Mobile Gait Match-on-Card Authentication from Acceleration Data with Offline-Simplified Models

Rainhard Dieter Findling
FH Upper Austria & JRC u'smile
Softwarepark 11
4232 Hagenberg, Austria
rainhard.findling@fh-hagenberg.at

Michael Hölzl
JKU Linz & JRC u'smile
Altenbergerstrasse 69
4040 Linz, Austria
hoelzl@ins.jku.at

René Mayrhofer
JKU Linz & JRC u'smile
Altenbergerstrasse 69
4040 Linz, Austria
rene.mayrhofer@jku.at

ABSTRACT

Mobile biometric match-on-card authentication requires matching procedures on smart cards (SCs) to be limited due to computational restrictions. We present an approach that uses offline training to obtain authentication models with a simplistic internal representation in the final trained state, whereat we adapt features and model representation to enable their usage on SCs. The obtained model is used on SCs without requiring retraining when enrolling individual users. We apply our approach to acceleration based mobile gait authentication, using a 16 bit integer range Java Card, and evaluate authentication performance and computation time on the SC using a publicly available dataset. Results indicate that our approach is feasible with an equal error rate of ∼12% and a computation time below 2 s on the SC, including data transmissions and computations. To the best of our knowledge, this thereby represents the first practically feasible approach towards acceleration based gait match-on-card authentication.

CCS Concepts

•**Security and privacy** → **Biometrics;** *Tamper-proof and tamper-resistant designs;* •**Human-centered computing** → **Mobile devices;**

Keywords

Acceleration; authentication; gait; match-on-card; mobile biometrics; smart card;

1. INTRODUCTION

Biometric authentication, such as fingerprint, gait, or voice authentication [28], 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 [58], user biometrics cannot easily be changed by users in case of them being disclosed. Consequently, leakage or theft of biometric information has severe consequences: attackers could e.g. reconstruct original biometrics form obtained information and use it for replay attacks [8]. Usage of reconstructed biometrics beyond the associated mobile device might be possible too, as they are naturally the same across all system they are used with (cf. [26, 42, 46, 57]). Further, in contrast to desktop computers, mobile devices are easier lost, stolen, or inwardly accessed by attackers. 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 need to be protected adequately. One approach to doing so is using smart cards (SC) [48], which become increasingly available in off the shelf mobile devices as so-called secure elements (SEs). These SEs are often shipped with NFC and can either be directly embedded in the phone hardware, extended with an SD card, or provided within modern SIM cards [21]. Biometric authentication with SCs can be done as either template on card (TOC) or match on card (MOC) [3,7,10,27,28]. 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, new biometrics are recorded, then the enrolled templates are fetched from the SC and compared with the new readings outside the SC. In contrast, with MOC authentication, new readings are instead transfered to the SC, where they are compared with previously stored templates directly on the SC. This leads to the following noticeable differences of MOC over TOC: one 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 readings 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 thoughtfully.

As reducing the risk of leakage or theft of biometric templates is considerably important, MOC is regularly preferred over TOC, despite the accompanying computational limitations. In turn, those lead to restrictions in how existing MOC approaches are frequently designed [3, 9, 18, 27, 28, 45]:

MoMM 2016 November 28–30, 2016, Singapore

DOI: 10.475/123_4

- MOC approaches usually rely on restricted operations and logic for matching templates with new readings, often not utilizing e.g. 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, as the underlying mechanisms have been strongly adapted to the used biometrics. This impedes the adaption of new biometrics in MOC approaches – which would benefit from having reusable concepts available for e.g. feature derivation, model representation, and matching operations.

To address those restrictions we aim for enabling a more generic usage of simple ML models on SCs, which are computed offline with sufficient computational power and don't 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 to train and generate ML models offline (e.g. using server infrastructure), then to use the simplified internal structure the trained models on SCs in the matching process (Fig. 1). 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 offline obtaining such a model, both features and models need to be adapted to suit SC restrictions. This comprises data types of features and models, as well as computations on those (e.g. by being restricted to integer space). 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 derivation. Further, note that feature derivation up to feature simplification for usage on the SC can be performed outside the SC. This allows for more complex feature derivation than possible

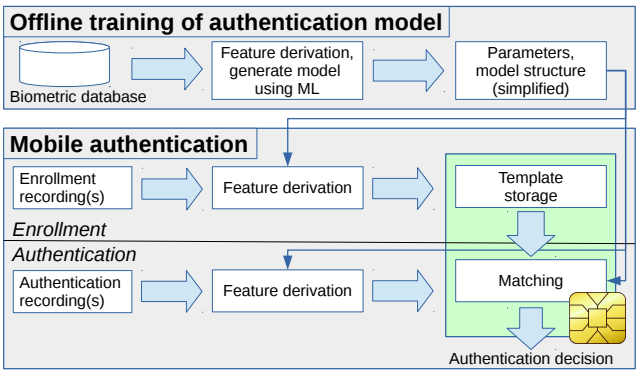


Figure 1: Conceptual overview of the proposed approach. The SC is highlighted in green.

on SCs. Doing so does not disclose any information about biometric information stored on the SC, as it exclusively involves information from new biometric readings.

