Mobile Match-on-Card Authentication Using Offline-Simplified Models with Gait and Face Biometrics

Rainhard Dieter Findling, Michael Hölzl, René Mayrhofer

Abstract—Biometrics have become important for mobile authentication, e.g. to unlock devices before using them. One way to protect biometric information stored on mobile devices from disclosure is using embedded smart cards (SCs) with biometric match-on-card (MOC) approaches. However, computational restrictions of SCs also limit biometric matching procedures. We present a mobile MOC approach that uses offline training to obtain authentication models with a simplistic internal representation in the final trained state, wherefore we adapt features and model representation to enable their usage on SCs. The pre-trained model can be shipped with SCs on mobile devices without requiring retraining to enroll users. We apply our approach to acceleration based mobile gait authentication as well as face authentication and compare authentication accuracy and computation time of 16 and 32 bit Java Card SCs. Using 16 instead of 32 bit SCs has little impact on authentication performance and is faster due to less data transfer and computations on the SC. Results indicate 11.4% and 2.4-5.4% EER for gait respectively face authentication, with transmission and computation durations on SCs in the range of 2 s respectively 1 s. To the best of our knowledge this work represents the first practical approach towards acceleration based gait MOC authentication.

Index Terms—Mobile Computing, Authentication, Smart cards, Gait biometrics, Face biometrics,

1 Introduction

Biometric authentication, such as fingerprint, gait, or voice authentication [2] becomes increasingly available and popular on mobile devices as device unlocking mechanism. In contrast to classic, knowledge based mobile authentication approaches like PIN, password, or graphical pattern [3], user biometrics cannot easily be changed by users in case they are disclosed. Consequently, leakage or theft of biometric information has severe consequences: attackers could e.g. reconstruct original biometrics from obtained information and use it for replay attacks [4]. Usage of reconstructed biometrics beyond the associated mobile device might be possible too, as they are self-evidently the same across all systems they are used with (cf. [5], [6], [7], [8]). Further, in contrast to desktop computers, mobile devices are more easily lost, stolen, or accessed by attackers without being noticed. This further increases the risk of biometric information stored and processed on mobile devices to fall into hands of attackers.

Consequently, biometrics processed and stored on mobile devices require adequate protection. One approach is by using smart cards (SC) [9], which are often shipped in off-the-shelf mobile devices in the form of secure elements (SEs). These can either be directly embedded in the phone hardware, extended with an SD card, or provided within modern SIM cards [10]. With biometrics on SCs, the storage and matching part can either be achieved with template-on-card (TOC) or match-on-card (MOC) techniques (cf. [2], [11], [12], [13], [14]). With TOC, biometric templates of the user are recorded by sensors of the mobile device and stored on the smart card during enrollment. During authentication the enrolled templates are fetched from the SC and compared with new recordings outside the SC. In contrast, with MOC authentication, new recordings are transfered to the SC and compared with previously stored templates directly on the SC.

This leads to the following noticeable differences of MOC over TOC: on the one hand, after a user's biometric templates have been stored on the SC during enrollment, they never leave the SC. Hence, MOC reduces the possibilities for leakage or theft of biometric templates over TOC. On the other hand, comparing users' biometric templates with new biometric recordings on the SC is subject to hardware limitations of the SC, namely transfer bandwidth to and computational limitations on the SC. Hence, the portion of data that can be transfered to the SC and the computations that can be done on the SC have to be selected carefully. As reducing the risk of leakage or theft of biometric templates is important, MOC is

- *Rainhard Dieter Findling is with the Ambient Intelligence Group, Department of Communications and Networking (COMNET), Aalto University, Finland. All authors are with u'smile, the Josef-Ressel-Center for User-Friendly Secure Mobile Environments, at the University of Applied Sciences Upper Austria, and the Institute of Networks and Security (INS), Johannes Kepler University Linz (JKU), Austria.*
  E-mail: rainhard.findling@fh-hagenberg.at
- *A preliminary version of this work was published in MoMM 2016 [1] which is extended to cover different smart card architectures and biometrics, features an extended discussion of related work focusing on features and matching approaches with respect to applicability in match-on-card approaches, and is evaluated using gait and face biometrics on 16 and 32 bit smart cards.*

regularly preferred over TOC, despite the accompanying computational limitations. In turn, these limitations lead to restrictions in how existing MOC approaches are frequently designed (cf. [2], [11], [14], [15], [16], [17]):

- MOC approaches usually rely on restricted operations and logic for matching templates with new recordings. Hence, they often do not utilize regular, offline trained machine learning (ML) models. Further, they are frequently restricted to a small set of – sometimes handpicked – features to be used in the matching process. Both necessarily limit the MOC discriminative power.
- To reduce computational requirements, most MOC operations are very domain specific. The underlying mechanisms are usually strongly adapted to the used biometrics. This impedes the adaption of new biometrics in MOC approaches, where it would1 be beneficial to have reusable concepts for feature extraction, model representation, and matching operations.

To address these restrictions, we aim for enabling a more generic usage of simple ML models on SCs, which are computed offline with sufficient computational power and do not need to be retrained during enrollment of individual users. The challenge therein lies with the mentioned limitations, which imply restrictions in how biometric features and ML models can be calculated and represented for usage on SCs. We therefore propose a scheme which trains and generates ML models offline (e.g. using server infrastructure), then uses the simplified internal structure of trained models on SCs in the matching process (Fig. 1).

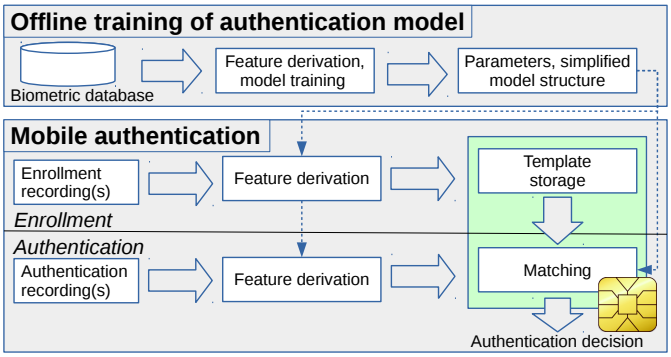


Fig. 1. Conceptual overview of the proposed approach. The SC is highlighted in green.

Models suitable for this approach are those where the internal structure translates to a simple representation in the final and fully trained state (e.g. an equation). In contrast to matching on the SC, the offline training, evaluation, and selection necessary to obtain this structure in the first place can be arbitrarily complex. After obtaining such a model offline, both features and models need to be adapted to suit SC restrictions. This includes data types of features and models, as well as computations using those. Note that it is desirable to integrate necessary adaption to features and models already in the offline modeling process. Doing so allows for more precise estimation of authentication performance, which is in turn important for model tuning and selecting a reasonable model and model configuration for usage on SCs. Consequently, both offline and on-device processing rely on identical preprocessing and feature extraction. Further, note that feature extraction up to feature simplification can be performed outside the SC. This allows for more complex and powerful feature extraction while not compromising any information previously stored on the SC.

In this paper, we demonstrate the proposed approach on acceleration based gait biometrics as well as face biometrics, using SCs restricted to either 16 or 32 bit range integer calculations. We transform features derived from biometric recordings and model structure used on the SC to be represented in half of the integer range available on the SC. This allows for multiplications within the available integer range. We demonstrate that adequate MOC authentication is still feasible using limited bit representation of the obtained model, stored biometric template, and new biometric recording. Summarizing, our contributions are:

- We present a generic approach towards biometric MOC authentication, wherefore we adapt both offline trained ML models and features to enable their computation and handling on SCs.
- We apply our approach to face authentication and acceleration based gait authentication as examples of biometrics with usually more complex matching and bigger templates. To the best of our knowledge this is the first practical approach to gait MOC authentication with acceleration data.
- We evaluate the feasibility and performance of our approach with publicly available data sets, using both 16 and 32 bit Java Card SCs. We thereby achieve 11.4% and 2.4-5.4% EER for gait respectively face authentication, while staying in the range of 2 s respectively 1 s for transmission and calculation durations on SCs.

2 Background on Smart Cards

Smart cards (SC) such as secure elements (SE) used in mobile devices, are special integrated circuits which provide certain characteristics that are useful for security sensitive applications: a) cryptographic operations (e.g. encryption, decryption, hashing) can be performed directly on the chip, often in hardware. b) SCs are intentionally kept small and less complex to make unintended behavior/bugs in the system less likely. That is, it is easier to verify that there are no major security flaws. c) data and application code in the memory is protected against unauthorized access and tampering. A serial interface, which is controlled by the operating system of the hardware, is the only way to access this data.

However, besides those advantageous characteristics, SCs also bring limitations that need to be considered for

applications relying on them: a) data transfer to/from SCs being restricted in bandwidth (cf. Hölzl et al. [10] with measurements of 329 B/s for contactless and 3,31 kB/s for contact cards). b) while some modern SCs already use a 32 bit architecture, many currently deployed cards are still based on a 16 bit architectures. That is, there are no 4 byte integers and integer calculations in hardware on those cards. c) persistent and volatile memory are highly limited with a maximum capacity of around 1 MB for current cards. d) finally, SCs are limited in computation capabilities: for example, there are no native floating point operations available in hardware. Computations performed in software are considerably slower than on PCs or mobile devices due to clock rate of SCs usually being in the MHz range.

