y_Q(\mathbf{x}_Q) \simeq \Delta_{weight}\mathbf{w}^{\mathrm{T}}\Delta_{template}\mathbf{x} + \Delta_{weight}b. \tag{5.10}$$

In order to minimize the quantization error, we divide $y_Q(.)$ by $\Delta_{weight}$ on the card to return back to Eq. (5.9). Therefore, we obtain decision scores on the card close to decision scores obtained in the trained model.

**Full MMOC authentication**

In this scheme, which is a *template-based authentication* scheme, after preprocessing, feature extraction, and quantization, features are stored on the SIM card for enrollment. After required number of templates stored on the card, the learning process starts. The impostors' templates, preloaded on the card, along with the captured images of the legitimate user are used for *threshold tuning* purpose. In this process, we first compute the distance between the anchor legitimate template and other legitimate/impostor templates, then we find the best classification threshold that satisfies the given False Positive Rate (FPR). This threshold is stored on the card for verification process (see Fig. 5.3b). We use distance metrics that are implementable on SIMs with low computational overhead. We select Minkowski distance [133] of order 1, 2, $\infty$, and evaluate the verification accuracy and the execution time on the SIM card. Suppose that $\mathbf{e} \in \mathbb{R}^d$, where $d$ is the number of extracted features, is the anchor template vector and $\mathbf{v} \in \mathbb{R}^d$ is a legitimate/impostor template vector. We have the following distances:

$$L_1 = \|\mathbf{e} - \mathbf{v}\|_1 = \sum_{i=1}^{d} (\mid e_i - v_i \mid) \tag{5.11}$$

$$L_2 = \|\mathbf{e} - \mathbf{v}\|_2 = \sqrt{\sum_{i=1}^{d} (e_i - v_i)^2} \tag{5.12}$$

$$L_\infty = \max_i (\mid e_i - v_i \mid) \tag{5.13}$$

Smart cards do not support square root function; therefore, in order to calculate $L_2$ distance, we use *Newton-Raphson* method [134] to approximate the square root in Eq. (5.12). Its fast convergence makes it suitable for implementation on constrained devices such as smart

cards. Algorithm 1 shows Newton-Raphson method to compute integer root square on the card. For template-based verification, since we do not have any model's parameters, only templates are quantized and used for validation and test purpose. Therefore, we quantize templates to int8 using Eq. (5.2). We find $\Delta_{template}$ using the minimum and maximum values from the training set, as we did in the model-based verification.

**Algorithm 1:** Newton-Raphson algorithm for square root on the SIM card

**input :** $n$
**output:** square root of $n$
$x_0 \leftarrow n$;
$x_1 \leftarrow \lfloor \frac{1}{2}(x_0 + 1) \rfloor$;
**while** $x_1 < x_0$ **do**
  $x_0 \leftarrow x_1$;
  $x_1 \leftarrow \lfloor \frac{1}{2}(x_0 + \frac{n}{x_0}) \rfloor$;
**end**
$x_1$ is square root of $n$

### 5.4.4 Bit-width analysis

In order to avoid bit-width overflow in on-card computations, we should study bit-width requirements of the inputs to the verification module. In the following sections, we analyse the worst case scenario in both CA-MMOC and F-MMOC architectures. These worst cases are immensely rare and do not happen in normal conditions. For instance in real experiment, having all extracted features and the model's weights set to the lower bound or the upper bound is meaningless. However, we should consider these cases to prevent any smart card malfunctionings in real implementations.

#### The worst case in CA-MMOC

In CA-MMOC architecture, the worst case for $y_Q(\mathbf{x}_Q)$ happens when all quantized values are saturated to the lower bound or the upper bound in the vector inner product calculation; therefore, it requires $\lfloor log_2(d*(\Lambda-1)^2) \rfloor + 2$ bits, where $d$ is the vector length and $\Lambda$ is defined in Eq. (5.3). One extra bit is added for the sign bit. For 16-bit smart card implementations, we should fit this value in int16 (16-bit signed integer). Therefore, we quantize templates and the model's weights to int4, we need $\lfloor log_2(128 * 7^2) \rfloor + 2 = 14$ bits for the vector inner product. This assignment gives us a wider range for the model's bias. We do not need to clip the quantized bias, which helps us to obtain more accurate scores for binary classification. The model's bias can be in range $[-2^{14}, 2^{14}]$. For 32-bit cards, templates and model's weights

are quantized to int8, we need $\lfloor log_2(128 * 127^2) \rfloor + 2 = 22$ bits for the inner product result. Considering the bit-length of int32 (i.e., 31 bits for data), the model's bias can take a value from $[-2^{30}, 2^{30}]$.

**The worst case in F-MMOC**

The bit-width requirement in this architecture is as follows. Maximum number of bits for $L_1$ metric is $\lfloor log_2(2*d*(\Lambda-1)) \rfloor + 2$ bits, where all templates at same positions take the opposite bounds, and the maximum absolute difference is twice the upper bound value. Using int8 (8-bit signed integer) quantization, we require $\lfloor log_2(2*128*127) \rfloor + 2 = 16$ bits to store the result of $L_1$. This value nicely fits int16 and is implementable on 16- and 32-bit smart cards. For $L_2$ distance, in the worst case, we require $\lfloor log_2(d*(2*(\Lambda-1))^2) \rfloor + 2$ bits to store the result. For 16-bit smart cards, we quantize templates to int4 (4-bit signed integer), we will have $\lfloor log_2(128 * (2 * 7)^2) \rfloor + 2 = 16$ bits that nicely fits int16. 32-bit cards give us more bit-width for computation and we quantize the templates to int8. Therefore, 16 bits are required for the squared maximum absolute difference. We will need, $\lfloor log_2(128 * (2 * 127)^2) \rfloor + 2 = 24$ bits for $L_2$ distance and assign int32 to it. $L_\infty$ distance only requires $\lfloor log_2(2*(\Lambda-1)) \rfloor + 2$ bits for the maximum absolute difference. Quantizing templates to int8 requires $\lfloor log_2(2 * 127) \rfloor + 2 = 9$ bits that can be implemented on both 16- and 32-bit smart cards.

### 5.4.5 Active authentication on SIM/eSIM

An AA system continuously and unobtrusively monitors the current user to reduce the risk of system's control take-over. In the proposed AA system, the front-facing camera captures images every $t_{sample}$. After preprocessing, feature extraction, and quantization, the feature vector is sent to the SIM for verification. On the card, we define a verification window to compute the final decision score. The score/distance of the newly captured images are stored chronologically in a buffer on the SIM. The window size, $n$, defines how many image scores are fused to obtain the final score. In other words, it shows how confident the system is in making classification decision. Larger window size reduces the false negative rate and false positive rate; however, it gives more time to an impostor if an attack happens in the initial stage (until all $n$ scores are available). The window slides $k$ slots when the required number of images are present in the buffer. It defines how fast the system is in detecting impostors after the initial stage. We need to store $n-1$ previous scores to compute the final score. We apply the average score fusion rule to obtain the final decision score. Fig. 5.3c shows the flow chart of active authentication on the SIM/eSIM card.

### 5.4.6 Execution speed-up

Even simple calculations on mobile devices are tedious tasks on SIM cards. In the following, two mechanisms are employed to reduce the execution time on the card.

#### Loop optimization

The main instruction in verification phase on the card is a `for` loop to compute the two vectors dot product or the distance between them. Using a regular `for` instruction, i.e., `for(int i=0; i<d; i++)`, only one operation is calculated at a time; moreover, for each iteration the compiler checks the loop termination and index modification. However, using the *unrolled* format of the `for` instruction with *unroll factor* of $r$, the compiler can take advantage of operation pipelining and interferes less in loop execution by reducing the number of loop termination checks and index modifications to $\lfloor \frac{d}{r} \rfloor + 1$. We replace

```
for(int i=0; i<d; i++){
     y+=w[i]∘x[i];
}
```

with

```
for(int i=0; i<⌊d/r⌋ * r; i+=r){
     y+=w[i]∘x[i]+w[i+1]∘x[i+1]+....
      +w[i+r-1]∘x[i+r-1];
}
y+=w[⌊d/r⌋ * r]∘x[⌊d/r⌋ * r]+...+w[d-1]∘x[d-1];
```

where symbol $\circ$ means different mathematical operations. The last line, which is called *peeled loop*, is added when $\lfloor \frac{d}{r} \rfloor * r < d$, to $\circ$-add the remaining entries from $\lfloor \frac{d}{r} \rfloor * r$ to $d - 1$. However, this loop transformation increases the program code size and may cause instruction cache overflow.

#### Memory management

SIM cards have EEPROM/Flash and RAM for data storage. EEPROM/Flash offers more storage space compared to RAM. On the other hand, writing on EEPROM/Flash is more than 30 times slower than RAM [121]. Moreover, EEPROM/Flash has limited write cycles. Writing many times on EEPROM/Flash would damage the card's memory. Card OS does not include memory management; therefore, in order to have a better use of the card's memory

and improve the execution time, we should manage the memory manually. The model's parameters are initially stored in EEPROM/Flash for later use in verification phase. When a feature vector is transferred to the card, we store it in RAM; moreover, in order to reduce the memory access time, we fetch the model's parameters manually from EEPROM/Flash to RAM before going through the multiply-accumulate calculation (MAC). This way, we keep EEPROM/Flash healthy and decrease on-card execution time.

## 5.5 Performance evaluation

### 5.5.1 Evaluation dataset

For evaluation purpose we use MOBIO dataset [43]. The MOBIO dataset consists of video and audio data captured from 152 subjects during around two years and from 5 different countries at 6 different sites [1]. Data were collected in two phases each consists of 6 sessions where people being asked different number of questions while recording their video and audio with a NOKIA N93i mobile or a standard 2008 MacBook. First session consists of face videos captured by the laptop while the 11 remaining sessions consist of videos from mobile phone. Sample images from this dataset are shown in Fig. 5.4.

### 5.5.2 Evaluation configuration

Transferability and platform independence feature of SIM cards make the MMOC system a good candidate for cross-platform authentication. In order to show the robustness of the system to platform changes, we evaluate it in *single platform* and *cross platform* scenarios. For single platform evaluation, we use videos captured by the mobile device from one session for training and validation purpose and the remaining sessions are used for testing. For *cross platform* evaluation, videos captured by the laptop is used for training and validation while other (mobile) sessions are used for testing. "One-vs-all" method [119] is used to train a person-specific model, where images from the legitimate user is considered as "one" class and other images are seen as "all" class. Applying one-vs-all method, an imbalanced dataset is generated that will affect the overall accuracy of the biometric system. It needs threshold tuning [120]. 20% of the training set is used as validation set for model selection and threshold tuning. Using cross-validation, we set the hyperparameter $C$ to $10^{-5}$ in L-SVM. For feature extraction, FaceNet model trained on *MS-Celeb-1M* dataset [38] containing over 10 million

[1]The data was recorded at the following sites: the Brno University of Technology (BUT), Idiap Research Institute (IDIAP), University of Avignon (LIA), University of Manchester (UMAN), University of Surrey (UNIS) and University of Oulu (OULU).

Figure 5.4 Sample images from MOBIO dataset. Each row shows a specific user in different conditions.

face images of nearly 100,000 individuals is used. Active authentication on SIM is performed with $t_{sample} = 10$ seconds, $n = 1$, and the verification window slides one slot at a time, $k = 1$.

We develop an applet for biometric template storage and user verification on the SIM card. We use a SIM card running Java Card version 3.0.4 classic that supports 32-bit integer[2] with a secure core processor that is tamper resistant and robust against side channel and fault injection attacks[3] [20]. It has 1.5 MB of Flash and 53 KB of RAM. Since eSIMs also support Java Card technology, this applet can be added to a profile on eSIM [122]. We disable 32-bit integer on the card to evaluate the system on 16-bit SIM cards as well. T=0 protocol is used to communicate with the card through a contact interface.

### 5.5.3 Evaluation results

We report the performance of the proposed MMOC active authentication systems in terms of on-card execution time, Area Under Curve (AUC), Equal Error Rate (EER), and False Negative Rate (FNR) at a given False Positive Rate (FPR). EER is an error rate where FPR = FNR. Moreover, we compare the proposed systems with the most relevant work to ours conducted recently by Findling et al. [19], named D-CSLDA.

[2] https://www.samsung.com/semiconductor/security/sim-esim/S3FV9RP_SIM-ESIM/
[3] https://www.arm.com/products/silicon-ip-cpu/securcore/sc300/

**Authentication accuracy**

Table 5.1 shows the evaluation results in terms of EER in single platform and cross platform scenarios on MOBIO dataset. In single platform scenario, among model-based methods, LDA performs worse, where LR and L-SVM perform almost equally on different dataset sites. On the other hand, among template-based methods, $L_2$ and $L_1$ outperform $L_\infty$ method on all sites. In cross platform scenario, LDA's performance degrades remarkably on different sites compared to the other model-based methods. L-SVM and LR perform similar to the single platform scenario with LR performing slightly better. We see a slight increase in EER on IDIAP and UOULU sites. Laptop sessions of these two sites contain partial face images, face images far from the camera, images with multiple faces almost at the same distance, and images with no legitimate users, which cause the performance degradation. Among template-based methods, the results remain consistent as in single platform scenario with the exception on IDIAP and UOULU sites that we explained the reason above. D-CSLDA shows the worst EER compared to CA-MMOC and F-MMOC. D-CSLDA uses 2D discrete wavelet transformation (2D-DWT) with a Daubechies Least-Asymmetric that does not perform well under varying face poses and illuminations; moreover, the authors converted the image quality to gray-scale, reduced the image size to $32 \times 32$, and decreased the feature vector size from 1365 features to 75 features in order to control the on-card execution time. Therefore, we see a significant increase in EER for D-CSLDA. This situation is worse in cross-platform scenario, where we see up to 31% of EER in the system while the worst EER in our systems belongs to $L_\infty$ with EER of 6.8%. Although the authors proposed to perform 32 comparisons between the enrollment set and the probe to increase the accuracy; however, we evaluated the scenario of one comparison (as used in our systems) in order to have a fair comparison between the systems.

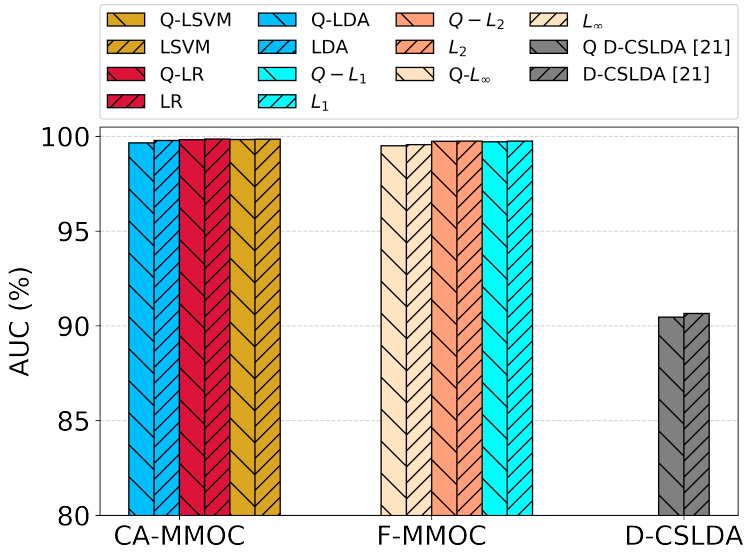
Fig. 5.5 illustrates AUC of each system under different platform scenarios on the whole dataset. In single platform scenario, as can be seen, CA-MMOC and F-MMOC perform similarly and reach the highest AUC of 99.83% (Q-LSVM), while D-CSLDA reaches AUC of 90.46%. In cross platform scenario, the image quality variation has remarkably affected the system performance and we see an AUC decrease in all systems, where D-SCLDA shows the worst performance with 10% reduction in AUC. In general, Minkowski distance metrics and LDA (or CSLDA) are more vulnerable to image quality variations in cross platform scenario. Moreover, reducing image size, quality, and the number of features affect system performance harder in cross platform scenario.