In this paper we demonstrate the proposed approach on mobile, acceleration based gait as biometrics on a SC restricted to 16 bit range integer calculations. We restrict both features derived from gait recordings and model structure used on the SC to 8 bit integer values. This allows to compute multiplications of such within a 16 bit integer range. The goal we thereby pursue is to demonstrate that using the obtained model, stored biometric template, and new biometric reading in 8 bit representation, adequate authentication on the SC as MOC is still feasible. Summarizing, our contributions are:

- We present a generic approach towards biometric MOC authentication, using offline trained ML models and adaption of models and features to enable their computation and handling on SCs.

- We apply our approach to acceleration based gait authentication, thereby – to the best of our knowledge – presenting the first practical approach to gait MOC authentication with acceleration data.

- We evaluate the performance and feasibility of our approach in an acceleration gait authentication scenario using a publicly available gait data set and a Java Card SC with 16 bit integer calculation range. We thereby find it to be feasible with an EER of ∼12% and authentication time below 2 s.

2. RELATED WORK

To this date, fingerprints are the best researched biometrics with match-on-card authentication approaches. They usually feature small templates, thereby a small amount of features (mostly minutiae based), which in turn leads to relatively simple matching procedures [3, 17, 19, 45].

Biometrics other than fingerprints for match-on-card authentication has been covered by few research, with some of it addressing e.g. the issue of face authentication usually utilizing bigger templates than fingerprint authentication. Tistarelli et al. [56] propose a face authentication TOC approach, in which they use using morphological filtering and adaptive template matching to extract the position of relevant facial features as features for matching. During matching they fetch enrolled templates from the card and compare them to new readings using a space-variant approach based on principal component analysis (PCA). Kittler et al. [29] state that PCA compresses templates in a suboptimal way for usage on SC. They therefore propose a MOC approach using a 1D, client specific linear discriminant analysis (LDA), of which they utilize the distance of new readings 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. [4] utilize the client specific LDA proposed in [29] as feature derivation mechanism, then use the vector dot product of new reading and enrolled template with a predefined threshold to obtain an authentication decision. Finally, Lee and Bun [32] 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.

Further examples of biometric match-on-card authentication include speaker verification and iris recognition. For example, Choi et al. [9] use support vector machines (SVM) with a limited amount of features and FPGAs for speaker verification in a match-on-card manner. Czajka et al. [10] perform iris recognition by deriving a 1024 bit iris code from samples outside the SC, then match new readings with enrolled templates on the card using a computationally simple Hamming distance. This approach therefore is more similar to fingerprint than e.g. face authentication in terms of template size. Another, still related example is human identification from CCTV records [37]. 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. BACKGROUND

3.1 Smart Cards

Smart cards (SC), as well as secure elements (SE) used in mobile devices, are special integrated circuits which provide certain characteristics that are useful for security sensitive applications: (i) Cryptographic operations (e.g. encryption, decryption, hashing) can be performed directly on the chip, often in hardware. (ii) SC 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. (iii) Data and application code on 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, SC also bring limitations that need to be considered for applications relying on them: (i) data transfer to/from SC with a restricted maximum bandwidth (cf. Hölzl et al. [21] with measurements of 329 B/s for contactless and 3,31 kB/s for contact cards). (ii) While some modern SC 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 on those cards. (iii) Persistent and volatile memory are highly limited with a maximum capacity of around one megabyte for current cards. (iv) Finally, SCs are limited in computation capabilities: for example, there are no native floating point operations in hardware available, and computations performed in software are considerably slower than on PCs or mobile devices (clock rate of SCs usually is in the MHz range).

These limitations in computation and data transfer directly affect the internal structure of authentication models that can be used on the SC as well as the number and type of features transmitted. For example, relying on 4 byte integers requires to use more complex data structures for the computations internally (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 SE is limited, which too limits the amount of data that can be sent to the SE on 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 rather generic way.

3.2 Gait Identification and Authentication

Gait identification and authentication are the processes of identifying and recognizing individuals by their distinctive walking style [31]. Thereby, identification deals with recognizing an individual from a set of individuals using gait data, while authentication deals with determining if two gait recordings have been originated by the same individual. Both identification and authentication can be based on different types of data, including visually sensed information (e.g. humans recorded in context of CCTV surveillance [51]), floor sensed information (sensors being embedded with floors humans walk on, such as pressure sensors [36]), and information from sensors worn by humans themselves [15]. With the latter, different sensor types and sensor positions on the human body have been utilized [14]. Besides dedicated sensors, also modern 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 identification and authentication [55].

Most gait identification and authentication approaches utilize a toolchain comprising data recording, preprocessing, segmentation, cleaning, features extraction, and a matching procedure. As human walking is of cyclic nature, each step (or pair of left-right or right-left steps) can be seen as repetitive cycle. For preprocessing, both step-cycle based approaches ("cycle based") and window based approaches have been utilized in literature [20]. With cycle based approaches, 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 to segment data chunks, which are again used for subsequent recognition.