These computation and data transfer limitations affect both the internal structure of authentication models and number and type of features that can be used with SCs. For example, using 4 byte integers in a 16 bit environment requires more complex data structures in internal computations (i.e. operations on arrays for simple multiplications). Hence, using small value ranges for both model representation and features transferred to the SC are preferred. Further, transmission bandwidth to/from the SC is limited, which limits the amount of data that can reasonably be sent to the SC during user authentication. In this paper, we consider all these limitations in the design of the biometric matching algorithm. We show that it is feasible to overcome the disadvantages of SCs and make use of their advantageous characteristics in a generic way.

3 Related Work

To this date, fingerprints are the best researched biometrics with MOC card authentication approaches. They usually utilize small templates, thereby a small amount of features (mostly minutiae based), which in turn leads to relatively simple matching procedures (cf. [11], [17], [18], [19]). MOC authentication with biometrics other than fingerprints has been covered by little research. Examples include Choi et al. [15], which use support vector machines (SVM) with a limited amount of features and FPGAs for speaker verification in a MOC manner. Czajka et al. [13] perform iris recognition by deriving a 1024 bit iris code from samples outside the SC, then match new recordings with enrolled templates on the card using a computationally lightweight Hamming distance. This approach is therefore more similar to fingerprint than e.g. face authentication in terms of template size. Another authentication related example is human identification from CCTV records [20]. Although the approach is conceptually similar to gait authentication from visual data (including the matching based on simple distance metrics), the processing chain, including used features such as cloth color and human height, represent a major difference.

3.1 Gait Authentication for SCs

Mobile gait authentication [21] can be based on different types of data, including visually [22] or floor sensed information [23], as well as information from sensors worn by humans themselves [24]. With the latter, different sensor types and sensor positions on the human body have been utilized [25], where mobile devices like smartphones have become a powerful source of such data. They usually feature a number of different sensors and are frequently with people while they are walking (e.g. inside a trousers pocket). Especially accelerometers shipped with mobile phones have been used for acceleration based gait authentication [26]. As human walk is of cyclic nature, each step can be seen as repetitive cycle. For acceleration preprocessing, both cycle and window based approaches have been utilized in literature [27]. With cycle based approaches – which we focus on in our gait MOC authentication approach – individual step cycles are segmented from recordings and used for subsequent recognition. Analogously, with window based approaches, a (possibly fixed length) sliding window is used on recordings instead to segment data chunks.

The matching procedure of acceleration based gait authentication often involves Dynamic Time Warping (DTW) as distance metric between two time series [28], [29], [30]. For two time series of length m and n, regular DTW brings a memory complexity of at minimum $m \cdot n$, which renders it unfeasible for usage on regular SCs. Though there exist some effective approaches to reduce the computational complexity of DTW (thereby also restricting its warping power), such as the Sakaboi-Chiba band [31], [32], even most limited DTW approaches are difficult to calculate on SCs.

For acceleration based gait authentication without using SCs and DTW, various features have been used. Those include: average, median, min, max, standard deviation (SD), and median absolute deviation (MAD) acceleration of individual axes and their magnitude [21], [33], root mean square (RMS) acceleration [33], mean- and zero-crossings [33], principal component coefficients of acceleration [34], [35], binned acceleration distribution [21], [24], [33], time between peaks [21], discrete cosine and fast Fourier transformation coefficients [36], [37], [38], [39], and Mel- and Bark-frequency cepstral coefficients [27], [33]. Further, wavelet transformations have been used with non-cycle-based acceleration gait data [27], [40] and floor sensor based gait data [41], as well as on acceleration based gait style recognition [42], which in contrast to gait identification or authentication does not distinguish individuals but gait styles. On those features, again a number of non-DTW based models have been applied, including cross-correlation based [43] or tree based models [21], artificial neural networks (ANN) [21], [44], support vector machines [33], [35], analysis of variance (ANOVA) [36], Gaussian mixture models (GMM) [38], and hidden Markov models (HMM) [33].

To the best of our knowledge there exist no approaches

to acceleration based gait MOC authentication yet. With the majority of the approaches described above, either retraining the model for individual users would be required, or neither training the model, nor using a ready trained model to predict new samples is feasible on SCs with respect to their computation requirements. Still, similar feature extraction mechanisms can be utilized in our approach as long as they are computed outside the SC.

3.2 Face Authentication for SCs

As with mobile gait authentication, mobile face authentication is possible with off-the-shelf mobile devices, as it only relies on a regular camera being available with the device. In terms of 2D face authentication outside SCs, both geometry and appearance based approaches have been used [45]. Geometric approaches derive facial features and key positions in face images, then decide on recognition or authentication using this information. In contrast, appearance based approaches – which we focus on in our face MOC authentication approach – derive features directly from the pixel representation of face images without considering facial features directly. In the past, a considerable amount of appearance based face recognition and authentication approaches has been discussed (cf. [46], [47], [48], [49]). Important examples include Eigenfaces [50], based on which further simple yet effective dimensionality transformation and reduction approaches have been proposed for face recognition and authentication, such as Linear Discriminant Analysis (LDA) [51] or Fisherfaces [52]. Further approaches additionally employ other models, such as neural networks [53] or support vector machines [54], or different appearance based feature extraction procedures, such as local binary pattern [55] or wavelet transformation and related approaches [56].

Previous face MOC authentication mostly relies on using limited matching on the SC. For example, Tistarelli et al. [57] propose a face authentication TOC approach in which they use morphological filtering and adaptive template matching to extract the position of relevant facial features for matching. During matching they fetch enrolled templates from the card and compare them to new recordings using a space-variant approach based on Principal Component Analysis (PCA). Lee and Bun [58] combine PCA projection weights, average intensity and edge values as features with genetic algorithms (GA) for feature selection. They thereby largely reduce the amount of features, which enables the usage of an SVM model for authentication.

Kittler et al. [59] state that PCA compresses templates in a suboptimal way for usage on SC. They therefore propose a MOC approach using a 1D client specific LDA, of which they utilize the distance of new recordings to both the stored client template and to the average impostor to derive a scalar distance measure. As tradeoff between computational requirements and authentication performance, Bourlai et al. [60] utilize the client specific LDA proposed in [59] as feature extraction mechanism, then use the vector dot product of a new recording and the enrolled template with a predefined threshold to obtain an authentication decision. While this approach has some commonalities with our MOC approach – such as using LDA, a linear combination, and threshold for authentication decision – both approaches rely on different core mechanics. a) we do not use samples such as faces directly, but distances between samples to distinguish between comparisons of samples of the same person from those of different people. As we only train our model once offline we can ship the pre-trained model with SCs on mobile devices. This allows enrolling new users without requiring any retraining, while the enrollment of one user is still completely independent of the enrollment of other users. b) with a client specific LDA, the distance to the client template is combined with the distance to the mean of impostors in a one dimensional way. In contrast, we use our model and multi-dimensional distances between a new sample and the reference template to derive an authentication decision. c) we perform feature extraction outside the SC. This prevents computing features for the enrolled template on the SC for each authentication attempt as done in [60] and allows for computationally more intensive operations during feature extraction in general. The downside is that this prevents exchanging feature extraction for existing templates at a later point in time.

Summarizing, in contrast to previous work on face MOC authentication, our approach utilizes the distances between samples to distinguish between comparisons of samples from the same person and those of different people. We can further ship the pre-trained model with SCs on mobile devices without requiring any retraining to enroll of users.

4 Threat Model

Biometrics require adequate protection because users cannot change them as easily as e.g. their passwords in case of disclosure. This is well known for strong biometrics such as face, fingerprint, or iris. With these obtaining a single biometric template might be sufficient to recreate the biometric information required for authentication and perform a spoofing attack [61]. In contrast, templates from weak/soft biometrics (e.g. behavioral biometrics) usually do not represent such a reliable basis for attacks. Recreating the information required for authentication is usually non-trivial and requires a sophisticated attacker, such as with acceleration based gait authentication. From obtained templates, attackers need to artificially reproduce the acceleration to be sensed by the device, e.g. by using a machine that accelerates the device according to the template. Simply walking like the legitimate subject has been shown to be unfeasible [62]. The high effort of such attacks reduces the probability, that they are actually performed successfully. However,

we need to assume that yet unknown ways of performing spoofing attacks with behavioral biometrics might emerge in the future. As they could simply use templates that have been unprotected and disclosed in the past, both strong and weak biometrics should be protected alike.

There are two main attack vectors for obtaining biometric information from mobile devices: attackers obtaining physical control over a device at a certain point in time, and attackers running malicious software/trojans on the mobile device for a certain duration. Assume attackers obtain physical access to a user's mobile device (thereby access to data stored on it) after the user has enrolled. If no SC is used to store biometrics, attackers could extract stored templates. This is possible even if the device comes under attackers' control long after enrollment or the last authentication of the legitimate user. If a TOC approach is used, attackers are required to trigger an authentication attempt for templates to be fetched from the SC. This further requires attackers to monitor the device memory/processor to obtain biometric information, but still works without interaction of the legitimate user.

If a MOC approach is used, attackers cannot directly extract stored biometric information as it never leaves the SC. Therefore, templates can only be obtained while they are processed outside the SC (from sensors up to the SC) – either during enrollment or authentication of the legitimate user. This implies that attackers need to access and manipulate a user's device unnoticed, then need to wait until the legitimate user enrolls or authenticates to obtain biometric information (they cannot freely choose timing anymore). Such attacks likely require attackers to run malicious software on the mobile device, which connects them to the second main attack vector. Attackers having online access and live eavesdropping capabilities (e.g. using malicious software/trojans) could monitor sensors and memory to obtain biometric information directly when it is processed. Protection against such attacks requires securing/hardening the whole processing chain from sensors up to the matching procedure and authentication decision. One approach is to combine MOC with a trusted execution environment (TEE, e.g. ARM TrustZone[1]) that protects information from sensors up to the SC. Another approach is to combine all steps in an all-in-one piece of hardware, which is referred to as system-on-card (SOC). Within the latter, MOC represents the essential part of matching biometric samples. This is why both the combination of MOC with a TEE as well as SOC can be seen as a superset of MOC. Consequently, providing generic and widely applicable mobile MOC approaches is an essential part of fully protecting biometric information on mobile devices from attackers with live eavesdropping capabilities.