These results show that MMOC authentication can be considered as a secure transferable verification system in which the model can be trained on one device, and the stored verifi-

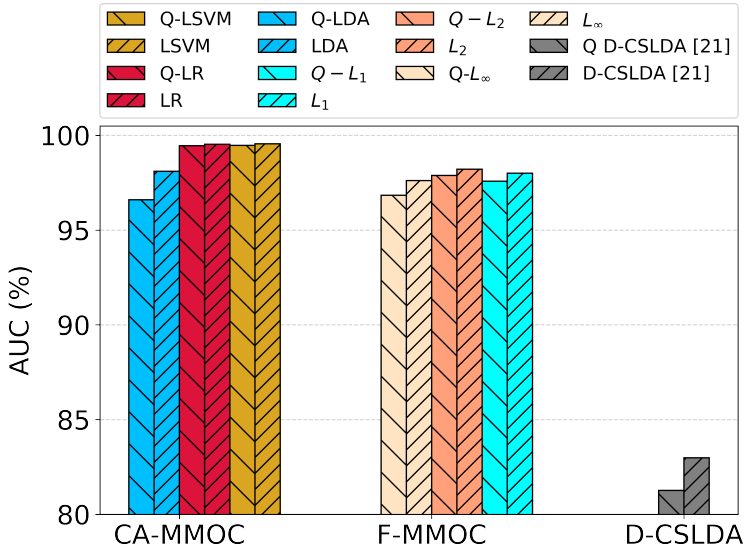
Table 5.1 Evaluation results in single platform and cross platform scenarios with standard deviation in parentheses and real values in square brackets.

| Site | MMOC architecture | Alg. | Mean EER (std) | |
|---|---|---|---|---|
| | | | Single Platform | Cross Platform |
| BUT | CA-MMOC | L-SVM | **0.1 (0.4)** [0.1 (0.3)] | **0.2 (0.4)** [0.2 (0.4)] |
| | | LDA | 0.3 (0.4) [0.3 (0.4)] | 3.5 (3.1) [2.9 (2.6)] |
| | | LR | **0.1 (0.4)** [0.1 (0.4)] | **0.2 (0.4)** [0.2 (0.4)] |
| | F-MMOC | $L_1$ | 0.1 (0.2) [0.1 (0.1)] | 0.1 (0.2) [0.1 (0.1)] |
| | | $L_2$ | **0.1 (0.1)** [0.1 (0.0)] | **0.1 (0.1)** [0.1 (0.1)] |
| | | $L_\infty$ | 0.8 (1.6) [0.7 (1.4)] | 0.9 (1.0) [0.7 (0.7)] |
| | D-CSLDA [19] | CSLDA | 13.5 (4.2) [13.1 (4.3)] | 21.9 (5.2) [20.2 (6.6)] |
| IDIAP | CA-MMOC | L-SVM | **0.0 (0.0)** [0.0 (0.0)] | 0.5 (0.8) [0.3 (0.5)] |
| | | LDA | 0.2 (0.2) [0.1 (0.1)] | 11.5 (12.3) [3.6 (4.1)] |
| | | LR | **0.0 (0.0)** [0.0 (0.0)] | **0.3 (0.4)** [0.2 (0.3)] |
| | F-MMOC | $L_1$ | **0.1 (0.1)** [0.1 (0.1)] | 2.6 (9.1) [2.5 (8.8)] |
| | | $L_2$ | **0.1 (0.1)** [0.1 (0.1)] | **2.4 (8.9)** [2.1 (8.1)] |
| | | $L_\infty$ | 0.2 (0.1) [0.2 (0.1)] | 2.8 (2.1) [2.4 (1.1)] |
| | D-CSLDA [19] | CSLDA | 12.8 (6.2) [12.3 (5.1)] | 28.1 (10.2) [25.6 (9.1)] |
| LIA | CA-MMOC | L-SVM | 1.4 (4.2) [1.4 (3.4)] | 1.5 (3.2) [1.5 (3.4)] |
| | | LDA | 1.6 (3.0) [1.4 (3.0)] | 2.1 (3.2) [1.8 (3.8)] |
| | | LR | **1.4 (3.8)** [1.4 (4.0)] | **1.5 (3.0)** [1.4 (3.4)] |
| | F-MMOC | $L_1$ | 1.2 (2.5) [1.0 (2.4)] | 1.3 (2.3) [1.2 (2.2)] |
| | | $L_2$ | **1.1 (3.0)** [1.0 (2.2)] | **1.2 (2.9)** [1.1 (2.3)] |
| | | $L_\infty$ | 1.3 (2.6) [1.0 (2.6)] | 1.4 (2.6) [1.2 (3.6)] |
| | D-CSLDA [19] | CSLDA | 19.1 (8.2) [18.7 (8.5)] | 24.7 (8.7) [23.1 (10.2)] |
| UMAN | CA-MMOC | L-SVM | **0.1 (0.1)** [0.0 (0.0)] | **0.1 (0.2)** [0.1 (0.2)] |
| | | LDA | 0.4 (0.4) [0.0 (0.1)] | 3.0 (2.7) [2.6 (2.3)] |
| | | LR | **0.1 (0.1)** [0.0 (0.1)] | **0.1 (0.2)** [0.1 (0.1)] |
| | F-MMOC | $L_1$ | 0.1 (0.2) [0.1 (0.1)] | **0.1 (0.1)** [0.1 (0.1)] |
| | | $L_2$ | **0.1 (0.1)** [0.1 (0.0)] | **0.1 (0.1)** [0.1 (0.2)] |
| | | $L_\infty$ | 0.7 (1.3) [0.7 (1.2)] | 1.1 (1.1) [0.7 (1.2)] |
| | D-CSLDA [19] | CSLDA | 16.1 (4.6) [15.6 (5.9)] | 23.1 (10.2) [21.1 (10.8)] |
| UNIS | CA-MMOC | L-SVM | **0.1 (0.2)** [0.1 (0.2)] | **0.1 (0.2)** [0.1 (0.1)] |
| | | LDA | 0.4 (0.4) [0.1 (0.2)] | 0.3 (0.4) [0.1 (0.3)] |
| | | LR | **0.1 (0.2)** [0.1 (0.2)] | **0.1 (0.2)** [0.1 (0.2)] |
| | F-MMOC | $L_1$ | 0.3 (0.3) [0.2 (0.3)] | 0.5 (0.8) [0.2 (0.9)] |
| | | $L_2$ | **0.2 (0.2)** [0.1 (0.2)] | **0.4 (0.7)** [0.2 (0.5)] |
| | | $L_\infty$ | 0.7 (1.1) [0.5 (0.8)] | 1.0 (1.1) [0.5 (0.6)] |
| | D-CSLDA [19] | CSLDA | 15.1 (7.1) [14.8 (7.1)] | 21.0 (8.4) [20.5 (9.6)] |
| UOULU | CA-MMOC | L-SVM | **0.1 (0.1)** [0.0 (0.0)] | 0.8 (0.6) [0.5 (0.7)] |
| | | LDA | 0.6 (0.4) [0.3 (0.4)] | 9.3 (6.5) [8.5 (7.4)] |
| | | LR | **0.1 (0.1)** [0.1 (0.1)] | **0.5 (0.9)** [0.3 (0.3)] |
| | F-MMOC | $L_1$ | 0.2 (0.1) [0.2 (0.1)] | 7.3 (11.1) [6.3 (10.1)] |
| | | $L_2$ | **0.1 (0.1)** [0.1 (0.1)] | **6.8 (13.1)** [5.5 (12.2)] |
| | | $L_\infty$ | 0.5 (0.6) [0.4 (0.6)] | 7.1 (10.8) [6.2 (10.6)] |
| | D-CSLDA [19] | CSLDA | 22.5 (9.2) [22.3 (9.3)] | 31.1 (11.5) [29.3 (13.4)] |

cation model on the SIM can be used for user verification on different devices. In the case of eSIM, the card is soldered into the device; however, RSP enables us to transfer the eSIM profile to a new device and use the verification module on that device [135] (see section 5.2).




(a)




(b)

Figure 5.5 System's AUC in different platform scenarios. (a) is the single platform scenario, and (b) is the cross-platform scenario.

Table 5.2 depicts the effect of quantization bit-width on the accuracy of the system in terms of EER. Q-LSVM and Q-$L_2$ are selected for comparison of model-based and template-based methods. The result shows that the authentication accuracy drops slightly by reducing num-

Table 5.2 Effect of quantization bit-width on the system's performance.

| **Quantization bits** | **Quantization level** | **Integer range** | **Method** | **Mean EER (std)** |
|---|---|---|---|---|
| 2-bits | 3 | [-1,1] | Q-LSVM | 1.5% (1.6) |
| | | | Q-$L_2$ | 0.8% (1.1) |
| 3-bits | 7 | [-3,3] | Q-LSVM | 0.6% (0.8) |
| | | | Q-$L_2$ | 0.5% (0.9) |
| 4-bits | 15 | [-7,7] | Q-LSVM | 0.5% (0.7) |
| | | | Q-$L_2$ | 0.4% (0.9) |
| 5-bits | 31 | [-15,15] | Q-LSVM | 0.5% (0.8) |
| | | | Q-$L_2$ | 0.4% (0.8) |
| 6-bits | 63 | [-31,31] | Q-LSVM | 0.4% (0.7) |
| | | | Q-$L_2$ | 0.3% (0.7) |
| 7-bits | 127 | [-63,63] | Q-LSVM | 0.4% (0.7) |
| | | | Q-$L_2$ | 0.3% (0.6) |
| 8-bits | 255 | [-127,127] | Q-LSVM | 0.3% (0.6) |
| | | | Q-$L_2$ | 0.3% (0.5) |

ber of quantization level. Q-$L_2$ performs better than Q-LSVM specially in 2-bit quantization, where its mean EER is about 0.8 %, while Q-LSVM's EER increases to 1.5%. This promising result shows that we can greatly reduce the systems' memory footprint up to 75% on the SIM card, and in total, up to 93.75% of the original implementation (floating point arithmetic) with a small reduction in the systems' performance. Storing 100 templates in floating point format requires 51.2 KB, in int8 format needs 12.8 KB of memory; however, applying 2-bit quantization, it requires 3.2 KB while keeping the system's EER less than 5%, which is the acceptable EER for a reliable system.

Fig. 5.6 shows FNR @ FPR $< 1\%$. Using the validation set, we find the best threshold for a given FPR, then it is used to obtain the corresponding FNR on the test set. For comparison, we select L-SVM and $L_2$ that showed the best results in single platform scenario on the dataset (see table 5.1). Although we see a larger quantization error in L-SVM than $L_2$, L-SVM (model-based verification) performs better than $L_2$ (template-based verification) at different FPR values. The reason for this gap is that in $L_2$ we only have templates for quantization, while in L-SVM we also need to quantize the model's parameters. The effect of this quntization is more sensible for FPR$< 0.1\%$, where more precision is needed to satisfy these strict FPRs. However, for FPR$> 1\%$ the quantization error is less than 0.2%.

F-MMOC enrollment includes threshold tuning. Therefore, considering memory and processing limitations of SIM cards, the minimum number of templates to load on the card that satisfies a given FPR should be determined in order to have a reasonable response time for

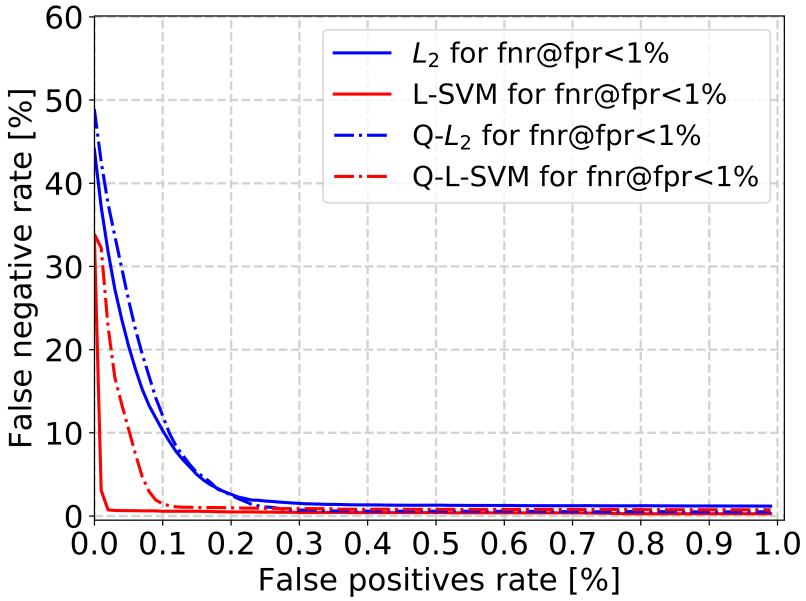



Figure 5.6 Fnr @ fpr $< 1\%$ for MOBIO dataset. L-SVM and $L_2$ is used for CA-MMOC and F-MMOC, respectively.

the enrollment. Fig. 5.7 shows the effect of number of samples on the system's FNR. FPR is set to 1% and Q-$L_2$ metric is used to compute the coresponding FNR for different number of samples on different MOBIO sites. As the results show, increasing the number of samples reduces the system's FNR. On all sites, FNR stabilizes at around 100 samples and adding more samples does not decrease the FNR significantly (around 0.1%). UNIS and LIA sites show the worst FNR while IDIAP has the best FNR @FPR= 1%.

**Execution time on SIM**

Fig. 5.8 shows the verification time on 16- and 32-bit SIM cards for different verification methods with and without speed-up technique. This duration contains feature transmission, score (distance) computation, comparison with the classification threshold and sending out the decision. Since linear models use a same concept to calculate the score, we depict "Score" in the figure for all linear classifiers used here. As can be seen in the figure, for distance-based verification methods, since $L_\infty$ distance needs less computation, it has the best execution time compared to the other methods. On 32-bit cards, it achieves 121.3 ms (std=10.6 ms) and 118.6 ms (std=10.7 ms) on 16-bit cards. In contrast, $L_2$ requires more computation and it has the worst execution time which is 138.6 ms (std=17.4 ms) and 134.3 (std=15.1 ms) on 32- and 16-bit cards, respectively. On the other hand, for score-based methods, the vector dot product takes 107.4 ms (std=12.2 ms) on 32-bit cards and 105.3 ms (std=10.7 ms) on

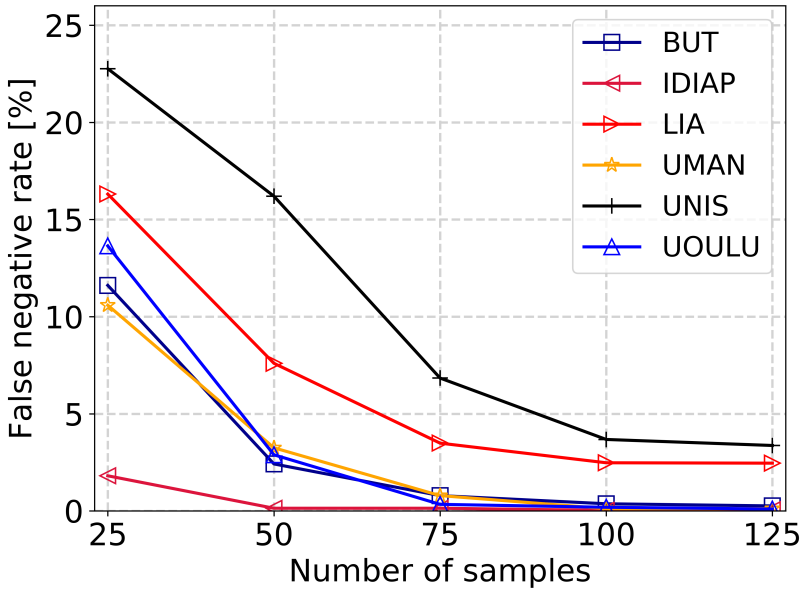



Figure 5.7 Fnr @ fpr = 1% for different MOBIO's sites and different number of samples with Q-$L_2$ metric.

16-bit cards that is the best execution time among the two architectures. Applying the loop unrolling technique does not decrease the execution time for distance-based methods; however, in score-based verification we see 15% decrease on the execution time resulting 92.7 ms (10.4 ms) and 95.2 ms (10.2 ms) execution time on 16- and 32-bit SIM cards, respectively. The reason for this SIM's behavior is that using loop unrolling, Java Card compiler takes advantage of pipelining which is really useful in computing vector dot product. The figure also illustrates that D-CSLDA [19] has the least execution time compared to the proposed systems. A Face template in D-CSLDA has 75 features which is fewer than the template size in our systems (128 features). Therefore, template transmission and score computation is faster on the card. The results also show that the execution is slightly faster (about 2%) on 16-bit cards than on 32-bit cards.