The matching procedure of acceleration gait identification and authentication often involves dynamic time warping (DTW) as distance metric between two time series [33,40,61]. 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 [44, 50], even most limited DTW approaches are still difficult to calculate on SCs.

For acceleration based gait identification and authentication without using SCs and DTW, a number of features has been used. Those include: average, median, min, max, standard deviation (SD), and median absolute deviation (MAD) acceleration of individual axes and their magnitude [31, 43], root mean square (RMS) acceleration [43], mean- and zero-crossings [43], principal component coefficients of acceleration [5, 54], binned acceleration distribution [15, 31, 43], time between peaks [31], discrete cosine and fast Fourier transformation coefficients [1,13,22,49], and Mel- and Bark-frequency cepstral coefficients [20, 43]. Further, wavelet transformations have been used with non-cycle-based acceleration gait data [20, 47] and floor sensor based gait data [38], as well on acceleration gait style recognition [24], which in contrast to gait identification or authentication does not distinguish indi-

viduals but gait styles. On those features, again a number of non-DTW based models have been applied, including cross-correlation based [34] or tree based models [31], artificial neural networks (ANN) [31,53], support vector machines [43,54], analysis of variance (ANOVA) [1], Gaussian mixture models (GMM) [22], and hidden Markov models (HMM) [43].

4. THREAT MODEL

Assuming attackers have access to a user's mobile device that stores and processes biometric information without using a SC, multiple attack vectors towards obtaining the user's biometric information become possible. When having access to the device, attackers could simply read and extract the biometric templates that have been stored on the mobile device during enrollment of the legitimate user. This does not require any authentication interaction and is therefore also possible if the device comes under attackers control long after enrollment. Further, templates could be extracted during enrollment of legitimate users, or when they try to authenticate. Doing so would be possible while biometrics are recorded or preprocessed, when their features are derived, during storing them on the mobile device, or during matching new readings with previously stored templates.

Using TOC mechanisms to store biometric information rules out the first attack vector: simply extracting biometric templates stored on the device is no longer possible. However, if attackers are also able to trigger an authentication attempt, the legitimate user's biometric templates will be fetched from the SC for comparison with the newly recorded biometric readings outside the SC. Again, this enables attackers to extract stored biometric templates at any time they are able to trigger an authentication attempt, without requiring any interaction of legitimate users. Using MOC mechanisms instead to store and compare biometric information on the SC also rules out the latter attack vector. Attackers cannot extract biometric templates even if they have full access to the mobile device and are able to trigger an authentication attempt, as biometric templates never leave the SC after enrollment.

What remains are attack vectors including interaction of legitimate users, e.g. while they try to enroll or authenticate to a compromised device. Information could thereby be extracted during recording, preprocessing, feature extraction – up to the point of handing data to the SC. Thereby, interaction of legitimate users being required greatly limits the time windows for such attacks. This underlines that the advantage of MOC approaches lies with attackers being required to having control over the mobile device *at the time* of legitimate users enrolling or authenticating. One possibility to prevent those remaining attack vectors is to protect the whole processing chain – from recording biometric readings at sensors to processing and transmitting them to the SC – in a so called trusted execution environment (TEE, e.g. ARM TrustZone[1]). Incorporating a TEE with the proposed approach is defined beyond the scope of this work, but might be in focus of future research. Further attacks on the security of SCs themselves, such as side-channel attacks by Kocher et al. [30] or Vermoen et al. [59], which try to extract the biometric template from the SC itself, are also defined out of scope of this paper.

[1]http://www.arm.com/products/processors/technologies/trustzone/

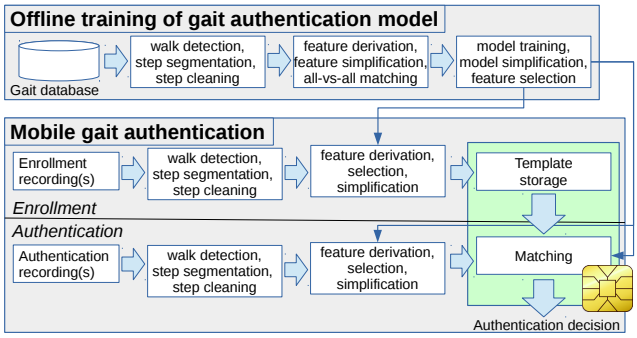


Figure 2: Overview of our approach on gait authentication with offline model computation and simplification, and on-device enrollment and authentication in MOC manner. The SC is highlighted in green).

5. METHOD

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 match-on-card approach that combines a non-DTW based model and features previously used in acceleration gait recognition with features from other domains. Our approach is divided into two major parts: offline model generation and usage of this model for enrollment/authentication on the mobile device (Fig. 2).