Our work is a first step towards the long-term goal of protecting mobile biometrics in a transparent and well evaluated way. For the first time it combines a MOC approach, generic matching concepts, and biometrics with traditionally bigger, therefore more challenging templates (such as facial images and gait cycles compared to e.g. fingerprints). This is why we purely focus on the MOC aspect in this article and, for the time being, declare malicious software/trojan attack vectors on the sensor data processing pipeline out of scope for this paper. Further attacks on the security of SCs themselves, such as side-channel attacks by Kocher et al. [63] or Vermoen et al. [64], which try to extract the biometric template from the SC itself, are also defined out of scope.

1. ARM Trust-Zone: http://www.arm.com/products/processors/technologies/trustzone/

5 METHOD

Our approach is divided into offline model generation and usage of the obtained model for enrollment and authentication on the mobile device. Both parts share steps for preprocessing, feature extraction, and feature simplification (Fig. 2). The offline part determines the parametrization which is then applied on mobile devices alike. On the mobile device those steps are done outside the SC, which thereby allows for computationally more complex operations or operations specific to certain biometrics. Based on preprocessed biometric samples, offline computation trains an authentication model, simplifies it, applies feature selection, and finally estimates the resulting authentication performance. The obtained model is stored on the SC integrated in mobile devices, which then performs the MOC operation using stored samples and newly recorded samples. Therefore, no (re)training of the model is required in order to enroll new users.

5.1 Offline Model Creation

With B bit SCs, integer operations within B bit range are done in hardware, therefore are fast. We consequently strive to keep computations on SCs within this range. More specifically, we use a linear model on the SC, which internally computes a result using a linear combination of feature vector and model slope vector. We therefore adapt features and model slope so that their linear combination is possible within B bit range on the SC.

On the one hand those simplifications lead to faster computations. On the other hand they also lead to a more coarse resolution of the feature space. For example: the feature space of 10 features expressed in 8 bit is limited to $2^{8^{10}} = 1.21 \cdot 10^{24}$ possibilities, which corresponds to a theoretical maximum entropy of 80 bit. Expressing the same features in 16 bit results in twice the theoretical maximum entropy of 160 bit[2]. One could assume that using less information in features and models (due to

2. Due to the uneven distribution of biometrics in feature space, biometric approaches are usually unable to exploit the full feature space [65]. Hence, depending on the used biometrics and features, the resulting true entropy is necessarily smaller than this theoretical boundary.

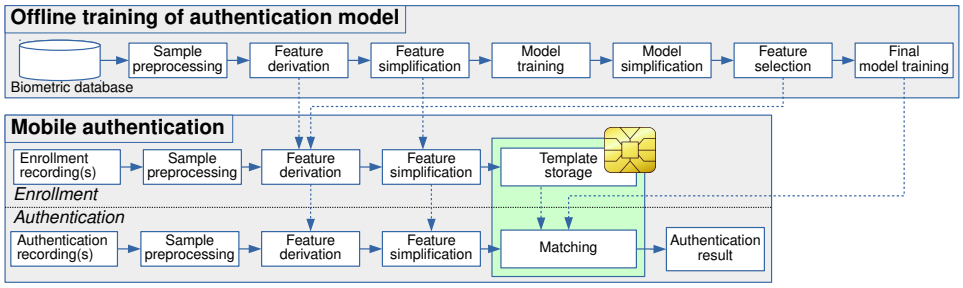


Fig. 2. The offline part of our approach computes and simplifies an authentication model, then selects the most important features to be used on mobile devices. On mobile devices, our approach uses the determined parameters and model to perform MOC authentication. The SC is highlighted in green.

using 16 instead of 32 bit SCs) would reduce the subsequent authentication accuracy. However, our evaluation indicates the impact to be negligible.

5.1.1 Feature Simplification

To work with B bit integer space SCs, we transform (scale, shift, and round) original real-valued features to fit $\frac{B}{2}$ bit integer range. The transformation uses a vector of features $\vec{f_o}$ that contains one individual feature from all samples in offline training data, then utilizes its mean and standard deviation (SD) for transformation (Eq. 1). The transformation applied to an original feature might result in values that are bigger or smaller than the $\frac{B}{2}$ bit space, which we cap at the boundaries (Eq. 2). This ensures that the $\frac{B}{2}$ bit space can be optimally used for the mainstream data, while boundaries are respected also for new, unseen data with potential outliers. The transformed vector of features $\vec{f_t}$ therefore consists of values in the range $[0, 2^{\frac{B}{2}} - 1]$, e.g. for 16 bit space the range of $[0, 255]$. This transformation is applied to all features.

$$\vec{f_r} = \text{round}\left(\frac{\vec{f_o} - mean(\vec{f_o})}{2 \cdot SD(\vec{f_o})}\right) \cdot \frac{(2^{\frac{B}{2}-1} - 1) + (2^{\frac{B}{2}-1} - 1)}{} \tag{1}$$

$$\vec{f_t} = \begin{cases} 0 & \text{for } \vec{f_r} < 0 \\ 2^{\frac{B}{2}} - 1 & \text{for } \vec{f_r} > 2^{\frac{B}{2}} - 1 \\ \vec{f_r} & \text{else} \end{cases} \tag{2}$$

On mobile devices, the same feature preprocessing and simplification transformation is applied to features of new recordings during enrollment and authentication. Therefore, the mean and SD per feature computed from offline training data are stored on mobile devices outside the SC[3]. After simplifying features, the obtained simplified biometrics feature vectors are handed to the SC for purpose of enrollment or authentication.

3. Due to subsequent feature selection only a subset of those features remain. Storing and performing the simplification is only done for actually used features.

5.1.2 Model Training

Offline model training uses pairs of samples represented by their feature vectors. At first, the distance between two biometric feature vectors $\vec{v_1}$ and $\vec{v_2}$ yields an absolute distance vector $d(\vec{v_1}, \vec{v_2})$ of same length, also in $\frac{B}{2}$ bit representation (Eq. 3).

$$d(\vec{v_1}, \vec{v_2}) = |\vec{v_1} - \vec{v_2}| \tag{3}$$

We refer to feature distance vectors originated by the same person as being of the positive class P and to those originated by different people as being of the negative class N. Using feature distance vectors from our offline training data we create a classification model able to distinguish between the P and N class (for details on how data partitioning is done for model training and evaluation see Sec. 6). The obtained model can then be used on the mobile device to decide if a new feature distance vector is a P or N sample.

As classification model we use an LDA model [66]. In contrast to the previously utilized [1] Generalized Linear Model (GLM) [67], LDA aims to maximize the P-N inter-class-distance and minimize the P and N intra-class-distances of samples. Therefore, LDA models can usually provide for better class separation over GLM models. However, as both models are linear models, in their ready trained state both can internally be represented by a slope $\vec{s_o}$ (model coefficients) and an additional intercept I (offset to the origin of the coordinate system). For a distance vector $\vec{d}$ from a template and a new recording, those are used to predict the class membership C_d using a linear combination (Eq. 4).

$$C_d = \begin{cases} P & \text{for } \sum_i \vec{s_o} \odot \vec{d} < I \\ N & \text{else} \end{cases} \tag{4}$$

Such linear combinations are simple enough to be computed on a SC, which is a core reason for choosing this model type. From training we obtain the optimal slope, intercept, and threshold – which are later used to predict the class of new samples in both an offline evaluation of our approach as well as the application case of on-device authentication.

5.1.3 Model Simplification

The slope $\vec{s_o}$ and intercept I obtained from model training are real-valued and, similar to biometric features, have to be simplified to enable their usage on a B bit integer SC. We therefore scale original model coefficients $\vec{s_o}$ to optimally fit an $\frac{B}{2}$ bit space and apply a cap at boundaries, resulting in a transformed slope $\vec{s_t}$ (Eq. 5 and 6). In contrast to transforming biometric features (Eq. 1), no shift is applied. This would otherwise change the meaning of coefficients, as coefficients around 0 have less influence on the result than those with higher absolute values.

$$\vec{s_r} = \text{round}\left(\frac{\vec{s_o}}{2 \cdot SD(\vec{s_o})}\right) \cdot (2^{\frac{B}{2}-1} - 1) \tag{5}$$

$$\vec{s_t} = \begin{cases} -(2^{\frac{B}{2}-1} - 1) & \text{for } \vec{s_r} < -(2^{\frac{B}{2}-1} - 1) \\ +(2^{\frac{B}{2}-1} - 1) & \text{for } \vec{s_r} > +(2^{\frac{B}{2}-1} - 1) \\ \vec{s_r} & \text{else} \end{cases} \tag{6}$$

Having both feature distance vectors and the slope in $\frac{B}{2}$ bit integer representation now allows for their piecewise multiplication on SCs in B bit integer range (see Sec. 5.2). Therefore, this can be done efficiently on SCs that only support calculations in B bit integer range in hardware.

5.1.4 Feature Selection

After model training, features that are associated to small coefficients necessarily have small influence on the output – hence both feature and coefficient can possibly be removed without severely influencing classification performance. As selection criteria we thereby use the strongest absolute coefficient c_{max} as reference: a coefficient c_i is selected if it fulfills $c_i \geq \alpha \cdot c_{\text{max}}$, with α in the range $[0, 1]$. For details on used thresholds α and number of selected features for individual biometrics see Section 6.