Fig. 5.9 shows the amount of time required for user enrollment in F-MMOC for different number of samples. The enrollment phase consists of transferring samples to the card, computing distance between the anchor template and positive (legitimate) and negative (impostor) templates, and finding the best threshold that satisfies the given FPR (here, we set FPR to 1%). Of total number of loaded samples for enrollment on the card, 20% of them are positive samples, while others are negative samples. We apply the loop unrolling technique to improve the execution time on the card. Since the original implementation takes a considerable time on the card, it is not depicted in the figure. For example, for 100 samples,

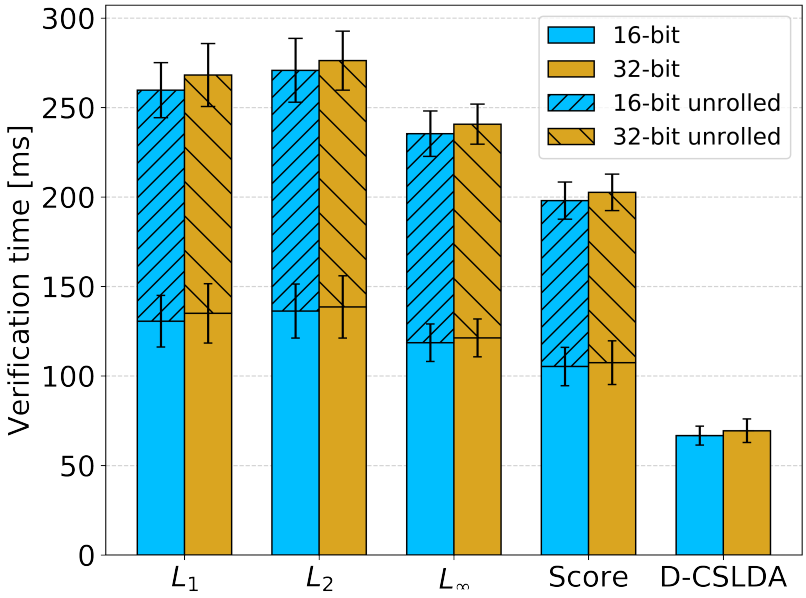



Figure 5.8 Verification time on 16- and 32-bit SIM cards.

the original implementation takes roughly 133.9 s (std = 2.7 s) to enroll a user and compute the best threshold, which emphasises the need of a technique to reduce the execution time and improve the overall system's performance. We apply loop unrolling and evaluate it on different number of samples with different unroll factor, $r$. We select $r$ in the form of $2^n$ in order to eliminate the peeled loop and decrease the duration on the card (see section 5.4.6). As can be seen in the figure, increasing the unroll factor reduces the enrollment time on the card. For 100 samples, $r$=64 decreases execution time 35% compared to unroll factor 16 and 15% compared to $r$=32. However, considering code buffer limitation on the card, we cannot apply large unroll factors for heavy computations on the card. The results show that for $r$=64, the code buffer overflows at sample 58 and at sample 86 for $r$=32, whereas we can use 125 samples for on-card threshold tuning using $r$=16. Considering the discussion in Fig. 5.7, we saw that FNR stabilizes at around 100 samples; therefore, Fig. 5.9 says that we can train our template-based authentication system on the card with 100 samples and $r$=16 in about 15.8 seconds (std = 0.9 s). We gain $\frac{133.9}{15.8} \approx 8.5\times$ speed-up over the original implementation.

The user enrollment phase happens only once and its duration will not impact the real-time active authentication. The user verification is done below 0.15 second on the card (see Fig. 5.8). During the enrollment phase, the primary on-card authentication method such as fingerprint, password, or PIN is activated to verify the legitimate user.

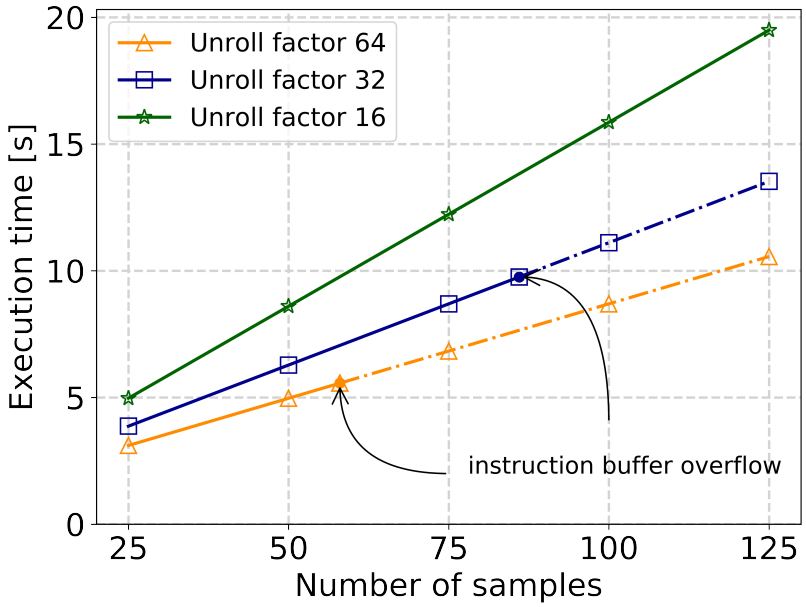



Figure 5.9 Enrollment time on the SIM card

## 5.6 Platform implementation

We implement the CA-MMOC active authentication on a real Android device in order to evaluate its performance overhead. This authentication system consists of three modules that we discuss in the following sections.

### 5.6.1 Cloud server

A python code running on Amazon Web Services (AWS) [136] is developed for model training purpose. Moreover, in order to train a "one-vs-all" classifier, we use samples from a public face dataset. For this implementation, we use sample images from MOBIO dataset to train a L-SVM classifier for the legitimate user. No model parameters or legitimate user's face samples are stored on the server. As soon as the model is trained, the parameters are sent back to the mobile application through an established secure channel. Design of the Authentication and Key Agreement (AKA) protocol is out of the scope of the current work. Briefly, it works as follows. First, the mobile application and the server authenticate each other using a proposed protocol. Then, if both parties are authentic, using Elliptic Curve Diffie-Hellman (ECDH) protocol [137], a shared secret is generated that later is used to exchange data (e.g., biometric templates and the model's parameters) securely using Advanced Encryption Standard (AES) on both sides.

### 5.6.2 Android application

The Android application is only used for computationally heavy tasks that currently are not feasible to be executed on SIM cards, such as image preprocessing and feature extraction. The application has two phases:

- Enrollment: In the enrollment phase, the application captures 30 images of the legitimate user. For each captured image, a `Thread` is created to handle image preprocessing, and feature extraction, quantization, and template storage on the SIM card. In order to have a high accuracy verification, FaceNet needs cropped face images of size $160 \times 160$ with face landmarks in the center. We used Multitask Cascaded Convolutional Networks (MTCNN) [138] for preprocessing. We load the model in `MappedByteBuffer` to initialize `Interpreter` class [139] and use `run()` method of TensorFlow Lite [140] for inference, to extract a 128-dimensional float vector from each image. Each vector is quantized using Eq. (5.2), converted to APDU format, and sent to the SIM for secure storage and authentication phase. These templates are fetched from the SIM and sent to the cloud application for model training through the established secure channel.
- Authentication: In this phase, one image is captured every 10 seconds, preprocessed, feature are extracted, and the quantized features are sent to the SIM card for verification. Fig. 5.10 shows the screenshots of the Android application.

### 5.6.3 SIM/eSIM applet

A SIM applet is developed to store the quantized 128-dimensional feature vector of each image, and verify the image sent by the application in authentication phase. It has two methods: `enroll()` method in CA-MMOC is only used for template storage; however, in F-MMOC architecture is used to store the legitimate user's templates and to find the classification threshold. `activeAuthenticate()` method consists of obtaining the decision score by computing the inner product of the model's weights and the probe's vector plus the model's bias in CA-MMOC. It also computes the average score of the verification window, compares it with the threshold, slides the window, and sends the verification decision to the Android application.

For communication between the application and the applet, first, we need to open a communication channel between the Android application and the SIM's applet. `TelephonyManager` class [141] of Android provides a method called `iccOpenLogicalChannel()` for opening a channel with a SIM applet. `iccTransmitApduLogicalChannel()` method is used to transmit APDU data to the applet. To use this class the application should have $MODIFY\_PHONE\_STATE$

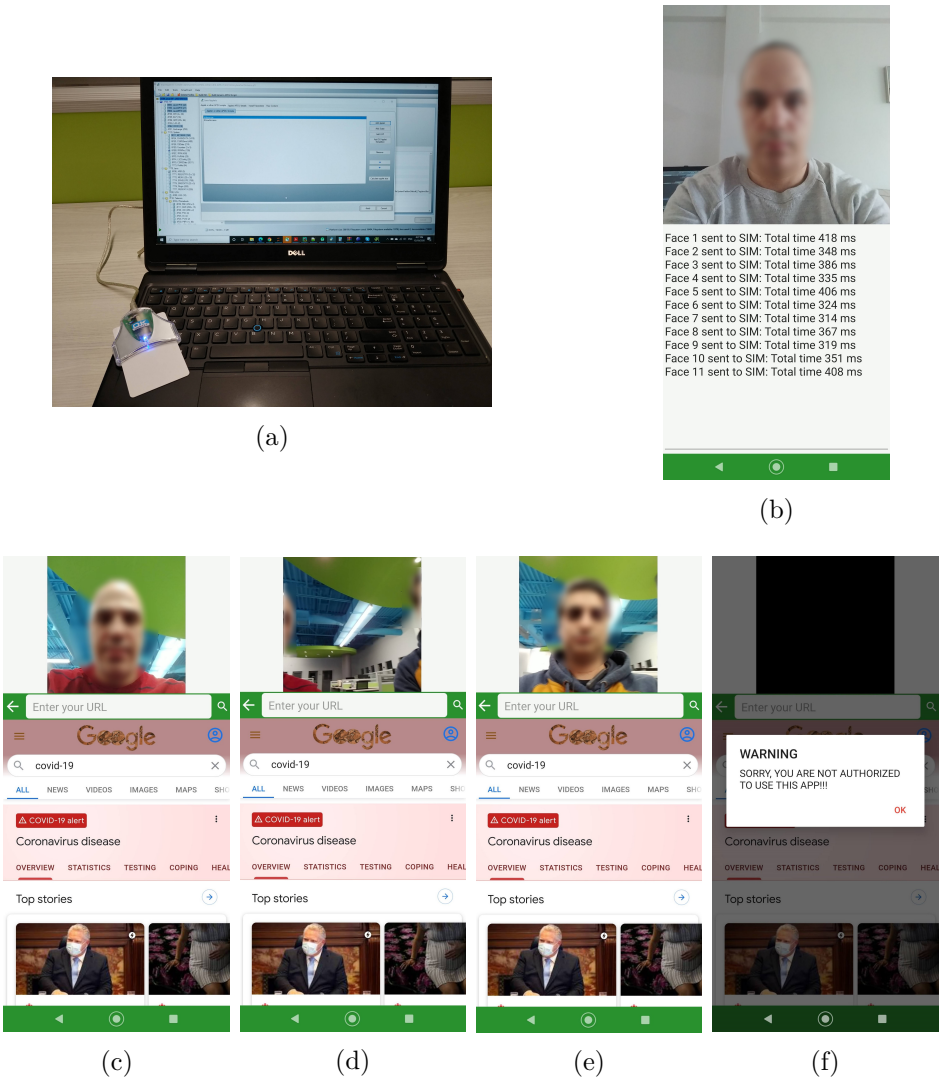
permission that is not granted to third party applications, or the application should have the carrier privileges which means that `hasCarrierPrivileges()` call should return `true`. In order to obtain the carrier privilages, an Access Rule Applet (ARA) with application identifier (AID) A00000015141434C00 is developed to grant carrier privileges to our Android application [142]. ARA stores signatures of applications allowed to access a specific applet on the card. Android platform checks this applet and grants permission to the applications signed by the certificates declared in ARA. We can manage ARA rules using over-the-air (OTA) platform for SIMs and RSP platform for eSIMs, in case malicious activities are reported.

### 5.6.4 Platform overhead

We use Android Profiler[4] to capture information about memory, cpu and battery usage, while running the application. Our testbed is a Samsung Galaxy A20 with a 6.4-inch display, an octa-core cpu (2x1.6 GHz + 6x1.35 GHz), 3 GB of memory, and 32 GB of internal storage running Android 9 pie. For devices running Android 8 and above, due to enhanced Android security, third party applications can not gather information about other applications' resource usage; therefore, we use the built-in tool in Android Studio to monitor cpu and memory usage. Battery and storage usage information are collected using a built-in Android application[5]. We let the legitimate user to work with the device while the AA system captures his face images automatically every 10 seconds. The user interacts with the phone for about 30 minutes (session duration) before he logs off the phone. For comparison, we also develop a simple version of the application without using a SIM card. However, in order to provide a basic protection of the model's parameters stored on the device, these data are encrypted using AES with the shared secret obtained in the AKA phase. When a new face image is captured, the AA system reads the encrypted model's parameters, decrypt them, computes the score and the final authentication result. Table 5.3 shows the measured overhead of the CA-MMOC active authentication and the simple secure active authentication with AES encryption. As can be seen in the figure, in the secure AA with AES, since the application stores the intermediate scores in the memory we see 10% increase in the app memory usage. Moreover, since the model's parameters are stored encrypted, more cpu processing is needed for decryption, and we see about 2% increase in cpu usage compared to the CA-MMOC AA system. CA-MMOC AA system needs only 132 bytes (128 bytes for the weight vector and 4 bytes for the bias) to store the model's parameters in SIM's Flash; however, the secure AA system with AES uses up about 0.2 MB of storage for the encrypted model's parameters. Fi-

[4] `https://developer.android.com/studio/profile/android-profiler`

[5] `https://play.google.com/store/apps/details?id=com.samsung.android.lool&hl=en_CA`https://play.google.com/store/apps/device-care




(a) (b) (c) (d) (e) (f)

Figure 5.10 Screenshots of the Android application. (a) shows the process of adding the applets to an MNO's profile on the SIM card. (b) is the enrollment phase. Total time consists of face detection time, image preprocessing time, feature extraction time, and transmission time to the SIM card. (c)-(f) show an active authentication scenario. In (c), the user is working with an in-app web browser while the front-facing camera captures face images every 10 seconds for a session of 30 minutes. (d) shows an impostor taking over the phone. In (e), the authentication module on the SIM detects an illegitimate face image. Access to the application is denied in (f).

nally, MMOC system consumes less battery power compared to the encrypted authentication system, where they use 1% and 1.1% of the battery in 30 minutes, respectively.

Table 5.3 Resource consumption of CA-MMOC AA and AA with AES.

| **AA system** | **Memory (MB)** | **Storage** | **CPU** | **Battery** |
|---|---|---|---|---|
| MMOC AA | 33.8 | 0 B | 12% | 1% |
| AA with AES | 37.2 | 195.5 KB | 14% | 1.1% |

## 5.7 Comparison of the two systems

Although the two systems take advantage of hardware security characteristics of SIM/eSIM cards to increase the privacy of biometric templates and security of the verification phase, they have their own pros and cons that are discussed in this section.

In CA-MMOC, the machine learning model is trained on a cloud server. Although the enrollment phase is fast in this architecture, we should consider the possible information leakage in communication channel if no secure channel is established, or in the cloud service if no security attack countermeasure is employed.

On the other hand, in F-MMOC, no information leaves the SIM/eSIM card in enrollment phase, which promises a higher secure solution compared to CA-MMOC architecture. However, the enrollment phase that includes threshold tuning is multiple of magnitude slower than the enrollment in CA-MMOC. Moreover, if we desire a highly secure system that satisfies a lower FPR, this process takes more time on the SIM, which requires a more sophisticated on-card optimization technique to reduce the execution time.