Offline model computation and enrollment/authentication on mobile devices use the same steps for preprocessing. Those include: detection of users walking, segmentation of steps, normalization and cleaning of extracted steps. Feature derivation, without feature selection and simplification, is identical for offline computation and mobile devices as well. The remaining steps differ slightly. Based on preprocessed step samples, offline computation first trains and evaluates a model, then performs model simplification and feature filtering, and finally estimates the resulting authentication performance. The obtained model is stored on SCs of mobile devices intended for authentication and does not need to be retrained during enrollment of individual users.

In contrast, enrollment and authentication on mobile devices use feature selection parameters obtained from previous offline computation, and thereafter simplify remaining features. Additionally, in enrollment, features of preprocessed step samples are used as biometric templates of the user and stored on the SC. In authentication, features of the new recordings are transfered to the SC, where they are used together with previously stored samples of the enrolled template and the offline-trained model to obtain an authentication decision.

5.1 Offline Model Creation

5.1.1 Gait Data Preprocessing

Preprocessing mechanisms are adapted from Nickel [43] as well as Muaaz and Mayrhofer [39, 41], which comprise of walking detection and preprocessing, as well as subsequent step detection and preprocessing, which be 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. Per segment, to compensate for gravity, we remove

the mean acceleration per 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 [52] with window length 15 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. 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 with 100 Hz sampling rate corresponds to a duration 1 s. Cycles that diverge largely from the majority of extracted cycles are further defined as outliers and discarded. For that purpose we compute the normalized dynamic time warping (DTW) distance[2] between all n cycles and discard those for which $\geq \frac{n}{2}$ distances are above a predefined threshold of 0.6. The remaining gait cycles are used in feature derivation and subsequently handed to the SC for enrollment or authentication (Fig. 3).

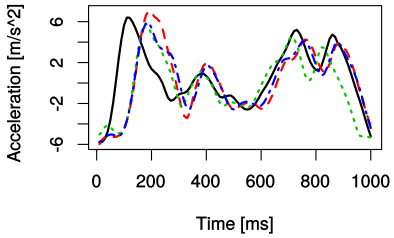


Figure 3: Examples of preprocessed gait cycles with a uniform length of 1 s, consisting of 100 values each.

5.1.2 Step Cycle Feature Derivation

For each preprocessed cycle we derive a number of time and frequency domain features, as well as a wavelet transformed representation. For time domain features we utilize the mean, median, standard deviation (SD), median absolute deviation (MAD), and autocorrelation (AC) series with a maximum shift of 100 values as features. AC has been used as signal preprocessing in other biometric recognition tasks, such as electrocardiography (ECG) recognition [2], 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 each third value as feature, which with a sampling rate of 100 Hz corresponds to a shift granularity of 30 ms. For frequency domain features 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 [6, 12, 62], we use both frequency power and phase in this range as features. Frequency power and phase are added as separate features to a) avoid handing complex values to models and b) enable separately treating them (e.g. normalizing and discarding features individually). For wavelet representation of steps, we perform a discrete wavelet transform (DWT) using a multiresolution analysis of 6 levels. As wavelet we utilize a least asymmetric Daubechies wavelet [11] of length 8. As with FFT features, all wavelet features are treated as individual features too.

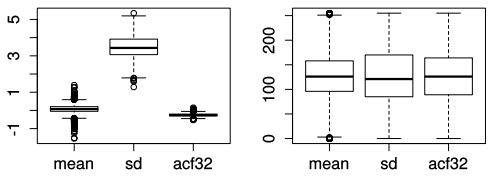
By combining mean, median, SD, MAD, AC, FFT and wavelet features we obtain a feature vector of length 177 for each gait cycle. We later only use a subset of those features in our generated model. However, as the used feature selection relies on at first using all features in offline model training, we describe feature simplification and offline model training before focusing on creating the feature subset in Sec. 5.1.6.

5.1.3 Feature Simplification

The derived features are in $\mathbb{R}$, hence need to be transformed (scaled, shifted, and rounded) to a representation fitting an 8 bit integer range. This is done by computing the mean and SD per feature over all data available in the offline training data. For feature simplification, a transformation is applied to each feature f (Eq. 1), with resulting features being bigger or smaller than the boundaries of the 8 bit space being capped (Eq. 2). This ensures that the 8 bit space can be optimally used for the mainstream data, while boundaries are respected also for new, unseen data with potential outliers. The resulting feature vector therefore consists of simplified features f_S in the range $[0, 255]$ (Fig. 4).

$$a = \text{round}\left(\frac{(f - \text{mean}(f))}{2 \cdot SD(f)} \cdot 127 + 127\right) \tag{1}$$

$$f_S = \begin{cases} 255 & a > 255 \\ a & 0 \leq a \leq 255 \\ 0 & a < 0 \end{cases} \tag{2}$$



(a) Original (b) Simplified

Figure 4: Feature simplification to 8 bit range $[0, 255]$, exemplary applied to the mean, SD, and autocorrelation feature 32 (ACF32).