By performing feature selection we achieve reduced storage requirements and computations on the SC, as well as reduced features to transfer to the SC, which therefore reduces the overall SC processing duration. Another, smaller advantage is that relying on stronger features could slightly increase overall predictive power of the model. However, as small coefficients do not necessarily denote features completely unimportant for separating classes, prediction capabilities might as well be slightly reduced by doing so.

5.2 Mobile Device: Enrollment and Authentication

Preparation of mobile devices comprises storing the feature normalization and simplification parameters on the mobile device, as well as storing the model (slope and intercept) directly on the SC. After data recording, enrollment and authentication perform data preprocessing, feature extraction, and feature simplification as stated in Sec. 5.1. On mobile devices those can be done outside the SC, as they do not use any information about templates previously stored on the SC. For enrollment, m feature vectors – derived from m newly recorded biometric samples – are transferred to the SC, where they are stored in the enrolled template for later usage. No further calculations are done on the SC. For authentication, n feature vectors from n newly recorded biometric samples are transferred to the SC. As this latter transmission is done for each authentication attempt, the transfer period is important and measured in our evaluation in Sec. 6.

On the SC we perform $m \cdot n$ comparisons between all m stored reference samples and all n newly transmitted samples using the stored, offline-computed model. To keep those $m \cdot n$ linear combination within a range of B bit (especially during summing intermediate, piecewise products of slope and difference vector), we utilize the mean value instead of a sum. Hence each intermediate product is immediately divided by the length of the slope vector to predict the class C_d (Eq. 7).

$$C_d = \begin{cases} P & \text{for } \sum_i \left(\frac{\vec{s_{t,i}} \cdot \vec{d_i}}{\text{length}(\vec{s_t})}\right) < I \\ N & \text{else} \end{cases} \tag{7}$$

The resulting $m \cdot n$ predictions, each indicating P or N class, are treated as votes. We compute a final, binary authentication decision from them, which is handed from the SC to the mobile device to authorize or deny an authentication attempt. If we would instead hand an authentication probability from the SC to the mobile device, this would conceptually allow for more flexible feedback to users. The downside of doing so is the danger of enabling hill climbing attacks to unlock the system or deriving information about users' biometrics (cf. [8], [68], [69], [70]), which is why we yield only binary authentication decisions from the SC.

Besides allowing for linear combination in hardware on B bit SCs, our approach has the advantage of requiring only $(i + 2) \cdot \frac{B}{2}$ bits of storage memory on a SC for the model, when using i features. For example, with 16 bit SCs, a model for 10 features could be expressed in only 12 bytes of SC storage. Similarly, m samples in an enrollment template require only $m \cdot i \cdot \frac{B}{2}$ bits of storage. For example, with 16 bit SCs, 8 samples consisting of 75 features require only 600 byte of SC storage.

6 Evaluation

We evaluate our approach on 16 and 32 bit SCs with face and gait biometrics, measuring both SC computation duration and authentication performance. We use a 16 bit JCOP 2.4.1 SC with 80 kB EEPROM memory running Java Card version 2.2.2 and a 32 bit SIM-card with 1 MB non-volatile memory and Java Card version 3.0.1. Communication is done over the contact interfaces of these cards using the same card reader. To compare the authentication accuracy of our approach with non-simplified features and models, we also evaluated the original, real valued features and models on the same data.

6.1 Duration on Smart Cards

The duration of transferring one sample with 75 features to the SC and yielding an authentication decision back was measured to be on average 31.5 ms (SD=0.14 ms) with 16 bit SCs and 16.7 ms (SD=0.08 ms) with 32 bit SCs. This duration excludes computations on the SC and scales linearly with the amount of samples sent. Computing our complete approach on SCs also shows a nearly linear increase of computation time over both number of samples in the enrolled template and number of features per sample (Fig. 3). Those calculations include the computation of distances between samples in the enrolled template stored on the SC with newly transmitted samples, the linear combination of distances with model parameters determined offline, the voting of individual results to obtain an authentication decision, and the yielding thereof.

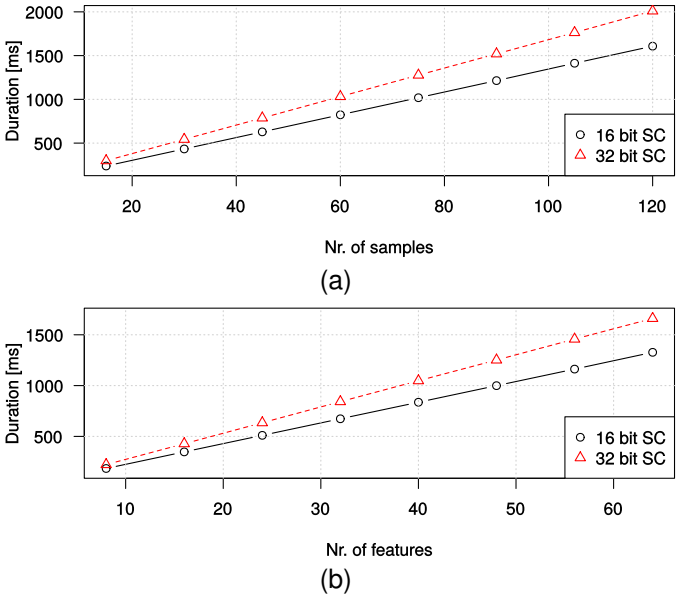


Fig. 3. Average duration of our approach on 16 and 32 bit SCs, including transmissions, for a) different number of samples in the enrolled template, using 75 features per sample, and b) different number of features per sample, using 32 samples in the enrolled template.

In absolute numbers, data transmission time becomes negligible compared to computation time on the SC. This implies that changing the number of samples m in the enrolled template and number of samples n in the new recording has little impact if the number of total votes $m \cdot n$ is unaffected. With using $m \cdot n = 64$ we achieve an average computation time of 1608 ms and 2010 ms for 16 and 32 bit SCs, and 824 ms and 1032 ms when using $m \cdot n = 32$ instead. The increased duration for 32 bit SCs has two reasons: a) twice the amount of data needs to be transmitted due to samples containing twice the amount of information as compared to 16 bit SCs. b) the amount of data that can be sent in one query is limited to 255 bytes by the transmission protocol of the SC (cf. application protocol data units (APDU) in [71]). Consequently, one 16 bit feature is transferred as two separate bytes, of which conversion to one 16 bit short on the SC requires additional time. While this limitation could be overcome by using the extended version of the protocol (extended length fields in [71]), in our measurements we consider the short and therefore slower variant for interoperability with all currently deployed smarts cards.

6.2 Evaluation Setup for Using Different Biometrics

To obtain realistic authentication performance estimates of people unseen by the model during training, we perform a non-overlapping, 50/50 population independent split [72] on the corresponding datasets. We thereby assign 50% of participants to the training partition, which is used for training the model, and 50% of participants to the test partition, which is only used once for estimating the performance of the chosen and trained final model on yet unseen people. We further use only training data to determine parameters for feature extraction, simplification, and selection, then use the determined parameters to transform test data the same way. Within both training and test partition we use all combinations of different samples originated by the same person to obtain P distances and all combinations of samples originated by different people (within the corresponding partition) to obtain N distances.

The training partition is used to train and evaluate different parametrizations of our model to find a suitable configuration for distinguishing between P and N distances. As training and evaluation procedure we thereby use well established 10-fold cross validation with 10 repetitions and report the fit as Receiver Operating Characteristics (ROC) curve, Area Under the ROC Curve (AUC), and Equal Error Rate (EER). After an optimal parametrization has been found (i.e. minimal coefficient threshold α and number of votes $m \cdot n$), the model is trained again using this configuration and all training data. The resulting model is evaluated once on the test partition to obtain a realistic authentication performance estimate on data of yet unseen people. For this we report the resulting True Positive Rate (TPR) and True Negative Rate (TNR). For comparability we additionally also report the ROC curve, AUC, and EER, when using all parametrization determined from training on the test partition, except the final decision threshold.

The resulting model further serves as basis for voting when using multiple biometric samples in both template stored on the SC and new recordings for authentication. Thereby, m cycles are contained in the enrolled template and n new recordings are provided during authentication – which results in a total of $m \cdot n$ samples and votes. For tuning the voting approach we use the same data partitions, with the training partition being used to evaluate the authentication performance of different amount of votes. Then, test data is again used only once for estimating the authentication performance for the

final, voting based authentication model on data of yet unseen people.

6.3 Evaluation with Gait Biometrics

For evaluating our approach with gait biometrics we utilize cycle based gait authentication based on acceleration data recorded by off-the-shelf mobile devices. In contrast to previous research we use a MOC approach, a non-DTW based model, and combine features previously used in acceleration gait recognition with features from other domains.

6.3.1 Gait Data Source

For our evaluation we use the gait data set of Muaaz and Mayrhofer [73] which contains 3D acceleration recordings of 35 people, each walking about 550 m in total. The data was recorded with off-the-shelf smartphones featuring 100 Hz 3D accelerometers, with phones being placed realistically in trousers pockets. Further, for each participant, recording was split into two sessions with a gap of on average 25 days between recording, which allows for realistic cross-day evaluations of gait authentication systems. From this data we utilize cross-day, left-pocket recordings of all participants to train and evaluate our approach with gait biometrics.