Summarizing, These architectures show a trade-off between real-time user enrollment (CA-MMOC architecture) and privacy-preserving authentication system (F-MMOC architecture). To enhance the privacy of the system, we should consider the second architecture; however, it will skew the system towards a non real-time enrollment phase, which may affect the user experience in the system. Likewise, in order to improve the enrollment time, the first architecture is a better choice; however, it decreases the privacy of the system. We must make sure a communication channel between the device and the cloud service exists, and trust the communication channel and the cloud service itself as well. Moreover, we should not forget the deployment expenses of the first architecture. Therefore, since the accuracy of F-MMOC is comarable to CA-MMOC, and considering the higher privacy of this architecture, F-MMOC is a more reliable system for secure active authentication on smartphones. We can apply several methods to mitigate the effect of enrollment time on the system: 1) SIM card

is an independent module, we can hide this phase in the system's initial configuration steps, where the user configures the device while the enrollment is in progress in the background on the SIM. 2) Improve the on-card optimization technique to reduce the enrollment time. 3) Use the baseline threshold to eliminate threshold tuning on the card (it affects system's security).

## 5.8 Conclusion

In this paper, a Mobile Match-on-Card (MMOC) face-based active authentication system for smartphones was presented. We proposed two architectures for the MMOC system. In the model-based architecture, we took advantage of cloud resources to train a machine learning model and migrate the model to the SIM/eSIM card for a real-time on-card user verification. The second architecture only relies on SIM/eSIM card resources for enrollment and verification. Using a good quantization scheme, we showed that the accuracy of the proposed MMOC system is comparable to the original model, while we can reduce its size up to 93.75%. On-card user verification takes less than 130 ms, which does not impact the active authentication process even with high verification frequency. The second architecture is more secure than the first one since no biometric templates leaves the card. The burden of this architecture lays on the enrollment phase, where it takes about 15 seconds to find the decision threshold on the card using 100 face templates. However, since the enrollment mostly happens at the system initialization phase, it does not affect the real-time on-card verification. We also showed the robustness of the model-based and template-based methods in cross-platform scenario, where the model-based method performed slightly better. This shows the potential of MMOC technique for cross-platform authentication. Finally, we implemented our active authentication system on a real device and showed the platform overhead reduction compared to a simple encrypted-biometric method. In the future, we aim to study techniques to decrease on-card execution time in order to enroll users in near real-time. Moreover, the fusion of face-based active authentication with other biometric schemes on the card is of our interests.

# CHAPTER 6 ARTICLE 3: A GENERIC MODEL FOR PRIVACY-PRESERVING AUTHENTICATION ON SMARTPHONES

Sepehr Keykhaie and Samuel Pierre

Mobile Computing and Network Research Laboratory (LARIM)
Department of Computer and Software Engineering, Ecole Polytechnique de Montréal,
Montreal, QC H3T 1J4, Canada
E-mail: sepehr.keykhaie@polymtl.ca; samuel.pierre@polymtl.ca.

Status: accepted for publication in Proceedings of 15th Annual IEEE International Systems Conference (IEEE SYSCON 2021), November 2020.

**Abstract**

With the increasing use of biometrics for user authentication especially on mobile devices, its privacy and resource requirements are becoming big challenges to consider. In this paper, we propose a generic model for privacy-preserving yet accurate authentication on smartphones using the mobile matching on card (MMOC) technique and transfer learning. MMOC technique takes advantage of SIM cards as a secure element (SE) on smartphones to increase the security and privacy of user verification with low performance overhead. In order to improve the performance accuracy of the system, we use transfer learning and fine-tune a network suitable for implementation on off-the-shelf SIM cards available on smartphones. The classification sub-network is migrated to the SIM card for a lightweight and secure user verification. However, the implementation of classification sub-network on constrained resource smart cards with high accuracy and efficiency is a challenging task. We propose log quantization scheme and an on-card optimization architecture to speed-up the forward pass of the sub-network and retain the system's accuracy close to the original model with low memory footprint and real-time verification response. Using a public mobile face dataset, we evaluate our privacy-preserving verification system. Our results show that the proposed system achieves Equal Error Rate (EER) of 0.4%-2% in real-time, with response time of 1.5 seconds.

**Keywords :** Authentication, face biometric, transfer learning, smart cards, privacy

## 6.1 Introduction

Biometrics are increasingly used for user authentication especially on smartphones. They use physiological (such as fingerprint, face, iris, etc.) or behavioral (such as gait, signature, touchscreen strokes, etc.) traits to authenticate users in a single point entry authentication or a continuous authentication (CA). In a single point entry authentication, users are authenticated only at the beginning of a session (from the login to the logout). However, in a CA system, users are continuously and transparently monitored during a session. Biometric templates are unique and long-lasting characteristics of their owners, and unlike traditional authentication methods such as PIN/password or graphical patterns, cannot be changed if compromised. Therefore, security and privacy of them is crucial to be a successful game changer in mobile authentication systems. Cancelable bioemtrics [84] or homomorphic encryption [12] are mostly proposed to store biometric templates securely on mobile devices. In the first approach, the verification accuracy would degrade due to template transformation. In the latter method, computing the matching score on encrypted templates using homomorphic encryption affects the real-time response of the authentication system owing to the heavy computation on ciphertext space.

Subscriber Identity Module (SIM) cards available on almost all smartphones can be used as a secure element (SE) on mobile devices to store biometric templates securely and verify users in an isolated secure environment. SIM cards can be used in two ways to increase the security and privacy of mobile authentication systems. In the first approach, which we call it mobile template on card (MTOC), only biometric templates are stored on the SIM card while the matching (verification) process is done outside the SIM card. The communication with the outside world can increase the probability of eavesdropping and the leakage of biometric information. In the latter approach which is called mobile match on card (MMOC), in addition to the secure storage of biometric templates on the SIM card, matching is also done on the card which enhances the security and privacy of the authentication system. However, the resource limitations of SIM cards such as a low performance processor, low data bandwidth or low memory capacity, makes the implementation of this approach a challenging task.

In this paper, we propose a generic secure and privacy-preserving authentication system for mobile users using MMOC technique, that can be used for different biometrics such as face, fingerprint, voice, etc. In order to improve the performance accuracy of the system, we use transfer learning, modify the network architecture, and fine-tune the model for on-card user verification. We reduce the dimensionality of the feature extraction network to extract less but discriminative features for the on-card classification sub-network. Moreover, we apply a

quantization scheme to the model's internals and a log quantization to the model's input to convert multiply-accumulate operations to a faster bit-shift-accumulate operations along with an on-card optimization architecture to speed-up forward pass while keeping the performance accuracy high on the smart card.

This paper makes the following contributions:

- A generic privacy-preserving authentication system using transfer learning is proposed that can use different biometrics for authentication.
- We implement classification sub-network of a fine-tuned model on SIM cards in order to have a privacy-preserving authentication system with high performance accuracy.
- We propose log quantization scheme to speed-up *forward pass* of Deep Neural Network (DNN) on the card.
- We design an on-card optimization architecture for real-time verification that 44.3× improves the execution time.

The remainder of the paper is orgonized as follows. In Section 6.2, we review several related works. Section 6.3 describes the proposed MMOC privacy-preserving authentication system. Section 6.4 shows our experimental results. Finally, we conclude the paper in section 6.5.

## 6.2 Related Work

Most MOC-based authentication studies have used fingerprint in their works [93,94,110,143]. Other biometrics such as iris, palm-print, face or behavioral traits are rarely considered in MOC systems. Lee and Byun [98] proposed an on-card face authentication system. They extracted the most discriminating features of a face using Genetic Algorithm (GA), followed by a Support Vector Machine (SVM) for user verification on the smart card. Their results show a performance improvement even with fewer face features.

Li in his PhD thesis [95], Linear Discriminant Analysis (LDA) method for an on-card face recognition system. Moreover, he investigated the effect of resource limitations such as memory capacity in the performance and efficiency of the system.

Czyz et al. [97] used Fisherfaces for face verification on smart cards. They showed the trade-off between the performance and the verification model's size on the smart card. With images size down to 256 pixels and the model' size (i.e., dimension of LDA subspace) of 25, the error rate stay equal. However, more reduction in image size and the model size increase the error rate noticeably.

Bourlai et al. [99] employed a client specific linear discriminant analysis (CS-LDA) technique for face verification on smart cards. After image normalization and feature extraction, the distance between the probe image and the user's template is computed on the smart card and thresholded to verify whether the probe belongs to the legitimate user or an impostor. They also optimized the verification system by reducing the spatial and grey-scale resolution of images.

Findling et al. [19] created an off-line model for gait recognition on smart cards. The obtained model is simplified and stored on the smart card. For verification, an absolute distance between the gait probe and the enrolled template is computed and the dot-product with the model's weights are compared to the model's bias to make the authentication decision. They evaluated their method on 16- and 32-bit smart cards, and showed that the system achieves 11.4% EER for gait authentication with transmission and computation duration on smart cards in the range of 2 seconds.

Keykhaie and Pierre [6] proposed a secure active authentication system for touchscreen authentication using match-on-card technique. A quantized deep neural network (DNN), trained on a cloud server, is migrated to the smart card for accurate continuous user authentication. In order to reduce the execution time on the smart card, an optimization technique is employed. They results showed EER of 2.6% for 15 strokes.

## 6.3 System Description

Considering the security features of smart cards, MMOC technique shows a great potential for a secure and privacy-preserving authentication system. However, resource constraints of smart cards affect the performance and real-time response of the MMOC system. Therefore, in order to address these challenges, we propose to off-load the heavy computation parts of the verification system to the mobile device and use the smart card only for secure storage and verification.

### 6.3.1 Deep Feature extraction

Deep learning-based approaches have drastically boosted the performance of recognition systems. In the heart of these methods, a deep convolutional neural network (CNN) with many layers is used for feature extraction. One important feature of these models is that they learn hierarchical feature representations. This means that first layers generate global features that can be transferred to another domain, while the last layers produce specialised features that belong to a certain domain. On the other hand, deep learning approaches need

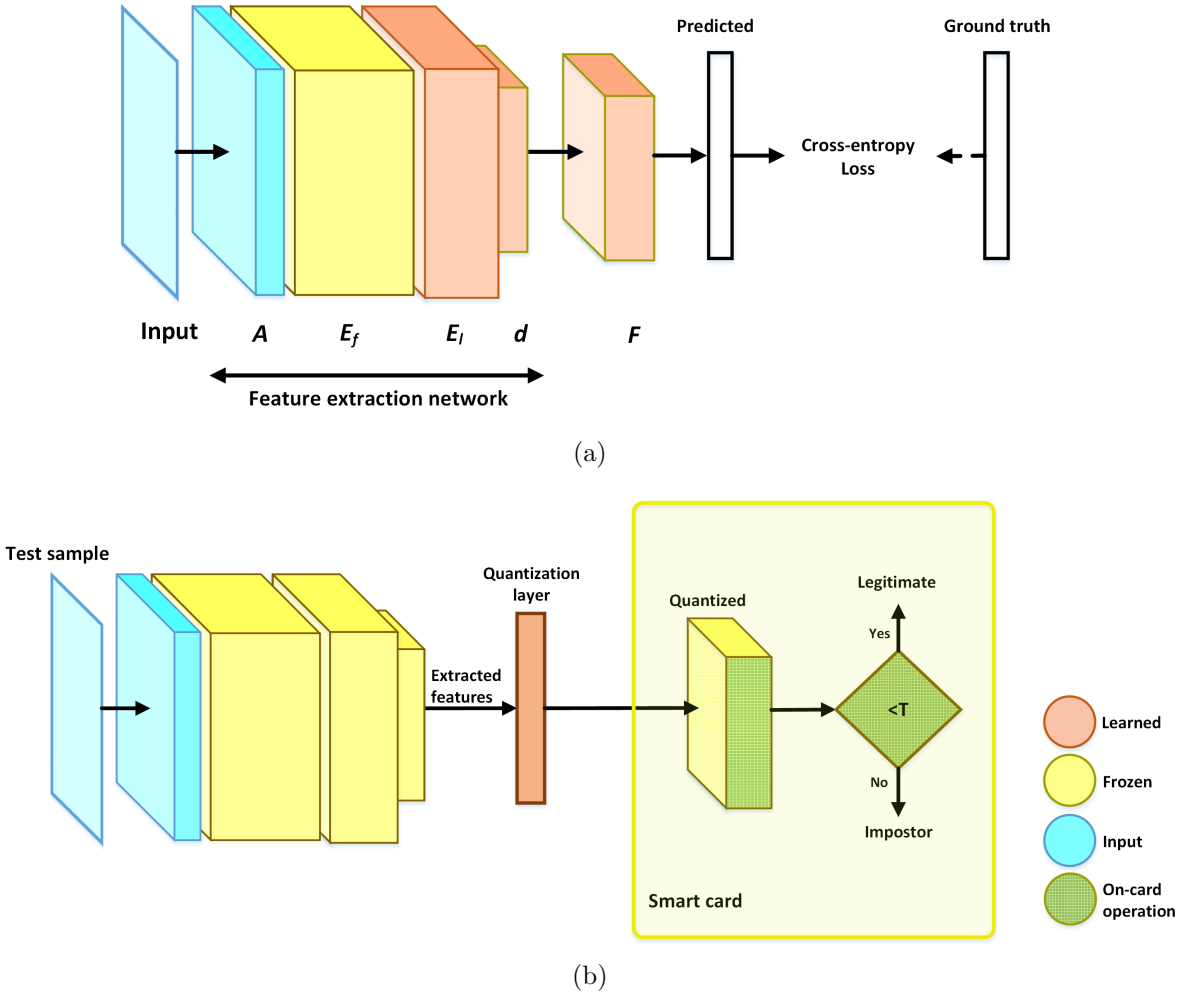



Figure 6.1 Overview of the proposed MMOC authentication system. (a) shows the model fine-tuning, and (b) shows the verification phase of the MMOC system.

a large dataset with many labels that is not available in every problem domains, or is a time consuming process. In *transfer learning*, a model is trained on a large dataset in one domain and the learned knowledge can used as a generic model to improve the optimization in another domain. Therefore, instead of training the model from the scratch, we customize the model by modifying the last layers of feature extraction network and adding our classification layers on top of it, then *fine-tune* the model by training a few top layers of feature extraction network and the classification layers.

Figure 6.1 shows the overview of the proposed method. Our fine-tune architecture consists of the following layers:

- One-to-many augmentation layer ($A$): Deep learning based approaches need large dataset.

However, most target datasets are small and may cause *overfitting* while fine-tuning the trained model. Therefore, a one-to-many augmentation layer is added to augment the training and the test data.

- Frozen sub-network ($E_f$): First layers of the pre-trained model are frozen, and are not learned during the fine-tune process.
- Learning sub-network ($E_l$): Unfrozen top layers of the pre-trained model to extract specialized features from the target dataset.
- Dimensionality reduction layer ($d$): Basically, the feature extraction network of trained models generates large feature vector from each image. However, the limited bandwidth to the SIM and constrained resources on the SIM, affect the real-time response of the proposed MOC-based system. Therefore we add a dimensionality reduction layer to reduce the size of the extracted feature vector.
- Classification sub-network ($F$): One or more fully connected layers are added on top of the network for classification.

This network is then trained end-to-end on the target dataset, and the feature extraction network is used to extract features for on-card user verification.

### 6.3.2 On-card user verification

For the on-card user verification, the feature extraction network is used to extract features from the test image. A *quantization* layer is employed to reduce the size of the model and the feature vector for user verification on the card. The frozen classification sub-network is quantized and migrated to the SIM card for verification. The classification sub-network outputs a score for the presented test image. This score is compared with the classification threshold to verify the legitimacy of the user. Fig 6.1b shows the on-card verification phase.

### 6.3.3 Quantization

Smart cards do not support floating point arithmetic, they only support signed integers, int-$k$, where $k$= 1,...,32. Therefore, we need a quantization scheme (Qt) to convert floating point data to $k$-bitwidth signed integers. In this section, we propose a scheme to quantize the extracted features (i.e., quantization layer) and the classification sub-network to migrate them to the smart card. Suppose that a floating point variable $r$ in range $[r_{min}, r_{max}]$ needs to be quantized to int-$k$ with $2^k$ quantization levels, $Q_{levels} = 2^k$, in range $[-2^{k-1}, 2^{k-1} - 1]$.