On the mobile device, the same feature preprocessing and simplification transformation is applied to features of new recordings during enrollment and authentication. The mean and SD per feature, computed from offline training data, are stored on the mobile device outside the SC for this purpose[3]. Simplified features of biometric readings are handed to the SC for enrollment or authentication purposes then. We refer to those feature vectors as biometric feature vectors.

5.1.4 Model Training

Offline model training uses pairs of samples represented by their feature vectors. At first, the distance between two

[2]The DTW distance calculation is done for preprocessing purposes and outside the SC, consequently is not related to the model utilized for gait authentication on the SC.

[3]As with subsequent feature selection only 75 features remain, the mean and SD for normalizing features is only stored for those features on the device.

biometric feature vectors v_1 and v_2 yields a 8 bit feature distance vector d (Eq. 3).

$$d = |v_1 - v_2| \tag{3}$$

We refer to feature distance vectors originated by the same person as being a sample of the positive class P (a "positive sample"), and to those originated by different people as being a sample of the negative class N (a "negative sample"). Using feature distance vectors of P and N samples from our offline training data we create a classification model able to distinguish between those (see Sec. 6 for details on how data partitioning was done with training data). 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 type we use a generalized linear model. In their internal, ready trained state, such models are represented by coefficients (the slope S) and an additional intercept I (the offset to the origin of the coordinate system). They predict a sample's class membership C_p using a linear combination of the sample's distance vector with slope and intercept, then use a predefined threshold to decide on the class (Eq. 4).

$$C_p = \begin{cases} P & \sum_i d_i \cdot S_i > I \\ N & \sum_i d_i \cdot S_i \leq I \end{cases} \tag{4}$$

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.

5.1.5 Model Simplification

The slope S and intercept I obtained from model training are in $\mathbb{R}$, hence, similar to biometric features, have to be transformed to an 8 bit representation for usage on the SC. We therefore scale model coefficients to optimally fit an 8 bit range of $[-128, 127]$, apply a cap at boundaries, and transform the intercept accordingly (Eq. 5 and. 6).

$$b = \text{round}\left(\frac{S}{2 \cdot SD(S)} \cdot 127\right) \tag{5}$$

$$S_s = \begin{cases} -128 & b < -128 \\ b & -128 \leq b \leq 127 \\ 127 & b > 127 \end{cases} \tag{6}$$

In contrast to transforming biometric features, no shift is applied. This would otherwise change the meaning of coefficients (coefficients around 0 have less influence on the result than those with higher absolute values). Having both feature distance vectors and the slope in 8 bit integer representation allows for piecewise multiplication of them in 16 bit integer range. This can therefore be done efficiently on SCs that only support integer calculations with 16 bit integers in hardware. To keep the linear combination of all products within a range of 16 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 divided by the length of the slope vector (Eq. 7 and 8).

$$c = \sum_i \left(\frac{S_{s,i} \cdot d_i}{\text{length}(S_{s,i})}\right) \tag{7}$$

$$C_p = \begin{cases} P & c > \frac{I}{\text{length}(S_{s,i})} \\ N & c \leq \frac{I}{\text{length}(S_{s,i})} \end{cases} \tag{8}$$

Besides allowing for linear combination in 16 bit range, this representation of the model has the advantage of requiring only $n + 2$ byte of storage memory with using n features (n byte for the slope and 2 byte for the intercept).

5.1.6 Feature Selection

After model training, we utilize a feature selection mechanism using the coefficients obtained from linear model training. Features which are associated to small coefficients necessarily have small influence on the output variable, hence can possibly be removed without severely reducing classification performance. The core advantage of doing so is that using less features causes less computations on the SE, therefore speeds up processing. Another, smaller advantage is that relying on stronger features could slightly increase overall predictive power of the model. However, as small coefficients don't necessarily denote features unimportant for separating classes, prediction capabilities might as well be slightly reduced by doing so. In preliminary tests we were able to exclude features of which the corresponding coefficients were 25% or smaller than the strongest coefficient, without severely downgrading the resulting distinguishability of the model. This leads to 76 of the original 177 features remaining in our model (Fig. 5).

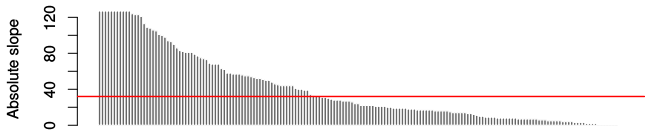


Figure 5: Ordered feature importance of all original 177 features, denoted by the absolute of the associated slope from the linear model. The red line denotes the border of discarding features with small slope.

Based on those features, we subsequently exclude features with correlation ≥ 0.8 to any other feature, to further reduce inter-correlation of features. As inter-feature correlation already is small before doing so (Fig. 6), only the MAD feature is excluded with this step. Consequently, both the final slope and biometric feature vector utilized in the model is of length 75, which leads to a required storage memory of $75 + 2 = 77$ byte for the model and 75 byte per stored biometric template of the user.