6.3.2 Gait Data Preprocessing and Feature Extraction

Preprocessing mechanisms are adapted from Nickel [33] as well as Muaaz and Mayrhofer [73], [74], which comprise of walking detection and preprocessing, as well as subsequent step detection and preprocessing, which we briefly summarize here. From 3D acceleration recordings, we extract walking segments with y-axis acceleration variance above $0.8\frac{m}{s^2}$ for at least 10 s. To compensate for gravity, we remove the mean acceleration segment and axis, then compute the resulting acceleration magnitude. As acceleration sampling is not necessarily uniform, we further perform a linear interpolation to obtain a uniform sampling rate of 100 Hz. For noise reduction we apply a Savitzky-Golay filter [75] with window length of 150 ms and polynomial of 1st order. The core advantage of this filter over frequently used running mean or median filters is the better retaining of the original signal shape.

For step cycle segmentation, reference cycles are extracted from each walking segment, around the middle of the segment [73]. Those are used to determine previous and successive starts of cycles in the same walking segment, which in turn are segmented into individual gait cycle samples of the corresponding individual. Furthermore, those are linearly interpolated to a uniform length of 100 acceleration values each, which correspond to a duration of 1 s at a 100 Hz sampling rate. Cycles that diverge largely from the majority of extracted cycles are further defined as outliers and discarded. For that purpose we compute the normalized DTW distance[4] between all n cycles and discard those cycles for which more than $\frac{n}{2}$ distances are above a predefined threshold of 0.6. The remaining gait cycles are used in feature extraction and subsequently handed to the SC for enrollment or authentication (Fig. 4).

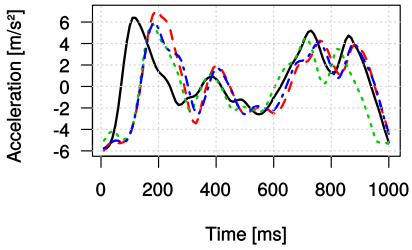


Fig. 4. Examples of preprocessed gait cycles with a uniform length of 1 s, consisting of 100 values each.

For each preprocessed cycle we derive a number of features. In the time domain we utilize the mean, median, SD, median absolute deviation (MAD), and autocorrelation (AC) series with a maximum shift of 100 values as features on one cycle. AC has been used as signal preprocessing in other biometric recognition tasks, such as electrocardiography recognition [76], but to our knowledge not yet in acceleration based gait authentication. To reduce naturally existing inter-feature correlation of the resulting AC feature vector, we use only every third value as feature. With a sampling rate of 100 Hz this corresponds to a shift granularity of 30 ms. In the frequency domain we compute the fast Fourier transformation (FFT) of the cycle. As human body motion sensed by accelerometers usually yield usable information in the frequency range of about 0-20 Hz (cf. [77], [78], [79]), we use both frequency power and phase in this range as features. Frequency power and phase are added as separate features to a) avoid passing complex values to models and b) enable separately treating them (e.g. normalizing and discarding features individually). Additionally, we also compute a discrete wavelet transform (DWT) representation of a cycle using a multiresolution analysis of 6 levels. As wavelet we utilize a least asymmetric Daubechies wavelet [80] of length 8. As with FFT features, all wavelet features are treated as individual features too. In total we thereby obtain a feature vector of length 177, which we can reduce to 64 features for both 16 and 32 bit SCs using a feature selection coefficient threshold of $\alpha = 0.35$. Therefore, with gait data our approach requires 66/132 bytes of storage (for 16/32 bit SCs) for the offline computed model and 64/128 bytes per gait cycle in the enrolled template. With 8 cycles in the template this leads to a total of 578/1156 bytes of storage requirement on the SC.

6.3.3 Gait Model Training and Authentication Results

Due to slightly different amounts of gait cycles being discarded per participant during preprocessing and data cleaning, preprocessing results in a total of 2132 and 1943 unique gait cycles in the training and test partition,

4. This DTW distance calculation is done for data cleaning purposes outside the SC, consequently is not related to the authentication model and matching procedure on the SC.

respectively. Due to the size of the training partition and the resulting training complexity, we use a random subset of 100000 P and 150000 N distances for training the model. However, for intra-training evaluation of trained models, the full training partition size is utilized (Tab. 1).

Partition	Cycles	P	N
Training	2 132	174 410	2 207 243
Test, pop. independent	1 943	168 976	2 158 427

TABLE 1
Gait biometrics: training and test partition sizes, as amount of gait cycles and the resulting amount of P and N comparisons.

Gait evaluation results indicate a test partition EER of about 0.21 when using a single gait cycle in both enrolled template and new recording for authentication (Tab. 2 and Fig. 5). When using 64 comparisons instead (e.g. 8 samples in both enrolled template and new recording), we achieve an EER of about 0.114. Results differ only marginally between 16 and 32 bit SCs, or using the original, real valued features and models.

Partition	Votes	SC	AUC	EER	TPR	TNR
Training	1	16 bit	0.892	0.179	–	–
Training	1	32 bit	0.892	0.179	–	–
Training	1	real v.	0.892	0.179	–	–
Test	1	16 bit	0.868	0.210	0.787	0.780
Test	1	32 bit	0.867	0.207	0.787	0.797
Test	1	real v.	0.867	0.208	0.787	0.797
Training	64	16 bit	0.927	0.123	–	–
Training	64	32 bit	0.928	0.123	–	–
Training	64	real v.	0.927	0.123	–	–
Test	64	16 bit	0.963	0.114	0.958	0.809
Test	64	32 bit	0.963	0.114	0.959	0.810
Test	64	real v.	0.962	0.115	0.952	0.809

TABLE 2
Gait evaluation results for using a single gait cycle in both the template and the new recording and a total of 64 votes (e.g. 8 templates and 8 new recordings to compare to).

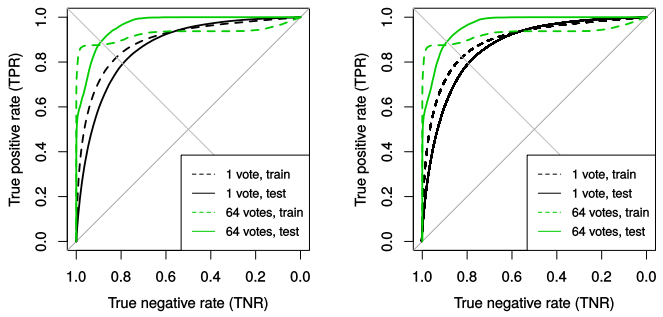


Fig. 5. ROC curves for using a single gait cycle in both the template and the new recording and a total of 64 votes (e.g. 8 templates and 8 new recordings to compare to).

These results indicate that for acceleration based gait data, increasing the granularity of model coefficient and feature space (as required for usage of our approach on 16 bit SCs) does not lead to considerably worse results over using 32 bit SCs – where the resolution of data is allowed to be twice as fine – or even real valued features and models. Using the feature space available with 16 bit features and model coefficients on 32 bit SCs results in longer durations, caused by higher feature precision and the corresponding higher total amount of data transferred and processed. Further, our results seem comparable with findings from previous research on the same dataset, with 18% EER when comparing single gait cycles [29] and 94% TNR and 64% TPR when using 4 gait cycles in one comparison [73]. However, in contrast to our work, those approaches utilize a computationally intensive DTW, and do not use SCs to protect gait biometrics.

6.4 Evaluation with Face Biometrics

For demonstrating our approach with face biometrics, we use a view-based face authentication approach based on 2D wavelet transformed representations of face images and estimate the authentication performance with two publicly available face databases.

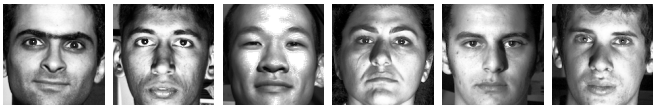
6.4.1 Face Data Source

To demonstrate our approach on face biometrics we use subsets of the Yale-B [81] and the Panshot Face Unlock Database [82]. The Yale-B database contains facial images illuminated with a light source from different azimuths and elevations relative to the face. We thereby utilize face images with maximum azimuth and elevation of $\pm 20°$ between light source and face, which results in a database subset 511 facial images of 27 participants. In contrast, the Panshot Face Unlock database contains face images recorded from 9 different perspectives in a $180°$ semi circle around the head using different recording hardware. We thereby utilize facial images recorded from a frontal perspective, which results in a total of 600 images of 30 different participants. For both databases, we use grayscale, unsegmented (neither face-detected nor cropped) images, then perform face detection and segmentation ourselves to obtain faces realistic for a mobile authentication scenario.

6.4.2 Face Data Preprocessing and Feature Extraction

At first we equalize the image histogram per image, then perform Viola and Jones face detection [83] to detect and segment the part of the image related to facial information into quadratic images. We only consider the face image if its diagonal is at least $\frac{1}{4}$ the diagonal of the original image. In mobile face authentication scenarios, where users are within arms reach of their mobile device, requiring such a relative minimal face image size effectively prevents a large portion of potential false positive face detections. Further, if multiple faces are detected, we only consider the biggest detection. We again equalize the histogram per face image. Equalization results are different than

before face segmentation, as background information that contributed to the equalization has now been removed from the images (Fig. 6).



(a) Yale-B face database

(b) Panshot Face Unlock database

Fig. 6. Examples of preprocessed, segmented, and equalized face images from the Yale-B and Panshot Face Unlock databases handed to feature extraction.