For int-8 we have $Q_{levels} = 256$ in range [-128, 127]. Real-valued $r$ is quantized to $r_q$ as follows [6].

$$r_q = Clip\Big\{round\Big(\Delta r + \Gamma\Big), -\Lambda, \Lambda - 1\Big\} \tag{6.1}$$

where $\Delta$, $\Lambda$, and $\Gamma$ are defined as

$$\begin{aligned} \Delta &= \frac{Q_{levels} - 1}{r_{max} - r_{min}}, \\ \Lambda &= Q_{levels}/2, \\ \Gamma &= -(\Lambda + \Delta r_{min}) \end{aligned} \tag{6.2}$$

*round(x)* stochastically rounds $x$ to $\lfloor x \rfloor$

$$round(x) = \begin{cases} \lfloor x \rfloor & \text{w.p } 1 - (x - \lfloor x \rfloor) \\ \lfloor x \rfloor + 1 & \text{w.p } x - \lfloor x \rfloor \end{cases} \tag{6.3}$$

*Clip* function is defined as

$$Clip(x, a, b) = max(a, min(b, x)) \tag{6.4}$$

Extracted feature are quantized to int-8 by passing through the quantization layer that applies Eq. (6.1) to the output of the feature extraction sub-network. In Eq. (6.1), we can also consider a symmetric range to save more bit space in intermediate operations, that is $r_q = clip\{round(\Delta r + \Gamma), (-\Lambda - 1), \Lambda - 1\}$; however, it does not impact the quantization accuracy. Considering the complexity of the classification sub-network, its quantization needs more careful attention that we discuss it in the next section.

**Classification sub-network quantization**

The frozen classification sub-network is, in fact, the *feed-forward* phase of the fully connected layers given by

$$y(\mathbf{x}) = \sigma\Big(\phi(\mathbf{x})^{(L+1)}\Big) \tag{6.5}$$

where $\sigma$ is the output activation function which squashes the prediction scores to [0,1], $\phi(\mathbf{x})$ is the basis function that transforms the feature-space, and $L$ is the total number of hidden layers. The basis function is a nonlinear function of linear combination of the inputs [130]. That is

$$\phi(\mathbf{x})^{(i)} = h^{(i)}\left(\mathbf{w}^{(i)} \cdot h^{(i-1)}(\mathbf{x}) + b^{(i)}\right) \tag{6.6}$$

where $h(.)$ is the layer activation function, $h^{(0)}(\mathbf{x}) = \mathbf{x}$, $(i)$ defines the hidden layer number, and $i = 1...L+1$. Rectified Linear Unit (ReLU) is used as the layer activation function. Since verification is a binary classification problem, *soft* or *hard sigmoid* [115] is used as the output activation function.

In the fine-tuning process, the model is trained end-to-end and then is frozen to obtain the model's parameters. Th parameters are quantized afterwards for the on-card verification. We quantize the feed-forward phase based on the following rules:

- Off-card quantization: the model's weights and the bias are quantized using Eq. (6.1). However, the model's weights and the bias are scaled but not shifted to keep the verification accuracy, that is, $\Gamma = 0$. The weights are quantized to int-8, $Q_{levels} = 2^8$, maximum and minimum of the weights are considered as $r_{min}$ and $r_{max}$ to calculate $\Delta_{weights}$ in (6.2). We assign a wider range to the model's bias compared to the model's weight to keep the performance accuracy close to the original model. Therefore, we quantize the bias to int-16, $Q_{levels} = 2^{16}$ in (6.1), and scale it in the range of the weight vector, that is $\Delta_{bias} = \Delta_{weights}$. Inputs to the model are quantized using Log-Qt method described in section 6.3.4.

- On-card quantization: To avoid overflow of the intermediate multiply-accumulate calculations in deeper layers, the output of layer activation function is clipped from the first layer. We use ReLU-$n$ as the layer activation function, defined as

$$h(x) = max(min(x, 0), n) \tag{6.7}$$

  where $-Q_{levels}/2 \leq n \leq Q_{levels}/2 - 1$ is obtained using validation set.

### 6.3.4 On-card Optimization

The main performance bottleneck in on-card verification is the multiply-accumulate calculation (MAC) in the forward pass in the form of

$$o = \mathbf{w}^{\mathrm{T}}\mathbf{x} = \sum_{i=0}^{d-1} w_i \times x_i \tag{6.8}$$

where $w_i$ and $x_i$, $0 \leq i \leq d-1$ are the entries of weight and input vectors of length $d$, respectively. As $d$ increases MAC takes more time on the card which affects the real-time functionality of the on-card verification system. Therefore, we *log-quantize* (Log-Qt) the input vector as follows

$$\tilde{r_q} = round(\log_2(|r_q|)) \tag{6.9}$$

where $r_q$ is obtained using Eq. (6.1), and $round(x)$ is round to the nearest integer or Eq. (6.3). Then, we can rewrite Eq. (6.8) as

$$o \simeq \sum_{i=0}^{d-1} \mathrm{sign}(x_i).(w_i \ll \tilde{x_{iq}}) \tag{6.10}$$

where $a \ll b$ shifts $a$, $b$ *bits* to the left, and

$$\mathrm{sign}(x) = \begin{cases} -1, & \text{if } x < 0. \\ 1, & \text{otherwise.} \end{cases} \tag{6.11}$$

$x_{iq}$ in Eq. (6.10) is obtained using Eq. (6.1) where $\Delta_{features}$ and $\Gamma_{features}$ are calculated with $r_{max}$ and $r_{min}$ among all input vectors in the training set.

The implementation of Eq. (6.8) or Eq. (6.10) is a simple `loop` instruction of order $O(n)$. However, as the size of input vectors increases the execution of the `loop` increases noticeably on the card leading to a non real-time authentication system. Therefore, we propose an optimization architecture based on *loop optimization* techniques used in modern compiler design [144] to improve the on-card MAC time. We employ the following techniques in our optimization architecture.

1. *Loop fission* is a compiler optimization technique in which a loop is split into multiple loops. This technique, especially for loops with large bodies, is used to achieve better utilization of locality of reference.

2. *Loop unrolling* in compiler optimization, replicates the loop body for multiple times. This method, improves the loop performance by reducing the overhead for loop index

increment and loop-exit condition test, and minimizing branch penalties. However, this loop transformation increases the program code size which arises a new implementation challenge [118].

3. *Loop peeling* splits the remainder of the loop iterations from the previous step, and performs them outside of the loop body.

4. *Loop fusion* combines several loops in a single loop. It improves the locality of reference especially when common data is accessed among distributed loops.

Fig. 6.2 compares the optimized on-card feed-forward phase with the original one.

#### Memory management

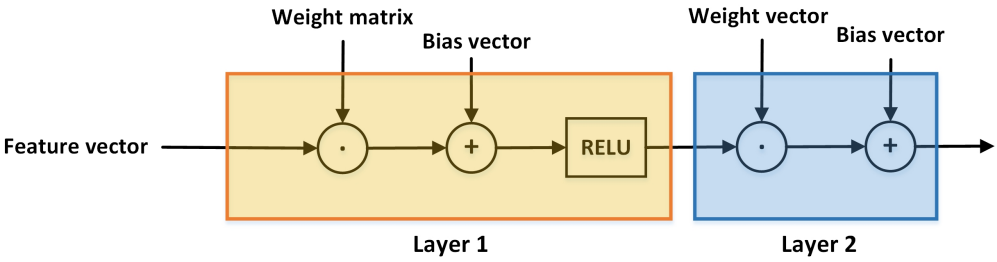
Smart cards have EEPROM and RAM for data storage. EEPROM offers more storage space compared to RAM. On the other hand, writing on EEPROM is more than 30 times slower than RAM [121]; moreover, EEPROM has limited write cycle. Writing many times on EEPROM would damage the card's EEPROM. Smart card OS does not handle memory management; therefore, in order to have a better use of the card's memory and improve the execution time, we should manage the memory manually. Layers' weights and biases are initially stored in EEPROM to be used in verification phase, when a feature vector is transmitted to the card, we store them in RAM; moreover, we fetch the model's internals manually from EEPROM to RAM before going through the classification sub-network in order to reduce memory access time. This way, we keep EEPROM healthy and decrease the on-card execution time.

## 6.4 Performance Evaluation

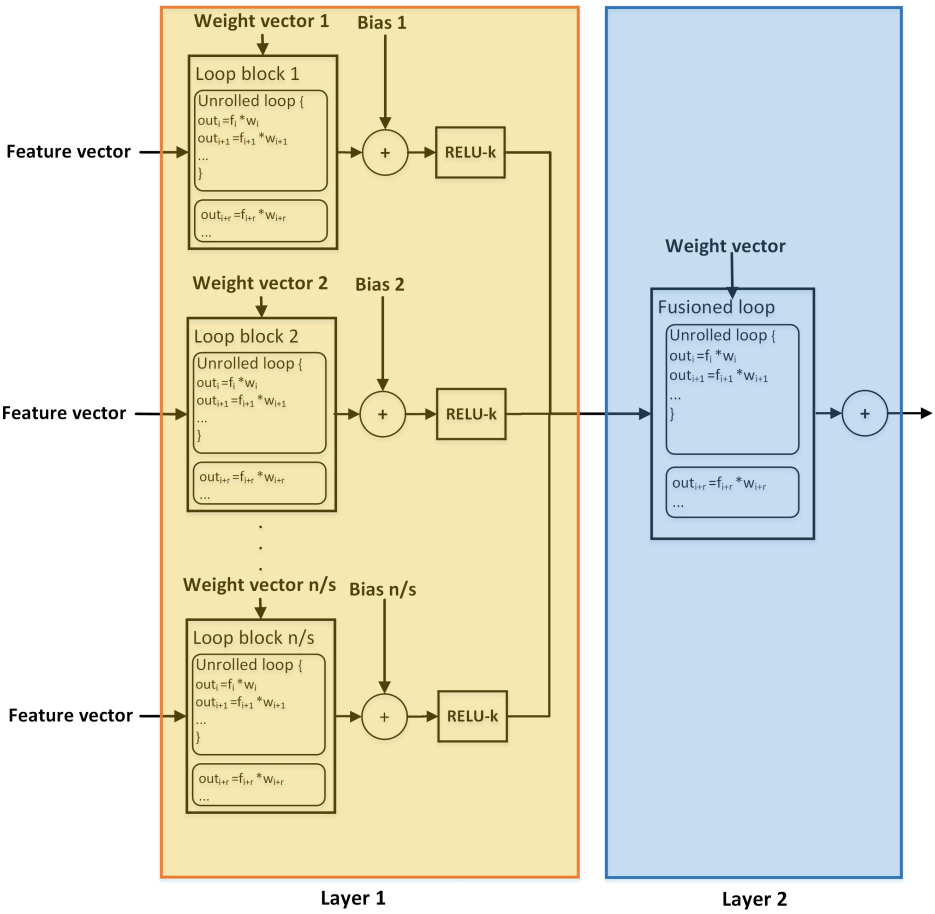
### 6.4.1 MOBIO face dataset

MOBIO dataset [43] contains videos and audios of 152 subjects from 5 different countries at 6 different sites[1]. The data were collected in 2 phases where each phase consists of 6 sessions. A NOKIA N93i mobile phone and a standard 2008 MacBook laptop computer were used to capture videos and audios. Session 1 consists of videos recorded by the laptop computer while the remaining 11 sessions contain the recorded data from the mobile phone. Fig. 6.3

[1]The data was recorded at the following sites: the Brno University of Technology (BUT), Idiap Research Institute (IDIAP), University of Avignon (LIA), University of Manchester (UMAN), University of Surrey (UNIS) and University of Oulu (OULU).




(a)




(b)

Figure 6.2 Classification sub-network. (a) is the original sub-network. (b) is the optimized version for on-card verification. The main loop with size $n$ is split to multiple loops with size $s$. The weight matrix is vectorized and $s$ vectors are fed into each loop block. Each loop is unrolled with unroll factor $r$. The remaining iterations are performed outside the main loop (peeled loops). The outputs of the first layer are fused in one loop in the second layer with loop unrolling and loop peeling.

shows samples from the MOBIO dataset. In our experiments we only use the mobile sessions

of the dataset. One mobile session is used for model training, one as a validation set, and the remaining sessions are used as a test set.

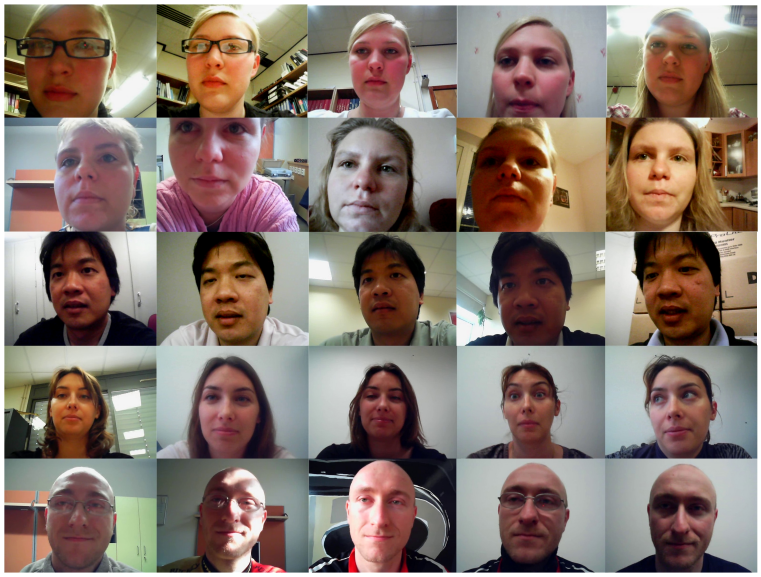


Figure 6.3 Subject's images from MOBIO dataset.

### 6.4.2 Evaluation configuration

For the evaluation purpose, we use *Resnet50* [33] trained on *VGGFace2* dataset [39]. VGGFace2 dataset contains 3.31 million images of 9131 subjects, with an average of 362.6 images for each subject. We fine-tune the trained model for our target dataset. Input images are of size $224 \times 224 \times 3$. The augmentation layer augments the input by flipping and rotating the input images. We use *one-vs-all* classification [119] to train a person-specific binary classifier for each subject. All images from the specific user are considered as "one" and the other samples in the set are "all" class. This approach leads to an imbalanced dataset; therefore, we apply *over sampling* technique to mitigate the imbalanced dataset effect on the performance evaluation [120]. For the end-to-end training, we minimize binary cross entropy by Adam optimizer with learning rate $\eta$ of $10^{-4}$. However, for fine-tuning, we reduce the learning rate by 10 to avoid model overfitting. The feature extraction sub-network of the model outputs 2048 deep features from each image. In order to make it suitable for on-card implementation, we reduce its dimension to 64. Out classification sub-network consists of two fully connected layer. First layer has 64 nodes followed by relu-$n$ activation function, and the second layer is one node layer with sigmoid function to classify the input as a legitimate or an impostor. For on-card verification module, we first flatten the first layer weight matrix column-wise and transfer it to the smart card. On the card, we vectorize the flatten

weight matrix column-wise. The optimization scheme splits the main loop to 4 loop blocks and 16 weight vectors are fed into each block. Each loop is unrolled with unroll factor of 64, $r = 64$. Moreover, we use ReLU-*10* as layer activation function. A SIM card with 1 MB of Flash and 40 KB of RAM with a secure processor running Java Card version 3.0.2 classic is used for performance evaluation. The SIM card also supports 32-bit integers. We use T=0 protocol for communication with the SIM card through a contact interface. We report the performance of the proposed system in Area Under Curve (AUC), Equal Error Rate (EER). EER is an error rate where False Positive Rate (FPR) equals False Negative Rate (FNR). We also report *threshold* metrics, recall (REC=$\frac{TP}{TP+FN}$), precision (PRE=$\frac{TP}{TP+FP}$) where TP, FP, and FN are the number of true positive, false positive, and false negative samples for a given threshold, respectively.

Fig. 6.4 shows the on-card execution time for user verification in original sub-network and the optimized architecture. Using the original sub-network (Fig. 6.2a), the verification takes 60.71 seconds (std = 1.75 s) in a sub-network with 64 nodes in the second layer. Using the optimized architecture (Fig. 6.2b) it drops to 2.13 seconds (std = 0.76 s). However, when we apply Log-Q to the model's input, the verification times decreases more and it outputs the decision in 1.37 seconds (std = 0.54 s). We gain about $\frac{60.71}{2.13} \approx 28.5\times$ speed-up over the original architecture using the optimized architecture and $\frac{60.71}{1.37} \approx 44.3\times$ speed-up using the optimized architecture with Log-Q. Moreover, the mean quantization error $\frac{1}{N}\|\mathbf{x_Q} - \mathbf{x}\|_1$ where $\mathbf{x_Q}$ is quantized input vector $\mathbf{x}$ is less in Log-Qt scheme using Eq. (6.9) than in Qt quantization scheme using Eq. (6.1) by around 25% in the test dataset. It also depicts that the model transfer time is unchanged in both implementations. The main bottle neck in model transfer is the transmission of first layer's bias vectors. We quantize the model's bias to a wider range (16-bit signed integers) in order to keep the performance accuracy; however, the communication channel to smart cards is a byte stream. Therefore, we need to convert bias vector to 8-bit integers on the device and convert them back to 16-bit integers on the card which increases the transfer time as the vector size increases. Obviously, as can be seen in the figure, the model transfer time is independent of the architecture (optimized or original) and increases as the number of nodes increases in the second layer.