5.2 Mobile Device: Enrollment and Authentication

Preparation of mobile devices comprises storing the feature normalization and simplification parameters as well as the model itself (slope and intercept) on the SC. For data recording on mobile devices we utilize step sensors (pedometers) and accelerometers. Pedometers are an integral part of many modern mobile devices already: they allow for monitoring device acceleration without constantly draining the battery. We use such to notify the main CPU and start recording

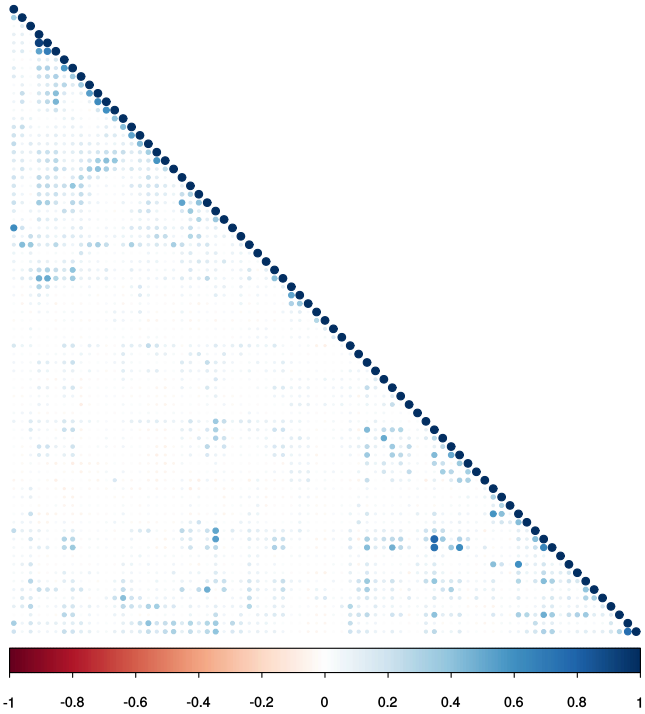


Figure 6: Inter-feature correlation of resulting features: the correlation is already small due to previous selection mechanisms, leading to only the MAD feature being excluded with a correlation threshold of 0.8.

of gait data using 3D acceleration sensors if steps are detected. After data recording, enrollment and authentication use the same approach as offline model creation towards data preprocessing, feature derivation, and feature simplification (Sec. 5.1). Note that on mobile devices those can be done outside the SC, as they don't use any information about templates stored previously on the SC, therefore don't require any calculations on the SC.

For enrollment, m feature vectors of m newly recorded gait cycles are handed 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 of n newly recorded gait cycles are again handed to the SC. As this transmission is done for each authentication attempt, the transfer period is important and measured in our evaluation later. On the SC we perform $m \cdot n$ comparisons between all m stored reference cycles and all n newly transmitted cycles using the stored, offline-computed model. The resulting $m \cdot n$ predictions (each P or N) are treated as votes, thereby yield a final, binary authentication decision – which is handed from the SC to the mobile device to authorize or deny an authentication attempt. If we would instead hand a 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 derive information about users' biometrics [16, 35, 57, 60] – which is why the SC only yields binary authentication decisions.

6. EVALUATION

6.1 Dataset and Data Partitioning

For evaluation of our approach we use the gait data set of Muaaz and Mayrhofer [41] 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. We train and evaluate our approach with subset of this data.

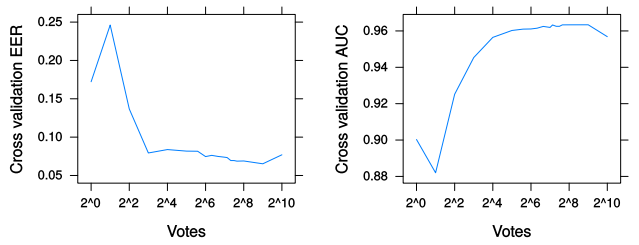
To obtain a realistic estimate of the authentication performance on people unseen by the model during training, we perform a 50/50 population independent split on the dataset [25]. We thereby assign 50% of participants to the training partition, which is used for training the model, and 50% of participants to the test partition, which is only used once for estimating the performance of the evaluated, chosen, and ready trained final model on yet unseen people. We apply feature derivation and simplification to the training partition as stated in Sec. 5, then use determined feature simplification parameters to apply the same to the test partition – as it would be done on mobile devices. Due to slightly different amounts of gait cycles being discarded per participant during preprocessing and data cleaning, this results in a total of 2132 and 1943 unique gait cycles in the training and test partition, respectively. For both training and test partition, we use all combinations of different gait cycles originated by the same person to obtain P samples, and all combinations of gait cycles originated by different people (within the corresponding partition) to obtain N samples. We further use all combinations of gait cycles between train and test partition (which are necessarily originated by different people) as additional, population dependent test partition. Due to the size of the training partition and the resulting training complexity, we use a random subset of 100000 P and 150000 N samples for training the model. However, for intra-training evaluation of trained models, the full training partition size is utilized nevertheless (Tab. 1).