Before deriving features, we downscale images to reduce processing power required in subsequent steps on mobile devices and SCs. In preliminary experiments we used face image sizes of 64×64 and 32×32, in which the latter turned out to be sufficient for subsequent feature extraction and MOC face authentication. We therefore used face images of size 32×32 – but our approach could be applied analogously to other image sizes as well. As feature extraction we use 2D discrete wavelet transformation (2D-DWT) and multiresolution analysis with a Daubechies Least-Asymmetric 2D Wavelet [80]. The resulting coefficients are treated as feature vector of length 1365, which can be reduced to 75 features (16 bit SC), respectively 72 features (32 bit SC), using a maximum feature coefficient threshold $\alpha = 0.95$. Therefore, with face biometrics our approach requires 77/148 bytes for storing the model (with 16/32 bit SCs) and 75/144 bytes per face in the enrolled template. With 8 face images in the template this leads to a total storage requirement of 677/1354 bytes.

6.4.3 Face Model Training and Authentication Results

Due to slightly different amounts of faces detected per participant we obtain slightly different training and test partitions for both databases (Tab. 3).

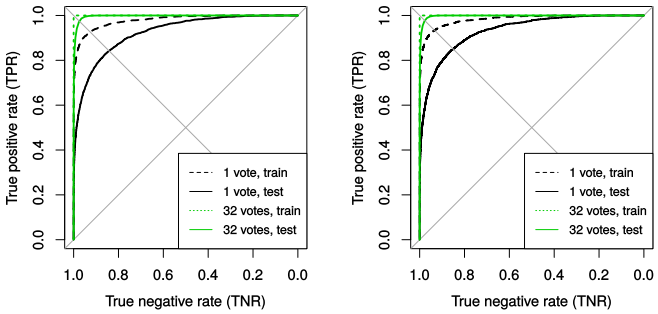
Database	Partition	Faces	P	N
Yale-B	Training	265	2 376	32 604
Yale-B	Test	246	2 205	27 930
Panshot	Training	296	2 780	40 880
Panshot	Test	273	2 536	34 592

TABLE 3
Face biometrics: training and test partition sizes, as amount of face images and the resulting amount of P and N comparisons.

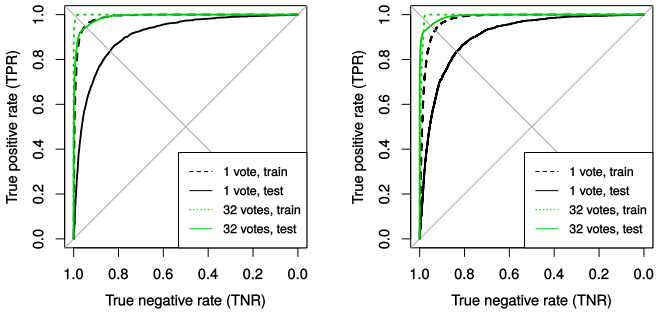
Similar to the results of the gait based evaluation, authentication performance differs only marginally between 16 and 32 bit SCs, or using the original, real valued features and models (Tab. 4 and Fig. 7). Using the Yale-

Database	Partition	Votes	SC	AUC	EER	TPR	TNR
Yale-B	Training	1	16 bit	0.980	0.075	–	–
Yale-B	Training	1	32 bit	0.983	0.067	–	–
Yale-B	Training	1	real v.	0.983	0.067	–	–
Yale-B	Test	1	16 bit	0.925	0.159	0.890	0.775
Yale-B	Test	1	32 bit	0.932	0.150	0.900	0.784
Yale-B	Test	1	real v.	0.932	0.150	0.900	0.784
Yale-B	Training	32	16 bit	1.000	1.000	–	–
Yale-B	Training	32	32 bit	1.000	1.000	–	–
Yale-B	Training	32	real v.	1.000	1.000	–	–
Yale-B	Test	32	16 bit	0.997	0.030	0.998	0.933
Yale-B	Test	32	32 bit	0.998	0.024	0.996	0.954
Yale-B	Test	32	real v.	0.998	0.025	1.000	0.921
Panshot	Training	1	16 bit	0.987	0.051	–	–
Panshot	Training	1	32 bit	0.977	0.070	–	–
Panshot	Training	1	real v.	0.977	0.070	–	–
Panshot	Test	1	16 bit	0.909	0.163	0.754	0.892
Panshot	Test	1	32 bit	0.907	0.164	0.748	0.885
Panshot	Test	1	real v.	0.906	0.164	0.748	0.885
Panshot	Training	32	16 bit	0.999	0.012	–	–
Panshot	Training	32	32 bit	0.995	0.022	–	–
Panshot	Training	32	real v.	0.994	0.022	–	–
Panshot	Test	32	16 bit	0.990	0.054	0.792	0.992
Panshot	Test	32	32 bit	0.993	0.053	0.797	0.999
Panshot	Test	32	real v.	0.993	0.053	0.797	0.999

TABLE 4
Face evaluation results for using a single face image in both the template and the new recording and a total of 32 votes (e.g. 8 templates and 4 new recording to compare to) for training and test partitions.



(a) Yale-B, 16 bit SC (b) Yale-B, 32 bit SC



(c) Panshot, 16 bit SC (d) Panshot, 32 bit SC

Fig. 7. ROC curves for using a single face image in both the template and the new recording and a total of 32 votes (e.g. 8 templates and 4 new recordings to compare to) for training and test partitions.

B database we obtain a test partition EER between 15-16% without majority voting of comparisons of multiple face images. Additionally employing a majority vote boosts results to 2.4-3% EER. Using a 32 instead of 16 bit SC marginally increases the overall authentication performance, visible in both decreased EER and increased AUC. Using the Panshot Face Unlock database, we obtain a slightly worse test partition performance of 16.3% EER without majority voting, which is decreased to 5.3-5.4% EER using majority voting. We assume that results being worse is due to the Panshot Face Unlock database containing faces with less distinctive features recorded more uniformly, which makes distinguishing them more difficult. Overall, results confirm that our approach is also applicable to both types of SCs with facial biometrics. Similar to gait results, the gain of using a 32 instead of 16 bit SC is minimal with face biometrics. Therefore, using the increased resolution of feature space and model coefficients available with 32 bit SCs seems unnecessary, as it primarily leads to an increased duration of our approach due to bigger amounts of data transferred and processed.

Summarizing, our approach achieves 11.4% EER for gait MOC authentication using majority voting over 64 comparisons and 2.4-5.4% EER for face MOC authentication using majority voting over 32 comparisons. Our approach is faster when using 16 instead of 32 bit SCs, while the increased granularity of feature and model coefficient space with 16 bit SCs seems to have little negative impact on authentication performance. Therefore, the granularity available with 16 bit SCs seems sufficient to adequately represent gait and face biometrics with the utilized features. The total duration of our approach on SCs is in the range of 2 s with gait and 1 s with face authentication (including data transmissions and overall slightly faster on 16 than 32 bit SCs). We argue this duration to be a reasonable trade-off between authentication performance and delay, as responsiveness will usually be more critical for face than gait authentication. This is because gait authentication can be performed passively and continuously. In contrast to face authentication, users would therefore not actively perceive delays from an ongoing authentication.

7 Conclusions

In this article we present an approach towards match-on-card (MOC) authentication on mobile devices that uses models created from offline machine learning. We use model types that feature a simple internal representation once they are fully trained. To enable their usage on SCs, we adapt and simplify both used features and models. The model is computed only once using a dataset of the corresponding biometrics, then stored on SCs of mobile devices. Enrollment on mobile devices involves recording samples of the authorized user and storing their feature vectors on SCs without requiring retraining the model. Authentication compares features of newly recorded samples with enrolled samples on the SC, using the previously stored model to derive a binary authentication decision. One major advantage of the proposed approach is that it can be applied on different biometrics alike, thereby facilitating the translation of mobile biometric matching procedures towards MOC in general.

We applied our approach to acceleration based mobile gait authentication as well as face authentication, utilizing both 16 and 32 bit Java Card SCs. With gait authentication, when using 8 cycles in the enrolled template and 8 newly recorded cycles for authentication, we found our approach to be feasible with an EER of 11.4%. Authentication time on the SC stays in the range of 2 s, including data transmissions and authentication computation. To the best of our knowledge this work represents the first practical approach towards acceleration based gait MOC authentication. With face authentication, when using 8 face images in the enrolled template and 4 newly recorded face images for authentication, we found our approach to be feasible with an EER of 2.4-5.4% EER. The authentication time on the SC thereby stays in the range of 1 s, again including both transmission and calculation time on SCs. Using 16 instead of 32 bit SCs seems to have little negative impact on authentication performance. From this we derive that an adequate representation of samples and models is possible in the more granular feature and model coefficient space on 16 bit SCs. Furthermore, using the higher resolution of information of 32 bit SCs leads to more data being transferred and more computations on SCs, which overall make the approach slower than on 16 bit SCs.

Summarizing, these results indicate that our approach for generic mobile MOC authentication is feasible with different biometrics on both 16 and 32 bit SCs. In the future, this work might thereby facilitate the transfer further mobile biometrics toward using MOC techniques.

Acknowledgments

This work has partially been carried out within the scope of *u'smile*, the Josef Ressel Center for User-Friendly Secure Mobile Environments, funded by the Christian Doppler Gesellschaft, A1 Telekom Austria AG, Drei-Banken-EDV GmbH, LG Nexera Business Solutions AG, NXP Semiconductors Austria GmbH, and Österreichische Staatsdruckerei GmbH.

References

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

[2] A. K. Jain, A. A. Ross, and K. Nandakumar, *Introduction to Biometrics*. Springer, 2011.