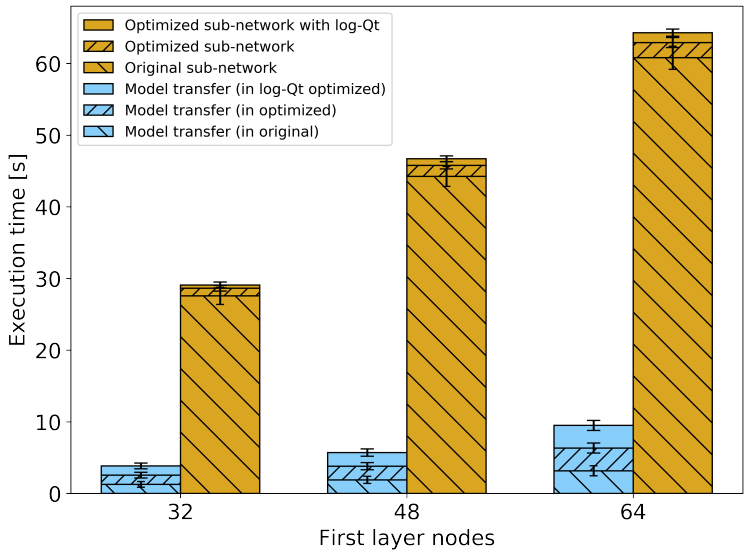



Figure 6.4 Execution time for different sub-network size.

Table 6.1 Performance accuracy of the proposed generic MMOC authentication system. Real-valued results are depicted in square brackets (no quantization is applied to obtain the real-valued results).

| Site | Nodes | EER (%) | AUC (%) | PRE (%) | REC (%) |
|---|---|---|---|---|---|
| BUT | 64 | 0.9 [0.4] | 99.2 [99.5] | 98.3 [99.1] | 97.0 [97.8] |
| | 48 | 1.0 [0.5] | 99.0 [99.4] | 98.3 [99.0] | 96.9 [97.7] |
| | 32 | 1.3 [0.6] | 98.6 [99.2] | 97.9 [98.9] | 96.1 [97.4] |
| IDIAP | 64 | 0.4 [0.1] | 99.7 [99.8] | 99.7 [99.8] | 97.1 [97.6] |
| | 48 | 0.4 [0.1] | 99.7 [99.8] | 99.7 [99.8] | 97.0 [97.6] |
| | 32 | 0.4 [0.1] | 99.7 [99.8] | 99.7 [99.8] | 97.1 [97.6] |
| LIA | 64 | 1.4 [0.8] | 98.8 [99.2] | 98.6 [99.1] | 95.7 [96.5] |
| | 48 | 1.7 [1.1] | 98.5 [98.9] | 98.1 [98.6] | 95.5 [96.1] |
| | 32 | 2.2 [1.6] | 98.0 [98.6] | 98.0 [98.5] | 94.8 [95.8] |
| UMAN | 64 | 0.5 [0.2] | 99.5 [99.7] | 99.5 [99.7] | 96.8 [97.2] |
| | 48 | 0.5 [0.2] | 99.5 [99.7] | 99.4 [99.7] | 96.7 [97.0] |
| | 32 | 0.6 [0.3] | 99.1 [99.5] | 99.1 [99.5] | 96.2 [96.7] |
| UNIS | 64 | 0.4 [0.2] | 99.4 [99.6] | 99.5 [99.7] | 96.6 [97.3] |
| | 48 | 0.5 [0.2] | 99.4 [99.6] | 99.4 [99.7] | 96.5 [96.9] |
| | 32 | 0.7 [0.3] | 99.0 [99.4] | 98.8 [99.3] | 96.2 [96.5] |
| UOULU | 64 | 0.4 [0.1] | 99.7 [99.8] | 99.7 [99.8] | 97.1 [97.6] |
| | 64 | 0.4 [0.1] | 99.6 [99.8] | 99.7 [99.8] | 97.0 [97.6] |
| | 32 | 0.5 [0.2] | 99.5 [99.7] | 99.5 [99.7] | 96.8 [97.2] |

Table 6.1 shows the system's performance accuracy on different MOBIO dataset sites. Applying the quantization scheme increases the EER of the system. In the worst case (i.e., LIA site), EER increases to 1.4%; however, this value is still less than 5% that is the acceptable EER for a reliable system [41]. On the other hand, decreasing the number of first layer's nodes affects the system's accuracy slightly. For example, in LIA site, decreasing the number of nodes to 32 increases the system's EER to 2.2% while decreases the execution time to about 500 ms. This depicts that we can improve the execution time by carefully cutting down the number of nodes in hidden layers while keeping the performance accuracy at an acceptable level. In authentication systems, false positive rate (FPR) is more crucial than false negative rate (FNR), and a good authentication system tries to keep the FPR as low as possible. This means that higher PER is more desired than REC. As can be seen in the table, the proposed system, thanks to the transfer learning, has both PRE and REC in an acceptable percentage (above 95%) with PRE slightly dominating REC .

## 6.5 Conclusion

In this paper, we proposed a generic model for privacy-preserving authentication systems using Mobile Match-on-card (MMOC) technique that can be used for various biometrics using transfer learning. We fine-tuned Resnet50 trained on VGGFace2 and modified the feature extraction network and classification sub-network to make the model suitable for on-card implementation. Moreover, we applied quantiation schemes (Qt and Log-Qt) to the model's internals and the input along with an on-card optimization architecture. The first one helped us to keep the performance accuracy close to the real-valued model by reducing the quantization error (EER of 0.4%-2%); moreover, it decreased the on-card memory foot-print and sped up the forward pass by converting multiply-accumulate to bit-shift-accumulate. The latter assisted us to reduce the on-card verification time drastically where we gained 44.3× speed-up over the original architecture. Our results showed the potential of MMOC as a privacy-preserving authentication system with high performance accuracy, low memory foot-print and real-time response even using resource consuming classification methods such as DNN.

# CHAPTER 7 GENERAL DISCUSSION

In this chapter, we summarize our results with respect to the research objectives set in section 1.4. After that, we will take a look at the methodological approach that we followed. Finally, we will end this chapter with an analysis of the overall results that we have obtained in order to show the potential of the proposed system for implementation on real smartphones.

## 7.1 Summary of results

The main objective of this thesis was to design a lightweight and secure biometric-based authentication system for smartphones. We defined three research phases in order to tackle the related issues, and fulfill the main objective of this research project.

In the first phase, a secure active authentication system for mobile users was proposed. In order to increase the security and the privacy of users' biometric information, the use of SIM card as a secure element was proposed. Touchscreen biometric that is the main means of interaction with smartphones was investigated in this phase. A DNN classifier has been employed that significantly improved the accuracy of authentication systems, and has shown promising accuracy in touchscreen biometric-based authentication. A cloud-assisted architecture was proposed with a SIM card in the heart of the system where the cloud part was used for model training and model selection. The model, then, is migrated to SIM card for secure authentication. Although smart cards show high security in many applications such as telecommunication, transportation, banking, and etc., their resource limitations hinder their successful presence in biometric systems with computationally expensive operations (i.e., large feature vectors and complicated classifiers such as DNN). We addressed this challenging task in MMOC authentication by proposing a quantization scheme for deep neural networks on smart cards. Moreover, we also introduced a speed-up technique to decrease the network inference (verification) time on the card. Multi-stroke authentication uses several touchscreen gestures (strokes) such as swipes, flicks, slides to improve the accuracy of the system. Therefore, a score fusion technique was also employed on the SIM card to achieve higher on-card accuracy.

In the second phase, we extended our research towards a more generalized privacy-preserving authentication by considering biometrics with big template size, particularly face biometric that is popular among the research community and the industry as well. Transfer learning has shown a prominent success in model-based object recognition systems by transferring

knowledge between domains. These networks can extract most discriminative features of objects in the target domain. An accurate and lightweight model based on transfer learning was introduced in this phase. Owing to the accuracy of the extracted features from the pre-trained model, the verification can be done on the card by a threshold comparison technique that shows a higher security compared to the first phase. Euclidean distance is used for thresholding. However, this simple operation is not possible on the card; therefore, we employed a numerical approach to implement this distance metric on the card. Quantization scheme and optimization technique were borrowed from phase 1 to make inputs and model's internals understandable for SIM cards. Finally, the proposed architecture was implemented on real Android devices in order to compare resource overhead of MMOC authentication with a simple encryption version of the system.

In the last phase, we proposed a generic secure authentication system based on MMOC technique that can be applied to different biometric systems that take advantage of transfer learning. In this architecture, we modified the network layers of the reference model to produce outputs that are suitable for smart cards, plus an extra quantization layer to convert model's output to an appropriate integer range for smart cards. The classification sub-network consisting of one or two fully connected layers were migrated to the SIM card for secure verification. However, feed forward phase of a DNN is a heavy computation for smart cards. Therefore, a quantization scheme and an optimization architecture were proposed to reduce forward pass time on the card and achieve real-time authentication response. A log quantization scheme was introduced to transform multiply-accumulate in vector dot product to bitshift-accumulate that can be done faster in hardware. In addition, using compiler optimization techniques, a new architecture for expensive on-card loop operations that are used in vector dot product calculations, was presented that reduced the execution time drastically.

## 7.2 Experimental environment

In order to evaluate the performance of the proposed system on the card, we built a test bed that consists of the following modules (software and hardware):

- A laptop computer with Core i7-5600U @ 2.60GHz CPU, and 16 GB of memory running Windows 10 Pro.
- A smartphone with a 6.4-inch display, an octa-core CPU (2x1.6 GHz + 6x1.35 GHz), 3 GB of memory, and 32 GB of internal storage running Android 9 pie.

- A SIM/eSIM with ARM SC300 secure core processor that is tamper resistant and robust against side channel and fault injection attacks, with 1.5 MB of Flash and 53 KB of RAM running Java Card version 3.0.4 classic.
- A SIM/eSIM card with 1 MB of Flash and 40 KB of RAM with ARM secure core processor running Java Card version 3.0.2 classic.
- Smart Card reader with bandwidth of 115 kbps.
- PC application written in Java for communication with the smart card.
- LOGOS TBII emulation environment for injecting applets into MNO's profile.
- SIM applets written in Java card (v3.0.2 or v3.0.4).
- Tensorflow and Tensorflow Lite for model training and transfer learning.

Using this test bed, we reported different performance metrics such as execution time, EER, AUC, REC, PRE, FNR@FPR to show the potential of the system as a secure authentication system for mobile users.

### 7.3 Results analysis

The conducted research shedded light on less discovered potentials of smart cards for secure authentication systems. Our results revealed that although smart cards do not have sufficient resources to implement a complete authentication system, they can be used as a secure element to store biometric templates to increase the privacy of users and to verify users in a secure and isolated environment. Smart cards do not support floating point values; therefore, the first step toward the MMOC implementation is a quantization scheme that results a performance accuracy close to the original model. The results showed that the proposed quantization scheme produces low quantization error and we saw that the results on the card are close to the real valued results.

In the touchscreen active authentication system that used a simplified on-card DNN inference for user verification, a multi-stroke fusion technique and a loop unrolling technique helped us to demonstrate the feasibility of continuous authentication system even when the frequency of sampling is low (i.e., around 1 sample/second), and an expensive classifier such as DNN is implemented on the card.

Then, we studied an MMOC active authentication system using more challenging biometrics such as face. The results showed that we can implement a full face-based active authentication system using template-based approach on smart cards without relying on the outside resources, with the enrollment and the verification phases completely implemented on the card. User on-card verification takes less than 130 ms which does not impact the active authentication process. The enrollment takes about 15 seconds to determine the decision threshold on the card using 100 face templates; however, it does not affect the real-time on-card verification. We implemented our active authentication system on a real device and showed the platform overhead reduction compared to a simple biometric encryption method. Moreover, the effect of quantization bit-width on performance accuracy revealed that we can reduce the memory footprint on the card drastically while keeping the system's accuracy at an acceptable level.

Finally, an optimization architecture proposed to reduce the inference time for biometric systems that rely on DNN classifiers such as those that use transfer learning. Our results revealed that using a sophisticated quantization scheme (log quantization) and a loop optimization technique to improve the locality of reference in accessing memory such as loop unrolling, loop fission and loop fusion, we can gain up to 45× speed up in forward pass time, leading to a real-time authentication system, even with a DNN model that has more parameters and needs more computation. Moreover, reducing the size of the classification sub-network gave us more gain in reducing the execution time on the card than increasing the EER of the system.

### 7.3.1 Cross-platform authentication

From privacy point of view, when users change their devices, they need to enroll their biometric templates (such as fingerprint or face) on the new device. This is tedious task for the user and more importantly it increases the risk of identity theft, spoofing attacks, and unauthorized access to the services. SIM cards have platform independency and potability features. This means that the user can enroll in one device, then she can insert her SIM card containing the biometric templates and the verification engine into another device and use the authentication system without re-enrolling the biometric templates, and not revealing any data to third parties. The results showed that the MMOC face biometric system is robust against platform changes, and can be used on different platforms.

### 7.3.2 eSIM implementation

eSIM technology as the emerging SIM card is gradually going to be used in new smartphones and smartwatches. This technology also supports Java Card technology that is used to develop applets for SIM cards. However, eSIM supports Java Card version 3.0.4. Therefore, any applet compatible with Java Card 3.0.4 can also be used in eSIM. Unlike SIM cards that are personalized (i.e., the MNO's profile with proprietary applets loaded into cards) in the production phase, in eSIM cards the personalization is carried out over the air using remote SIM provisioning (RSP) platform that loads the profile into the eSIM using transport protocols such as https or SMS. Moreover, RSP also supports profile portability which means that the user can request to transfer her MNO's profile with the applets to a new device. Therefore, our cross-platform authentication solution is also applicable to smart devices with eSIM cards. Since many IOT devices will be equipped with eSIM in future, our proposed MMOC system is a promising solution for IOT devices as well.

# CHAPTER 8 CONCLUSION AND RECOMMENDATIONS

This chapter provides a summary of the research work presented in this thesis. First, we present the main contributions of the thesis. After that, the limitations of the work will be discussed. We will finalize the thesis by several recommendations to show directions for the future works.

## 8.1 Summary of works

This thesis aimed to design a secure biometric-based authentication system for mobile users. This is a challenging task because increasing the security and privacy of an authentication system affects other properties of the system. More precisely, increasing the security will decrease the system's performance (i.e., increases EER) and will decrease the efficiency (i.e., increases the verification response time). To address these challenges, a Mobile Match On Card (MMOC) authentication system was proposed. Two system architectures were proposed to evaluate the system under different processing loads, namely using model-based authentication by employing a DNN classifier and a template-based authentication by applying distance metric thresholding. To address performance issues, a quantization scheme was proposed to keep the system accuracy close to the original model even on smart cards. To improve the system's efficiency, an optimization architecture was introduced that helped us to decrease the verification time drastically on the card. Moreover, the system was implemented on real Android devices to evaluate its resource consumption compared to encryption-based methods. This thesis made the following contributions, most of them are among the first works in the field of biometric-base authentication systems for constrained devices:

1. **Design a secure system for authentication on smartphones**: two architectures for a secure biometric-based active authentication system were proposed. A cloud-assisted architecture, a model-based authentication system, that used cloud resources for model training and model selection which has a faster but less secure enrollment phase. A full MMOC architecture, a template-based authentication system, that only relies on card resources. This architecture has a slower enrollment but is a more secure architecture.

2. **Quantization scheme**: we introduced a quantization scheme to make inputs readable for smart cards. This quantization scheme should be designed in a way that produces low quantization error in order to keep the system's performance comparable to the

original model. This quanitzation scheme was also applied to the model's internals in the cloud-assisted architecture.