Partition	Cycles	P	N
Training	2 132	174 410	2 207 243
Test, pop. independent	1 943	168 976	2 158 427
Test, pop. dependent	–	0	6 918 157

Table 1: Size of the training, population independent, and population dependent partitions, as amount of gait cycles and the resulting amount of P and N samples.

6.2 Model Evaluation

P and N samples from the subset training partition are used to train and evaluate all different parametrization of our model to find a suitable configuration for distinguishing between P and N samples. As training and evaluation procedure we thereby use well established 10-fold cross validation with 10 repetitions and measure the authentication performance as receiver operating characteristics (ROC) curve, area under the ROC curve (AUC), and equal error rate (EER). After an optimal configuration has been found, the final model is trained with it using all training data. It is



(a) EER (b) AUC

Figure 7: Authentication performance as AUC and EER over the total amount of votes, as result of comparing multiple gait cycles from the enrolled template with multiple cycles from the new reading during authentication.

then evaluated by separately applying it once on the population independent and population dependent test partitions 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 determined parametrization except the final decision threshold on the test partition.

The resulting model further serves as basis for voting when using multiple gait cycles in both template stored on the SC and new reading sensed for authentication. Thereby, m cycles are contained in the enrolled template and n new readings 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 partitioning, with the training partition being used to evaluate the authentication performance of different voting parameters (nr. of votes used in voting), and the test partitions being used only for reporting a final authentication performance estimate on the final, voting based model.

Based on a single comparison between one cycle as enrolled template and one cycle as new reading for authentication, we achieve an AUC of 0.900 and EER of 0.828 in our cross validation evaluation. The authentication performance over number of votes (Fig. 7) supports intuition, as increasing the number of votes goes alongside an increased authentication performance. As tradeoff between authentication performance and computation time, we achieved a reasonable AUC of 0.961 and EER of 0.921 with a total of $2^6 = 64$ votes (e.g. using $n = m = 8$) and a voting threshold of 39.8% (P being predicted with at least 39.8% votes for P, and N otherwise). We want to emphasize that, although the increase in EER slows down with around $2^3 = 8$ votes, the overall authentication performance still rises further when increasing the amount of votes (as reflected by the AUC). We also observed this effect in our experiments when we applied the held-back test set to a voting classifier using only 8 votes, which resulted in a noticeably worse false positive rate (FPR) as compared to using $2^6 = 64$ votes instead. An intuitive explanation for this could be that a higher number of total comparisons has more capabilities of reflecting the allowed intra-person variance in gait cycles, while a comparison of few samples lacks this information. When applying the resulting models to the population independent test set, we obtain a TPR of 0.770, TNR of 0.804, and intra-partition AUC of 0.863 for using single samples – and a TPR of 0.899, TNR of 0.868, and intra-partition AUC of 0.958 for using 64 votes (Tab. 2 and Fig. 8).

Partition	Votes	AUC	EER	TPR	TNR
Training	1	0.900	0.828	–	–
Test, pop. indep.	1	0.863	0.789	0.770	0.804
Test, pop. dep.	1	–	–	–	0.793
Training	1	0.921	–	–	–
Test, pop. indep.	1	0.958	0.881	0.899	0.868
Test, pop. dep.	1	–	–	–	0.896

Table 2: Evaluation results for using a single vote (single gait cycle in both the template and the new reading), and a total of 64 votes (e.g. 8 templates and 8 new readings to compare to) for the training, population independent, and population dependent test partition.

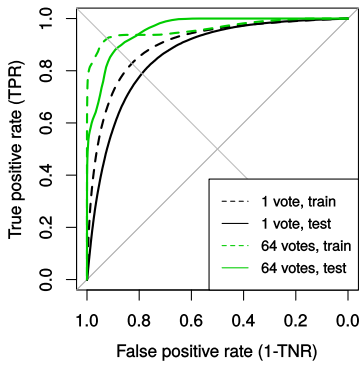


Figure 8: Receiver operating characteristics (ROC) curves for using a single gait cycle in both the template and the new reading, and a total of 64 votes (e.g. 8 templates and 8 new readings to compare to) for the training and population independent test partition.

6.3 Smart Card Computation Time

For evaluating the computation time of our approach on SCs we use a 16 bit JCOP 2.4.1 smart card with 80 kB EEPROM memory running Java Card version 2.2.2 and communicate over the contactless interface. The round-trip time of transferring a 75 byte gait cycle to the SC card and yielding a 2 byte authentication decision back to the mobile device was measured to be 6.6 ms on average. This duration excludes computations on the SC and scales linearly when sending multiple gait cycles instead (e.g. 8 cycles with an average of 52.8 ms). Similarly, the measured mean computation time of our approach on SCs (Tab. 3 and Fig. 9, both including transmission time) indicates a nearly linear increase of computation time over number of used gait cycle votes. Those include the calculation of distances between transmitted gait cycles and cycles stored on the SC in the enrolled template, the linear combination of distances with model parameters determined offline, the voting of individual results to obtain an authentication decision, and the yielding thereof.