[3] D. Van Bruggen, S. Liu, M. Kajzer, A. Striegel, C. R. Crowell, and J. D'Arcy, "Modifying smartphone user locking behavior," in *Proc. SOUPS 2013*. NY, USA: ACM, 2013, pp. 10:1–10:14. [Online]. Available: http://doi.acm.org/10.1145/2501604.2501614

[4] K. Cao and A. Jain, "Learning fingerprint reconstruction: From minutiae to image," *IEEE Information Forensics and Security*, vol. 10, no. 1, pp. 104–117, Jan. 2015.

[5] A. K. Jain and K. Nandakumar, "Biometric authentication: System security and user privacy." *IEEE Computer*, vol. 45, no. 11, pp. 87–92, 2012.

[6] D. C. L. Ngo, A. B. J. Teoh, and J. Hu, *Biometric Security*. Cambridge Scholars Publishing, 2015.

[7] S. Prabhakar, S. Pankanti, and A. Jain, "Biometric recognition: security and privacy concerns," *IEEE Security & Privacy*, vol. 1, no. 2, pp. 33–42, Mar. 2003.

[8] U. Uludag and A. K. Jain, "Attacks on biometric systems: a case study in fingerprints," in *Security, Steganography, and Watermarking of Multimedia Contents VI*, ser. Society of Photo-Optical Instrumentation Engineers (SPIE), vol. 5306, Jun. 2004, pp. 622–633.

[9] W. Rankl and W. Effing, *Smart Card Handbook*. Wiley, 2004.

[10] M. Hölzl, R. Mayrhofer, and M. Roland, "Requirements for an open ecosystem for embedded tamper resistant hardware on mobile devices," in *Proc. MoMM 2013*. ACM, 2013, p. 249.

[11] S. Bistarelli, F. Santini, and A. Vaccarelli, "An asymmetric fingerprint matching algorithm for Java Card TM," *Pattern Analysis and Applications*, vol. 9, no. 4, pp. 359–376, 2006.

[12] J. Bringer, H. Chabanne, D. Le Métayer, and R. Lescuyer, "Privacy by design in practice: Reasoning about privacy properties of biometric system architectures," in *Formal Methods*. Springer, 2015, vol. 9109, pp. 90–107.

[13] A. Czajka, P. Strzelczyk, M. Chochowski, and A. Pacut, "Iris recognition with match-on-card," in *Proc. EUSIPCO 2007*, Poznan, Poland, Sep. 2007, pp. 189–192.

[14] A. K. Jain, K. Nandakumar, and A. Nagar, "Biometric template security," *EURASIP Advances in Signal Processing*, vol. 2008, pp. 113:1–113:17, Jan. 2008.

[15] W.-Y. Choi, D. Ahn, S. B. Pan, K. I. Chung, Y. Chung, and S.-H. Chung, "SVM-based speaker verification system for match-on-card and its hardware implementation," *ETRI*, vol. 28, no. 3, pp. 320–328, Jun. 2006.

[16] M. Govan and T. Buggy, "A computationally efficient fingerprint matching algorithm for implementation on smartcards," in *Proc. BTAS 2007*, Sep. 2007, pp. 1–6.

[17] S. B. Pan, D. Moon, Y. Gil, D. Ahn, and Y. Chung, "An ultra-low memory fingerprint matching algorithm and its implementation on a 32-bit smart card," *IEEE Consumer Electronics*, vol. 49, no. 2, pp. 453–459, May 2003.

[18] S. Garcia-Salicetti, C. Beumier, G. Chollet, B. Dorizzi, J. L. l. Jardins, J. Lunter, Y. Ni, and D. Petrovska-Delacrétaz, *BIOMET: A Multimodal Person Authentication Database Including Face, Voice, Fingerprint, Hand and Signature Modalities*. Berlin, Heidelberg: Springer, 2003, pp. 845–853.

[19] P. Grother, W. Salamon, C. Watson, M. Indovina, and P. Flanagan, "MINEX II: Performance of fingerprint match-on-card algorithms phase II / III report. NIST interagency report 7477 (rev. I)," Information Access Division, National Institute of Standards and Technology (NIST), Tech. Rep., May 2009.

[20] H. M. Moon, C. Won, and S. B. Pan, "The multi-modal human identification based on smartcard in video surveillance system," in *Proc. IEEE/ACM GreenCom and CPSCom 2010*, Dec. 2010, pp. 691–698.

[21] J. R. Kwapisz, G. M. Weiss, and S. A. Moore, "Cell phone-based biometric identification," in *Proc. BTAS 2010*, Sep. 2010, pp. 1–7.

[22] S. Sarkar, P. J. Phillips, Z. Liu, I. R. Vega, P. Grother, and K. W. Bowyer, "The humanid gait challenge problem: data sets, performance, and analysis," *IEEE TPAMI*, vol. 27, no. 2, pp. 162–177, Feb. 2005.

[23] L. Middleton, A. A. Buss, A. Bazin, and M. S. Nixon, "A floor sensor system for gait recognition," in *Proc. IEEE AutoID 2005*, Oct. 2005, pp. 171–176.

[24] D. Gafurov, E. Snekkenes, and P. Bours, "Gait authentication and identification using wearable accelerometer sensor," in *Automatic Identification Advanced Technologies*, Jun. 2007, pp. 220–225.

[25] D. Gafurov and E. Snekkenes, "Gait recognition using wearable motion recording sensors," *EURASIP Advances in Signal Processing*, vol. 2009, pp. 7:1–7:16, Jan. 2009.

[26] M. Tamviruzzaman, S. I. Ahamed, C. S. Hasan, and C. O'brien, "ePet: When cellular phone learns to recognize its owner," in *Proc. SafeConfig 2009*. NY, USA: ACM, 2009, pp. 13–18.

[27] 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*, Apr. 2012, pp. 205–210.

[28] A. Mannini and A. M. Sabatini, "Machine learning methods for classifying human physical activity from on-body accelerometers," *Sensors*, vol. 10, no. 2, pp. 1154–1175, 2010.

[29] M. Muaaz and R. Mayrhofer, "Orientation independent cell phone based gait authentication," in *Proc. MoMM 2014*. NY, USA: ACM, 2014, pp. 161–164.

[30] X. Wang, Y. Li, and F. Qiao, "Gait authentication based on multi-criterion model of acceleration features," in *Proc. ICMIC 2010*, Jul. 2010, pp. 664–669.

[31] V. Niennattrakul and C. A. Ratanamahatana, "Learning DTW global constraint for time series classification," *CoRR*, vol. abs/0903.0041, 2009. [Online]. Available: http://arxiv.org/abs/0903.0041

[32] H. Sakoe and S. Chiba, "Dynamic programming algorithm optimization for spoken word recognition," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 26, no. 1, pp. 43–49, Feb. 1978.

[33] C. Nickel, "Accelerometer-based biometric gait recognition for authentication on smartphones," Ph.D. dissertation, Technische Universität Darmstadt, 2012.

[34] P. Bours and R. Shrestha, "Eigensteps: A giant leap for gait recognition," in *Proc. IWSCN 2010*, May 2010, pp. 1–6.

[35] S. Sprager and D. Zazula, "A cumulant-based method for gait identification using accelerometer data with principal component analysis and support vector machine," *WSEAS Transactions on Signal Processing*, vol. 5, no. 11, pp. 369–378, Nov. 2009.

[36] M. Bächlin, J. Schumm, D. Roggen, and G. Töster, *Quantifying Gait Similarity: User Authentication and Real-World Challenge*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 1040–1049.

[37] D. Gafurov, "Performance and security analysis of gait-based user authentication," Ph.D. dissertation, Faculty of Mathematics and Natural Sciences at the University of Oslo, 2008.

[38] R. K. Ibrahim, E. Ambikairajah, B. Celler, N. H. Lovell, and L. Kilmartin, "Gait patterns classification using spectral features," in *Proc. ISSC 2008*, Jun. 2008, pp. 98–102.

[39] L. Rong, Z. Jianzhong, L. Ming, and H. Xiangfeng, "A wearable acceleration sensor system for gait recognition," in *Proc. Industrial Electronics and Applications*, May 2007, pp. 2654–2659.

[40] S. Rahati, R. Moravejian, and F. M. Kazemi, "Gait recognition using wavelet transform," in *Proc. ITNG 2008*, Apr. 2008, pp. 932–936.

[41] A. Mostayed, S. Kim, M. M. G. Mazumder, and S. J. Park, "Foot step based person identification using histogram similarity and wavelet decomposition," in *Proc. ISA 2008*, Apr. 2008, pp. 307–311.

[42] T. Iso and K. Yamazaki, "Gait analyzer based on a cell phone with a single three-axis accelerometer," in *Pro. MobileHCI 2006*. NY, USA: ACM, 2006, pp. 141–144.

[43] J. Mäntyjärvi, M. Lindholm, E. Vildjiounaite, S.-M. Mäkelä, and H. Ailisto, "Identifying users of portable devices from gait pattern with accelerometers," in *Proc. ICASSP 2005*, vol. 2, 2005, pp. ii–973.

[44] E. S. Sazonov, T. Bumpus, S. Zeigler, and S. Marocco, "Classification of plantar pressure and heel acceleration patterns using neural networks," in *Proc. Neural Networks 2005*, vol. 5, Jul. 2005, pp. 3007–3010.

[45] S. G. Kong, J. Heo, B. R. Abidi, J. Paik, and M. A. Abidi, "Recent advances in visual and infrared face recognitiona review," *Computer Vision and Image Understanding*, vol. 97, no. 1, pp. 103–135, 2005.

[46] A. F. Abate, M. Nappi, D. Riccio, and G. Sabatino, "2D and 3D face recognition: A survey," *Pattern Recognition Letters*, vol. 28, no. 14, pp. 1885–1906, Oct. 2007.