3. **On-card deep neural network classifier**: we proposed a DNN-based on-card verification system. Considering the resource limitations of smart cards, the original DNN model is not implemented on the card. Since only the inference of the network is migrated to the smart card, we qunatized the feed forward of the network by considering the nature of the model's internals and inputs and bit-width overflow of intermediate computations.

4. **Design a generic model for MMOC using transfer learning**: we designed a generic model for MMOC authentication using transfer learning. Deep learning-based approaches have made a significant improvement in recognition problems. These systems show high recognition accuracy by extracting deep representation of the given object. This technique can also improve user verification in biometric systems. However, this method is not readily implementable on smart cards. Therefore, we proposed a generic model to modify the network architecture and make it suitable for MMOC systems.

5. **Optimization architecture**: an optimization architecture using compiler optimization techniques was presented to decrease the execution time on the card. This optimiazation technique is really crucial especially when the verification phase is a deep neural network inference. This helped us to make MMOC authentication a feasible solution for secure authentication on smartphones.

6. **Platform implementation**: we implemented an MMOC face-based active authentication system on real Android devices. The challenge in the implementation is granting access to the SIM card from the Android OS. This needs development of an access controller applet on the SIM. The results revealed a slight performance gain compared to encryption-based biometric protection solutions.

## 8.2 Limitations

Despite the aforementioned contributions, the work carried out within the framework of this thesis still has certain limitations:

- Resource limitations: The main limitation of this research study is related to the resource limitations of smart cards. Smart cards do not have high processing power and

high memory capacity. The feasibility of implementing a specific system should be studied prior. SIM cards are basically used to store an MNO's profile that needs small memory to run. Therefore, the proposed solution to store biometric templates privately on SIM cards may not work on SIM cards with low memory capacity. Another limitation is the channel bandwidth for smart card's contact interface which is roughly 115 kbps. It can cause a higher delay in transferring large biometric templates to the SIM card.

- Security flaws: The proposed solution uses the SIM/eSIM card as a secure element for secure authentication. Most of the modern SIM/eSIM cards are robust against attacks on smart cards such as side channel attacks or fault injection attacks. However, the system will be vulnerable to these attacks if the card OS does not implement appropriate mechanisms to defeat these attacks. Moreover, the communication channel between the application on the phone with the applet on the card is not a secure channel. Although eavesdropping the communication channel needs a sophisticated attack, this issue can be resolved by establishing a secure channel to the SIM card. Moreover, if a secure channel between the phone and the cloud server is not establish, the system will be exposed to man in the middle (MITM) attacks as well.

- DNN implementation limitation: The proposed model-based authentication system with on-card DNN inference uses a network with two hidden layers with the maximum of 4224 parameters. This network configuration gives high performance accuracy in a real-time manner using our proposed optimization solution. However, increasing the input vector size and the number of hidden layers, makes the applet oversized and affects the efficiency of the system. Moreover, since 2-dimensional arrays are not supported in smart cards, matrix inner product should be split into separate vector dot products increasing the total inference time.

- Carrier privilege: The proposed secure solution is a solution in the hands of mobile network operators. In fact, MNOs lock the SIM cards in the field to prevent any potential attacks to the subscribers' information, and even if the smartphone's OS grants access to the SIM, the SIM's OS blocks any communications from the outside world. Therefore, the implementation on real and in the field SIM cards needs the carrier privilege that is not granted to untrustworthy third party developers.

- eSIM implementation limitation: Although eSIMs support Java Card technology used in SIM cards, and we implemented our solution on the version of Java Card compatible with eSIM; however, the real device implementation was done on the SIM card. The

main limitation on eSIM implementation is that eSIM is soldered into the device and conventional personalization process is not possible for eSIM. A remote SIM provisioning (RSP) system is required to download the profile and the applet into the eSIM which was not available at the time of implementation. Moreover, profile portability of eSIMs are dependent on the RSP platform to support this functionality.

- Generic model limitation: The generic model presented in section 6 is based on transfer learning method. Therefore, it is applicable to any biometric system that has a pre-trained model on a large and relevant dataset. If a specific biometric system does not have any pre-trained models, the proposed model is not a good solution for it. However, other components of the system such as quantization or on-card optimization are generic to all MOC-based authentication systems.

## 8.3 Future Work

We end this thesis by suggesting several research directions than can lighten the path for further research works in this interesting research topic, also can tackle the limitations listed above that we encountered during this research.

- Quantization scheme is the key part to improve the performance of an authentication system on the card. A wiser quantization mechanism with a more innovative idea can be proposed to build an authentication system with close to zero quantization error even when quantizing the model with fewer number of bits (e.g., 2-bit quantization)
- The core of verification phase in DNN-based verification is a vector dot product which is an expensive computation on smart cards. Therefore, a good optimization technique is crucial to make a real-time authentication system even when the number of hidden layers are increased and more parameters are in the network. Low level programming and taking advantage of hardware implementations are encouraged to address this issue.
- The implementation of the proposed method on eSIMs and integrated SIM (iSIM), the future of SIM cards, is also an interesting research direction. iSIM enables hardware Original Equipment Manufacturers (OEMs) and processor design companies to design system-on-a-chip (SOC) architectures that integrate SIM functionality with an existing, onboard processor and cellular modem.
- In chapter 4, we successfully fused multi strokes to decrease the recognition error. We can extend the fusion technique to integrate two or more biometrics such as face, fingerprint, or behavioral biometrics, as another research line.

- With many built-in sensors in modern smartphones such as camera, microphone, accelerometer, and gyroscope, the sensor-based authentication systems are becoming more popular in mobile user authentication. However, security of this newly emerged authentication is not studied enough in the literature, which needs more attention from the academic community.

- Throughout the thesis, we showed the feasibility of implementation of deep neural network on smart cards. Some problem domains may need more specialized neural networks. Therefore, study of implementation feasibility of other types of neural network such as convolutional neural network (CNN) or recurrent neural network (RNN) on smart cards also brings up a new challenging research line.

- Proliferation of IOT devices and wearables in our personal lives is undeniable which requires a reliable authentication solution. Although our proposed method is also applicable to other resource constrained devices, biometric authentication on IOT devices can be studied within a new research work, to assess its performance and to show its limitations.

# REFERENCES


[1] M. Shahzad, A. X. Liu, and A. Samuel, "Behavior based human authentication on touch screen devices using gestures and signatures," *IEEE Trans. Mobile Comput.*, vol. 16, no. 10, pp. 2726–2741, Oct. 2017.

[2] M. Harbach *et al.*, "It's a hard lock life: A field study of smartphone (un)locking behavior and risk perception," in *Proc. Symp. Usable Privacy Secur. (SOUPS)*, July 2014, pp. 213–230.

[3] T. McGill and N. Thompson, "Old risks, new challenges: exploring differences in security between home computer and mobile device use," *Behav. Inform. Technol.*, vol. 36, no. 11, pp. 1111–1124, July 2017.

[4] M. Alsaleh, N. Alomar, and A. Alarifi, "Smartphone users: Understanding how security mechanisms are perceived and new persuasive methods," *PLOS ONE*, vol. 12, no. 3, pp. 1–35, Mar. 2017.

[5] D. Tapellini, "Smart phone thefts rose to 3.1 million in 2013 industry solution falls short, while legislative efforts to curb theft continue," 2014. [Online]. Available: https://www.consumerreports.org/cro/news/2014/04/smart-phone-thefts-rose-to-3-1-million-last-year/index.htm

[6] S. Keykhaie and S. Pierre, "Mobile match on card active authentication using touchscreen biometric," *IEEE Trans. Consum. Electron.*, vol. 66, no. 4, 2020.

[7] M. Koyuncu and T. Pusatli, "Security awareness level of smartphone users: An exploratory case study," *Mob. Inf. Syst.*, vol. 2019, pp. 1–11, May 2019.

[8] A. Mylonas, A. Kastania, and D. Gritzalis, "Delegate the smartphone user? security awareness in smartphone platforms," *Comput Secur.*, vol. 34, pp. 47–66, May 2013.

[9] Z. Benenson, O. Kroll-Peters, and M. Krupp, "Attitudes to it security when using a smartphone," in *Proc. FedCSIS*, Sept. 2012, pp. 1179–1183.

[10] G. Davis and R. Samani, "Mcafee mobile threat report q1, 2018," McAfee, Santa Clara, CA, USA, Tech. Rep., 2018.

[11] A. Juels and M. Wattenberg, "A fuzzy commitment scheme," in *Proc. 6th ACM Conf. Comput. Commun. Secur. (CCS)*, 1999, pp. 28–36.

[12] M. van Dijk *et al.*, "Fully homomorphic encryption over the integers," in *Proc. Annu. Int. Conf. Theory Appl. Cryptograph. Techn.*, 2010, pp. 24–43.

[13] Arm, "Arm cortex-m series processors." [Online]. Available: https://developer.arm.com/ip-products/processors/cortex-m

[14] Android, "Uicc carrier privileges." [Online]. Available: https://source.android.com/devices/tech/config/uicc

[15] M. Abuhamad *et al.*, "Sensor-based continuous authentication of smartphones' users using behavioral biometrics: A contemporary survey," *arXiv preprint arXiv:2001.08578*, 2020.

[16] A. K. Jain, A. Ross, and S. Pankanti, "Biometrics: A tool for information security," *IEEE Trans. Inf. Forensics Security*, vol. 1, no. 2, pp. 561–572, 2006.

[17] P. Perera and V. M. Patel, "Efficient and low latency detection of intruders in mobile active authentication," *IEEE Trans. Inf. Forensics Security*, vol. 13, no. 6, pp. 1392–1405, Dec. 2018.

[18] R. Tronci, G. Giacinto, and F. Roli, "Dynamic score combination: a supervised and unsupervised score combination method," in *International Workshop on Machine Learning and Data Mining in Pattern Recognition*, 2009, pp. 163–177.

[19] R. D. Findling, M. Holzl, and R. Mayrhofer, "Mobile match-on-card authentication using offline-simplified models with gait and face biometrics," *IEEE Trans. Mobile Comput.*, vol. 17, no. 11, pp. 2578–2590, Nov. 2018.

[20] R. Spreitzer *et al.*, "Systematic classification of side-channel attacks: A case study for mobile devices," *IEEE Commun. Surv. Tutor.*, vol. 20, no. 1, pp. 465–488, 2018.

[21] S. microsystems, "Java card applet developer's guide," Sun microsystems, Palo Alto, CA 94303 USA, Rapport technique, 19981.

[22] G. Association, "Rsp technical specification version 2.2.1," GSMA, Tech. Rep. Official Document SGP.22, 2018.

[23] Z. Liu and S. Song, "An embedded real-time finger-vein recognition system for mobile devices," *IEEE Trans. Consum. Electron.*, vol. 58, no. 2, pp. 522—527, 2012.

[24] M. O. Derawi, B. Yang, and C. Busch, "Fingerprint recognition with embedded cameras on mobile phones," in *Proc. MobiSec, 2012*, 2012, p. 136–147.

[25] Y. Han *et al.*, "Embedded palmprint recognition system on mobile devices," in *Proc. Int. Conf. Biometrics*, 2007, pp. 1184–1193.

[26] A.-S. Ungureanu *et al.*, "Unconstrained palmprint as a smartphone biometric," *IEEE Trans. Consum. Electron.*, vol. 63, no. 3, pp. 334–342, Aug. 2017.

[27] A. Jain, K. Nandakumar, and A. Ross, "Score normalization in multimodal biometric systems," *Pattern Recognit.*, vol. 38, no. 12, pp. 2270–2285, Dec. 2005.

[28] P. Khaw, "Iris recognition technology for improved authentication," in *SANS*, 2002.

[29] K. R. Park *et al.*, "A study on iris localization and recognition on mobile phones," *EURASIP J. Adv.*, vol. 2008, no. 1, pp. 1–12, 2008.

[30] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *NIPS*, 2012, pp. 1097–1105.

[31] M. Wang and W. Deng, "Deep face recognition: A survey," *arXiv preprint arXiv:1804.06655v9*, 2020.

[32] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556v6*, 2015.

[33] K. He *et al.*, "Deep residual learning for image recognition," in *CVPR*, 2016, pp. 770–778.

[34] J. Hu *et al.*, "Squeeze-and-excitation networks," *arXiv preprint arXiv:1709.01507*, 2017.

[35] Y. Taigman *et al.*, "Deepface: Closing the gap to human-level performance in face verification," in *CVPR*, 2014, pp. 1701–1708.

[36] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Proc. CVPR*, 2015.

[37] O. M. Parkhi, A. Vedaldi, and A. Zisserman, "Deep face recognition," in *BMVC*, 2015.

[38] Y. Guo *et al.*, "Ms-celeb-1m: A dataset and benchmark for large-scale face recognition," in *ECCV*, 2016, pp. 87–102.

[39] Q. Cao *et al.*, "Vggface2: A dataset for recognising faces across pose and age," in *International Conference on Automatic Face and Gesture Recognition*, 2018.

[40] M. E. Fathy, V. M. Patel, and R. Chellappa, "Face-based active authentication on mobile devices," in *Proc. IEEE Int. Conf. Acoust., Speech and Signal Process. (ICASSP)*, 2015.

[41] U. Mahbub *et al.*, "Active user authentication for smartphones: A challenge data set and benchmark results," in *Proc. IEEE BTAS*, 2016, pp. 1–8.

[42] P. Samangouei, V. M. Patel, and R. Chellappa, "Facial attributes for active authentication on mobile devices," *Image Vis. Comput.*, vol. 58, pp. 181–192, Feb. 2017.

[43] C. McCool *et al.*, "Bi-modal person recognition on a mobile phone: using mobile phone data," in *Proc. IEEE ICME Workshop on Hot Topics in Mobile Mutlimedia*, 2012.

[44] D. Crouse *et al.*, "Continuous authentication of mobile user: Fusion of face image and inertial measurement unit data," in *Proc. Int. Conf. Biometrics*, May 2015, pp. 135–142.

[45] P. Perera and V. M. Patel, "Face-based multiple user active authentication on mobile devices," *IEEE Trans. Inf. Forensics Security*, vol. 14, no. 5, pp. 1240–1250, May 2019.

[46] M. E. ul Haq *et al.*, "Continuous authentication of smartphone users based on activity pattern recognition using passive mobile sensing," *J. Netw. Comput. Appl.*, vol. 109, no. C, pp. 1–12, 2018.

[47] Y. Li, H. Hu, and G. Zhou, "Using data augmentation in continuous authentication on smartphones," *IEEE Internet Things J.*, vol. 6, no. 1, pp. 628–640, 2019.

[48] W. Lee and R. B. Lee, "Multi-sensor authentication to improve smartphone security," in *Proc. ICISSP*, 2015, pp. 270–280.

[49] S. Amini *et al.*, "Deepauth: A framework for continuous user re-authentication in mobile apps," in *Proc. CIKM*, 2018, pp. 2027—-2035.

[50] Y. Zhang *et al.*, "Continuous authentication using eye movement response of implicit visual stimuli," in *Proc. IMWUT*, 2018.

[51] M. Tamviruzzaman *et al.*, "epet: When cellular phone learns to recognize its owner," in *Proc. SafeConfig 2009*, 2009, pp. 13–18.

[52] M. R. Hestbek, C. Nickel, and C. Busch, "Biometric gait recognition for mobile devices using wavelet transform and support vector machines," in *Proc. IWSSIP 2012*, 2012, pp. 205–210.

[53] K.-H. Yeh *et al.*, "I walk, therefore i am: Continuous user authentication with plantar biometrics," *IEEE Commun. Mag.*, vol. 56, no. 2, pp. 150–157, 2018.

[54] M. Muaaz and R. Mayrhofer, "Orientation independent cell phone based gait authentication," in *Proc. 12th Int. Conf. Advances Mobile Comput. Multimedia*, 2014, pp. 161–164.

[55] D. Gafurov and E. Snekkenes, "Gait recognition using wearable motion recording sensors," *EURASIP J. Adv. Signal Process*, vol. 2009, 2009.

[56] S. Mondal and P. Bours, "Swipe gesture based continuous authentication for mobile devices," in *Proc. ICB 2015*, 2015, pp. 458—465.

[57] T. Nohara and R. Uda, "Personal identification by flick input using self-organizing maps with acceleration sensor and gyroscope," in *Proc. 10th International Conference on Ubiquitous Information Management and Communication. ACM*, 2016, pp. 58—65.

[58] D.-H. Shih, C.-M. Lu, and M.-H. Shih, "A flick biometric authentication mechanism on mobile devices," in *Proc. ICCSS 2015*, 2015, pp. 31—-33.