[47] K. W. Bowyer, K. Chang, and P. Flynn, "A survey of approaches and challenges in 3D and multi-modal 3D + 2D face recognition," *Computer Vision and Image Understanding*, vol. 101, no. 1, pp. 1 – 15, 2006.

[48] X. Zhang and Y. Gao, "Face recognition across pose: A review," *Pattern Recognition*, vol. 42, no. 11, pp. 2876–2896, Nov. 2009.

[49] X. Zou, J. Kittler, and K. Messer, "Illumination invariant face recognition: A survey," in *Proc. BTAS 2007*, Sep. 2007, pp. 1–8.

[50] L. Sirovich and M. Kirby, "Low-Dimensional Procedure for the Characterization of Human Faces," *Journal of the Optical Society of America A*, vol. 4, no. 3, pp. 519–524, 1987.

[51] J. Lu, K. N. Plataniotis, and A. N. Venetsanopoulos, "Face recognition using lda-based algorithms," *IEEE Transactions on Neural Networks*, vol. 14, no. 1, pp. 195–200, Jan. 2003.

[52] P. N. Belhumeur, J. a. P. Hespanha, and D. J. Kriegman, "Eigenfaces vs. fisherfaces: Recognition using class specific linear projection," *IEEE TPAMI*, vol. 19, no. 7, pp. 711–720, Jul. 1997.

[53] M. J. Er, S. Wu, J. Lu, and H. L. Toh, "Face recognition with radial basis function (rbf) neural networks," *IEEE Transactions on Neural Networks*, vol. 13, no. 3, pp. 697–710, May 2002.
[54] R. Singh, M. Vatsa, A. Ross, and A. Noore, "A mosaicing scheme for pose-invariant face recognition," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 37, no. 5, pp. 1212–1225, Oct. 2007.
[55] P. Tresadern, T. Cootes, N. Poh, P. Matejka, A. Hadid, C. Lvy, C. McCool, and S. Marcel, "Mobile biometrics: Combined face and voice verification for a mobile platform," *IEEE Pervasive Computing*, vol. 12, no. 1, pp. 79–87, 2013.
[56] C. Liu, "Gabor-based kernel pca with fractional power polynomial models for face recognition," *IEEE TPAMI*, vol. 26, no. 5, pp. 572–581, May 2004.
[57] M. Tistarelli and E. Grosso, "Active vision-based face authentication," *Image and Vision Computing*, vol. 18, no. 4, pp. 299–314, 2000.
[58] K. Lee and H. Byun, "A new face authentication system for memory-constrained devices," *IEEE Consumer Electronics*, vol. 49, no. 4, pp. 1214–1222, Nov. 2003.
[59] J. Kittler, Y. Li, and J. Matas, "Face authentication using client specific fisherfaces," *The Statistics of Directions, Shapes and Images*, pp. 63–66, 1999.
[60] T. Bourlai, K. Messer, and J. Kittler, "Face verification system architecture using smart cards," in *Proc. ICPR 2004*, vol. 1, Aug. 2004, pp. 793–796.
[61] A. Hadid, N. Evans, S. Marcel, and J. Fierrez, "Biometrics systems under spoofing attack: An evaluation methodology and lessons learned," *IEEE Signal Processing Magazine*, vol. 32, no. 5, pp. 20–30, Sep. 2015.
[62] M. Muaaz and R. Mayrhofer, "Smartphone-based gait recognition: From authentication to imitation," *IEEE TMC*, 2017, *to appear*.
[63] P. Kocher, J. Jaffe, and B. Jun, "Differential power analysis," in *Proc. CRYPTO99 1999*, 1999, pp. 388–397.
[64] D. Vermoen, M. Witteman, and G. N. Gaydadjiev, "Reverse engineering java card applets using power analysis," in *Proc. IFIP 2007*. Springer, 2007, pp. 138–149.
[65] A. Nagar, "Biometric template security," Ph.D. dissertation, Michigan State University, 2012.
[66] T. Hastie, R. Tibshirani, and J. Friedman, *The elements of statistical learning: Data Mining, Inference, and Prediction*, 2nd ed., ser. Series in Statistics. Berlin: Springer, 2011.
[67] A. Dobson, *An Introduction to Generalized Linear Models, Second Edition*, ser. Chapman & Hall/CRC Texts in Statistical Science. Taylor & Francis, 2010.
[68] J. Galbally, C. McCool, J. Fierrez, S. Marcel, and J. Ortega-Garcia, "On the vulnerability of face verification systems to hill-climbing attacks," *Pattern Recognition*, vol. 43, no. 3, pp. 1027–1038, 2010.
[69] M. Martinez-Diaz, J. Fierrez, J. Galbally, and J. Ortega-Garcia, "An evaluation of indirect attacks and countermeasures in fingerprint verification systems," *Pattern Recognition Letters*, vol. 32, no. 12, pp. 1643–1651, 2011.
[70] B. Vibert, C. Rosenberger, and A. Ninassi, "Security and performance evaluation platform of biometric match on card," in *Proc. WCCIT 2013*, Jun. 2013, pp. 1–6.
[71] ISO, *Identification cards – Integrated circuit cards – Part 4: Organization, security and commands for interchange*, 2005.
[72] A. K. Jain, B. F. Klare, and A. Ross, "Guidelines for best practices in biometrics research," in *International Conference on Biometrics (ICB)*, vol. 8, Phuket, Thailand, May 2015.
[73] M. Muaaz and R. Mayrhofer, "Cross pocket gait authentication using mobile phone based accelerometer sensor," in *Proc. EUROCAST 2015*. Las Palmas de Gran Canaria, Spain: Springer, Feb. 2015, pp. 731–738.
[74] ——, "An analysis of different approaches to gait recognition using cell phone based accelerometers," in *Proc. MoMM 2013*. NY, USA: ACM, 2013, pp. 293:293–293:300.
[75] A. Savitzky and M. J. E. Golay, "Smoothing and differentiation of data by simplified least squares procedures," *Analytical Chemistry*, vol. 36, no. 8, pp. 1627–1639, 1964.
[76] N. Belgacem, A. Ali, R. Fournier, and F. Bereksi-Reguig, "ECG based human authentication using wavelets and random forests," *IJCIS*, vol. 2, no. 2, pp. 1–11, 2012.
[77] C. Bouten, K. Koekkoek, M. Verduin, R. Kodde, and J. Janssen, "A triaxial accelerometer and portable data processing unit for the assessment of daily physical activity," *IEEE Biomedical Engineering*, vol. 44, no. 3, pp. 136–147, 1997.
[78] R. D. Findling, M. Muaaz, D. Hintze, and R. Mayrhofer, "Shake-unlock: Securely transfer authentication states between mobile devices," *IEEE TMC*, vol. 16, no. 4, pp. 1163–1175, Apr. 2017.
[79] D. Winter, *Biomechanics and Motor Control of Human Movement*. Wiley, 2004.
[80] I. Daubechies, "Orthonormal bases of compactly supported wavelets ii. variations on a theme," *SIAM Journal on Mathematical Analysis*, vol. 24, no. 2, pp. 499–519, 1993.
[81] K.-C. Lee, J. Ho, and D. J. Kriegman, "Acquiring linear subspaces for face recognition under variable lighting," *IEEE TPAMI*, vol. 27, no. 5, pp. 684–698, May 2005.
[82] R. D. Findling, "Pan shot face unlock: Towards unlocking personal mobile devices using stereo vision and biometric face information from multiple perspectives," Master's thesis, Department of Mobile Computing, School of Informatics, Communication and Media, University of Applied Sciences Upper Austria, Softwarepark 11, 4232 Hagenberg/Austria, Sep. 2013, *Awarded the OCG Incentive Award FH 2014 and IFAC Fred Margulies Award 2015*.
[83] P. Viola and M. Jones, "Robust real-time face detection," *International Journal of Computer Vision*, vol. 57, pp. 137–154, 2004.

Rainhard Dieter Findling received his Dr. techn. (PhD) degree in Computer Science in 2017 from the Institute of Networks and Security (INS) at the Johannes Kepler University Linz (JKU), Austria, and his BSc and MSc degrees in Engineering in 2011 and 2013 from the Department of Mobile Computing at the University of Applied Sciences Upper Austria. He currently is a postdoctoral researcher in the Ambient Intelligence Group at Aalto University, Finland. His research interests include machine learning and signal analysis with mobile devices and sensors to obtain unobtrusive authentication and security mechanisms in modern mobile and ubiquitous computing environments.

Michael Hölzl received his MSc degree in Engineering at the University of Applied Sciences Upper Austria at the Mobile Computing degree program in 2012. He is currently employed at the Institute of Networks and Security (INS) at the Johannes Kepler University Linz (JKU), where he is working towards his PhD. His main research interests include the integration of tamper-resistant hardware for security-sensitive application as well as cryptographic protocols and privacy-preserving digital identities on mobile devices.

René Mayrhofer heads the Institute of Networks and Security (INS) at Johannes Kepler University Linz (JKU), Austria. Previously, he held a full professorship for Mobile Computing at Upper Austria University of Applied Sciences, Campus Hagenberg, a guest professorship for Mobile Computing at University of Vienna, and a Marie Curie Fellowship at Lancaster University, UK. His research interests include computer security, mobile devices, network communication, and machine learning, which he brings together in his research on securing spontaneous, mobile interaction. René has contributed to over 60 peer-reviewed publications and is a reviewer for numerous journals and conferences. He received Dipl.-Ing. (MSc) and Dr. techn. (PhD) degrees from Johannes Kepler University Linz, Austria and his Venia Docendi for Applied Computer Science from University of Vienna, Austria.