[59] L. Lu and Y. Liu, "Safeguard: User reauthentication on smartphones via behavioral biometrics," *IEEE Trans. Comput. Soc. Syst.*, vol. 2, no. 3, pp. 53–64, 2015.

[60] H. Zhu *et al.*, "Shakein: Secure user authentication of smartphones with single-handed shakes," *IEEE Trans. Mobile Comput.*, vol. 16, no. 10, pp. 2901–2912, 2017.

[61] J. Fierrez *et al.*, "Benchmarking touchscreen biometrics for mobile authentication," *IEEE Trans. Inf. Forensics Security*, vol. 13, no. 10, pp. 2720–2733, Nov. 2018.

[62] M. Frank *et al.*, "Touchalytics: On the applicability of touchscreen input as a behavioral biometric for continuous authentication," *IEEE Trans. Inf. Forensics Security*, vol. 8, no. 1, pp. 136–148, Jan. 2013.

[63] M. Antal, Z. Bokor, and L. Z. Szabó, "Information revealed from scrolling interactions on mobile devices," *Pattern Recognit. Lett.*, vol. 56, pp. 7–13, Apr. 2015.

[64] A. Serwadda, V. V. Phoha, and Z. Wang, "Which verifiers work?: A benchmark evaluation of touch-based authentication algorithms," in *Proc. IEEE BTAS*, Sept. 2013, pp. 1–8.

[65] C. Shen *et al.*, "Performance analysis of touch-interaction behavior for active smartphone authentication," *IEEE Trans. Inf. Forensics Security*, vol. 11, no. 3, pp. 498–513, Mar. 2016.

[66] O. Miguel-Hurtado *et al.*, "Predicting sex as a soft-biometrics from device interaction swipe gestures," *Pattern Recogn. Lett.*, vol. 79, pp. 44–51, 2016.

[67] F. Anjomshoa *et al.*, "Social behaviometrics for personalized devices in the internet of things era," *IEEE Access*, vol. 5, pp. 12 199–12 213, 2017.

[68] F. Li *et al.*, "Behaviour profiling for transparent authentication for mobile devices," in *Proc. Eur. Conf. Inf. Warfare Security*, 2011, pp. 307—-314.

[69] A. Acien *et al.*, "Multilock: Mobile active authentication based on multiple biometric and behavioral patterns," *arXiv preprint arXiv:1901.10312v1*, 2019.

[70] C. Galdi *et al.*, "Exploring new authentication protocols for sensitive data protection on smartphones," *IEEE Commun. Mag.*, vol. 56, no. 2, pp. 136–142, 2018.

[71] M. Sultana, P. P. Paul, and M. L. Gavrilova, "Social behavioral information fusion in multimodal biometrics," *IEEE Trans. Syst., Man, Cybern.*, vol. 48, no. 12, pp. 2176–2187, 2018.

[72] M. M. Monwar and M. L. Gavrilova, "Multimodal biometric system using rank-level fusion approach," *IEEE Trans. Syst., Man, Cybern.*, vol. 39, no. 4, pp. 867–878, 2009.

[73] N. A. Fox *et al.*, "Robust biometric person identification using automatic classifier fusion of speech, mouth, and face experts," *IEEE Trans. Multimedia*, vol. 9, no. 4, pp. 701–714, 2007.

[74] P. P. Paul, M. L. Gavrilova, and R. Alhajj, "Decision fusion for multimodal biometrics using social network analysis," *IEEE Trans. Syst., Man, Cybern.*, vol. 44, no. 11, pp. 1522–1533, 2014.

[75] T. Zhu *et al.*, "Riskcog: Unobtrusive real-time user authentication on mobile devices in the wild," *IEEE Trans. Mobile Comput.*, vol. 19, no. 2, pp. 466–483, 2020.

[76] Y. Dodis, L. Reyzin, and A. Smith, "Fuzzy extractors: How to generate strong keys from biometrics and other noisy data," in *EUROCRYPT (LNCS, vol. 3027)*, 2004, p. 523–540.

[77] B. Chen and G. Wornell, "Quantization index modulation: A class of provably good methods for digital watermarking and information embedding," *IEEE Trans. Inform. Theory*, vol. 47, no. 4, p. 1423–1443, 2001.

[78] T. Ignatenko and F. M. J. Willems, "Information leakage in fuzzy commitment schemes," *IEEE Trans. Inf. Forensics Security*, vol. 5, no. 2, p. 337–348, 2010.

[79] S. Rane *et al.*, "Secure biometrics: Concepts, authentication architectures, and challenges," *IEEE Signal Processing Mag.*, vol. 30, no. 5, p. 51–64, 2013.

[80] I. Lagendijk, Z. Erkin, and M. Barni, "Encrypted signal processing for privacy protection," *IEEE Signal Processing Mag.*, vol. 30, no. 1, p. 82–105, 2013.

[81] S. Rane and P. Boufounos, "Privacy-preserving nearest neighbor methods," *IEEE Signal Processing Mag.*, vol. 30, no. 2, pp. 18–28, 2013.

[82] J. Bringer *et al.*, "An application of the goldwasser–micali cryptosystem to biometric authentication," in *Proc. Aust. Conf. Information Security and Privacy*, 2007, p. 96–106.

[83] A. C.-C. Yao, "How to generate and exchange secrets," in *Proc. SFCS 1986*, 1986, p. 162–167.

[84] N. K. Ratha *et al.*, "Generating cancelable fingerprint templates," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 4, pp. 561–572, 2007.

[85] N. Ratha, J. Connell, and R. Bolle, "Enhancing security and privacy in biometrics-based authentication systems," *IBM Syst. J.*, vol. 40, no. 3, p. 614–634, 2001.

[86] A. Kong *et al.*, "An analysis of biohashing and its variants," *Pattern Recognit.*, vol. 39, no. 7, p. 1359–1368, 2006.

[87] A. Teoh, T. Connie, and D. Ngo, "Remarks on biohash and its mathematical foundation," *Inform. Process. Lett.*, vol. 100, p. 145–150, 2006.

[88] S. W. Shin *et al.*, "Dictionary attack on functional transform-based cancelable fingerprint templates," *ETRI J.*, vol. 31, no. 5, p. 628–630, 2009.

[89] T. Bourlai, K. Messer, and J. Kittler, "Face verification system architecture using smart cards," in *Proc. ICPR 2004*, 2004, p. 793–796.

[90] A. Noore, "Highly robust biometric smart card," *Inform. Process. Lett.*, vol. 46, no. 4, p. 1059–1063, 2000.

[91] Y. Seto, "Development of personal authentication systems using fingerprint with smart cards and digital signature technologies," in *Proc. ICARCV 2002*, 2002, p. 996–1001.

[92] P. Y. Kumar and T. S. Ganesh, "Integration of smart card and gabor filter method based fingerprint matching for faster verification," in *Proc. Indicon 2005*, 2005, p. 526–529.

[93] S. Bistarelli, F. Santini, and A. Vaccarelli, "An asymmetric fingerprint matching algorithm for java card," *Pattern Anal. Applic.*, vol. 9, no. 4, pp. 359–376, 2006.

[94] N. Nedjaha *et al.*, "Efficient fingerprint matching on smart cards for high security and privacy in smart systems," *Inform. Sciences*, vol. 479, pp. 622–639, Apr. 2019.

[95] Y. Li, "Linear discriminant analysis and its application to face identification," PhD Thesis, School of Electronics and Physical Sciences, University of Surrey, Guildford, Surrey GU2 7XH, U.K., 2011.

[96] J. Kittler, Y. Li, and J. Matas, "Face authentication using client specific fisherfaces," *The Statistics of Directions, Shapes and Images*, p. 63–66, 1999.

[97] J. Czyz and L. Vandendorpe, "Evaluation of lda-based face verification with respect to available computational resources," in *Proc. Int'l Workshop on Pattern Recognition in Information Systems*, 2002.

[98] K. Lee and H. Byun, "A new face authentication system for memory-constrained devices," *IEEE Trans. Consum. Electron.*, vol. 49, no. 4, pp. 1214–1222, Nov. 2003.

[99] T. Bourlai, J. Kittler, and K. Messer, "On design and optimization of face verification systems that are smart-card based," *Mach. Vision Appl.*, vol. 21, no. 5, pp. 695–711, 2010.

[100] N. Nedjah *et al.*, "Efficient yet robust biometric iris matching on smart cards for data high security and privacy," *Future Gener. Comp. Sy.*, vol. 76, pp. 18–32, Nov. 2017.

[101] M. Sabri, M.-S. Moin, and F. Razzazi, "A new framework for match on card and match on host quality based multimodal biometric authentication," *J. Sign. Process Syst.*, vol. 91, no. 2, pp. 163–177, Feb. 2019.

[102] W.-Y. Choi *et al.*, "Svm-based speaker verification system for match-on-card and its hardware implementation," *ETRI*, vol. 28, no. 3, pp. 320–328, June 2006.

[103] R. D. Findling and R. Mayrhofer, "Mobile gait match-on-card authentication from acceleration data with offline-simplified models," in *Proc. MoMM 2016*, 2016, p. 250–260.

[104] L. Hong, Y. Wan, and A. Jain, "Fingerprint image enhancement: algorithm and performance evaluation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 8, pp. 777–789, Aug. 1998.

[105] S. Thavalengal, P. Bigioi, and P. Corcoran, "Iris authentication in handheld devices – considerations for constraint-free acquisition," *IEEE Trans. Consum. Electron.*, vol. 61, no. 2, pp. 245–253, May 2015.

[106] M. Muaaz and R. Mayrhofer, "Smartphone-based gait recognition: From authentication to imitation," *IEEE Trans. Mobile Comput.*, vol. 16, no. 11, pp. 3209–3221, Nov. 2017.

[107] P. M. Corcoran, "Biometrics and consumer electronics: A brave new world or the road to dystopia?" *IEEE Consum. Electron. Mag.*, vol. 2, no. 2, pp. 22–33, Apr. 2013.

[108] GlobalPlatform, "Secure element access control version 1.1," Sep. 2014. [Online]. Available: https://globalplatform.org/specs-library/secure-element-access-control-v1-1/

[109] ——, "Secure channel protocol '11' card specification v2.3 – amendment f version 1.2.1," Mar. 2019. [Online]. Available: https://globalplatform.org/specs-library/

[110] S. B. Pan *et al.*, "An ultra-low memory fingerprint matching algorithm and its implementation on a 32-bit smart card," *IEEE Trans. Consum. Electron.*, vol. 49, no. 2, pp. 453–459, May 2003.

[111] N. Nedjah, R. S. Wyant, and L. M. Mourelle, "Efficient biometric palm-print matching on smart-cards for high security and privacy," *Multimed Tools Appl*, vol. 76, pp. 22 671–22 701, Nov. 2017.

[112] B. Jacob *et al.*, "Quantization and training of neural networks for efficient integer-arithmetic-only inference," in *Proc. CVPR*, 2018.

[113] S. Gupta *et al.*, "Deep learning with limited numerical precision," in *Proc. ICML'15*, Feb. 2015, pp. 1737–1746.

[114] C. M. Bishop, *Pattern Recognition and Machine Learning*. New York, NY, USA: Springer-Verlag, 2006.

[115] M. Courbariaux, Y. Bengio, and J.-P. David, "Binaryconnect: Training deep neural networks with binary weights during propagations," in *Proc. NIPS*, Nov. 2015, pp. 3123–3131.

[116] C. Sanderson and K. K. Paliwal, "Information fusion and person verification using speech & face information," IDIAP Research Institute, Research Paper, IDIAP Research Report 02-33, 2002.

[117] A. Aho and J. Ullman, *Principles of Compiler Design*. Addison-Wesley, 1977.

[118] J. W. Davidson and S. Jinturkar, "An aggressive approach to loop unrolling," Dept. Comput. Sci., Univ. Virginia, Charlottesville, VA 22903 U. S. A., Tech. Rep., 1995.

[119] R. Rifkin and A. Klautau, "In defense of one-vs-all classification," *J. Machine Learning Research*, vol. 5, pp. 101–141, Jan. 2004.

[120] Y. Sun, A. K. C. Wong, and M. S. Kamel, "Classification of imbalanced data: A review," *Int. J. Pattern Recogn.*, vol. 23, no. 4, pp. 687–719, 2009.

[121] M. Oestreicher, "Transactions in java card," in *Proc. 15th Annual Computer Security Applications Conference*, 1999, pp. 291–298.

[122] GSM-Association, "Rsp technical specification version 2.2.1," Dec. 2018. [Online]. Available: https://www.gsma.com/newsroom/wp-content/uploads//SGP.22-v2.2.1-2.pdf

[123] C. Shen *et al.*, "Performance analysis of multi-motion sensor behavior for active smartphone authentication," *IEEE Trans. Inf. Forensics Security*, vol. 13, no. 1, pp. 48–62, Jan. 2018.

[124] Z. Syed *et al.*, "Touch gesture-based authentication on mobile devices: The effects of user posture, device size, configuration, and inter-session variability," *J. Syst. Software*, vol. 149, pp. 158–173, Mar. 2019.

[125] Statista, "Global market share held by the leading smartphone operating systems in sales to end users from 1st quarter 2009 to 2nd quarter 2018," 2018. [Online]. Available: https://www.statista.com/statistics/266136/global-market-share-held-by-smartphone-operating-systems/

[126] GSM-Association, "Remote provisioning architecture for embedded uicc technical specification version 3.2," Jun. 2017. [Online]. Available: https://www.gsma.com/newsroom/wp-content/uploads/SGP.02_v3.2_updated.pdf

[127] B. Amos, B. Ludwiczuk, and M. Satyanarayanan, "Openface: A general-purpose face recognition library with mobile applications," CMU-CS-16-118, CMU School of Computer Science, Tech. Rep., 2016.

[128] M. Wang and W. Deng, "Deep face recognition: A survey," Feb. 2019. [Online]. Available: https://arxiv.org/abs/1804.06655

[129] Samsung, "The market frontrunner for sim/esim." [Online]. Available: https://www.samsung.com/semiconductor/security/sim-esim/

[130] C. M. Bishop, *Pattern Recognition and Machine Learning*. New York, NY, USA: Springer, 2007.

[131] C. Cortes and V. Vapnik, "Support-vector networks," *Mach. Learning*, vol. 20, pp. 273–297, Sept. 1995.

[132] R. A. Fisher, "The use of multiple measurements in taxonomic problems," *Annals of Eugenics*, vol. 7, no. 2, pp. 179–188, 1936.

[133] S.-H. Cha, "Comprehensive survey on distance/similarity measures between probability density functions," *Int'l J. Math. Models and Methods in Applied Sciences*, vol. 1, no. 4, pp. 300–307, 2007.

[134] K. E. Atkinson, *An Introduction to Numerical Analysis*, 2nd ed. John Wiley & Sons, 1988.

[135] Support-Apple, "Transfer an esim from your previous iphone to your new iphone." [Online]. Available: https://support.apple.com/en-us/HT210655

[136] Amazon, "Amazon web services." [Online]. Available: https://aws.amazon.com/

[137] N. Koblitz, "Elliptic curve cryptosystems," *Math. Comput.*, vol. 48, no. 177, pp. 203–209, 1987.

[138] K. Zhang *et al.*, "Joint face detection and alignment using multitask cascaded convolutional networks," *IEEE Signal Processing Letters*, vol. 23, no. 10, pp. 1499–1503, Oct 2016.

[139] TensorFlow-API, "tf.lite.interpreter." [Online]. Available: https://www.tensorflow.org/api_docs/python/tf/lite/Interpreter

[140] Google, "Deploy machine learning models on mobile and iot devices." [Online]. Available: https://www.tensorflow.org/lite

[141] Android-Developers, "Telephonymanager." [Online]. Available: https://developer.android.com/reference/android/telephony/TelephonyManager

[142] GlobalPlatform, "Secure element access control v1.1," Oct. 2014. [Online]. Available: https://globalplatform.org/specs-library/secure-element-access-control-v1-1/

[143] P. J. Grother and W. J. Salamon, "Minex ii performance of fingerprint match-on-card algorithms - phase ii/iii report interagency report 7477 (revision i)," NIST, Tech. Rep., May 2009.

[144] A. W. Appel and J. Palsberg, *Modern Compiler Implementation in Java*, 2nd ed. Cambridge University Press, Oct. 2002.