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 reading has little impact if the number of total votes $m \cdot n$ is unaffected. With using 8 cycles in the enrolled template and 8 cycles in new

T. cycles	8	16	24	32	40	48	56	64
Mean [ms]	235	437	641	842	1047	1247	1451	1654
SD [ms]	2.83	3.42	3.48	3.54	3.98	3.90	4.14	3.97

Table 3: Mean and standard deviation (SD) of the computation time of our approach on the SC. The time includes sending 1 sample from a new reading to the SC, computing votes with all samples stored in the enrolled template (T. cycles), and computing and yielding a 2 byte authentication decision.

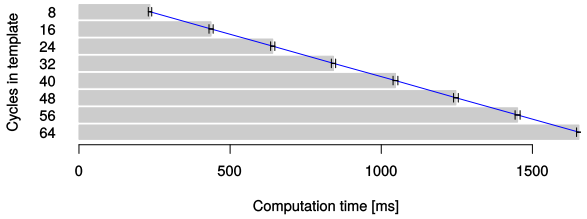


Figure 9: Visualization of Tab. 3, with error bars depicting the doubled standard deviation.

readings we achieve a computation time of at maximum $8 \cdot 235\,\text{ms} = 1880\,\text{ms}$ on average – which we argue to be reasonable as gait authentication delay, as recording the corresponding gait data on the mobile necessarily takes longer. Note that the computation time could be reduced further by sending multiple cycles to the SC at the same time (i.e. each single cycle sent to the SC requires initialization and additional computations). However, 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 [23]). While this limitation could be overcome by using the extended version of the protocol (extended length fields in [23]), we consider the short and therefore slower variant for interoperability with all currently deployed smarts cards in our measurements.

Therefore, our approach is able to yield a TPR of 0.899 and TNR of 0.868 with the associated MOC computations staying below a total duration of 2 s on most currently deployed smart cards. We argue this to be a reasonable and applicable tradeoff between authentication performance and authentication delay, while keeping acceleration based gait biometrics secure with MOC.

7. CONCLUSIONS

In this paper we presented an approach towards match-on-card 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 further adapt and simplify both used features and models. The model is computed only once using a database of the corresponding biometrics – then stored on the SC of mobile devices intended for authentication. Enrollment on mobile devices involves recording samples of the authorized user and storing their feature vectors on the SC, 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.

We applied our approach to acceleration based mobile gait authentication using a Java Card with 16 bit integer calculation range. Using a mobile acceleration gait dataset we found our approach to require 77 byte of storage on the SC for the the offline computed model, and 75 byte per gait step cycle. With 8 cycles in the enrolled template this leads to a total of 677 byte of storage requirement on the SC. When also using 8 newly recorded cycles for authentication, leading to a total of 64 comparisons performed on the SC, we found our approach to be feasible with a true positive rate of 0.899 and true negative rate of 0.868. Authentication time on the SC thereby stays below 2 s, including data transmissions and authentication computation. To the best of our knowledge, this work thereby represents the first practical approach towards acceleration based gait match-on-card authentication. One advantage of the proposed approach is that it can conceptually be applied on different biometrics alike, thereby possibly facilitate the translation of matching procedures of other biometrics to match-on-card. The majority of changes would be in preprocessing and feature derivation, while offline model computation as well as feature and model simplification would likely be similar – which could be investigated in future research.

8. ACKNOWLEDGMENTS

This work has 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.

9. REFERENCES

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

[2] N. Belgacem, A. Ali, R. Fournier, and F. Bereksi-Reguig. ECG based human authentication using wavelets and random forests. *International Journal on Cryptography and Information Security (IJCIS)*, 2(2):1–11, 2012.

[3] S. Bistarelli, F. Santini, and A. Vaccarelli. An asymmetric fingerprint matching algorithm for Java Card TM. *Pattern Analysis and Applications*, 9(4):359–376, 2006.

[4] T. Bourlai, K. Messer, and J. Kittler. Face verification system architecture using smart cards. In *Proc. ICPR 2004*, volume 1, pages 793–796, Aug. 2004.

[5] P. Bours and R. Shrestha. Eigensteps: A giant leap for gait recognition. In *Security and Communication Networks (IWSCN) 2010*, pages 1–6, May 2010.

[6] 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*, 44(3):136–147, 1997.

[7] 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 (FM) 2015*, volume

9109 of *Lecture Notes in Computer Science*, pages 90–107. Springer, 2015.

[8] K. Cao and A. Jain. Learning fingerprint reconstruction: From minutiae to image. *IEEE Information Forensics and Security*, 10(1):104–117, Jan. 2015.

[9] 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. *Electronics and Telecommunications Research Institute Journal (ETRI)*, 28(3):320–328, June 2006.

[10] A. Czajka, P. Strzelczyk, M. Chochowski, and A. Pacut. Iris recognition with match-on-card. In *Proc. European Signal Processing Conference (EUSIPCO)*, pages 189–192, Poznan, Poland, Sept. 2007.

[11] I. Daubechies. Orthonormal bases of compactly supported wavelets ii. variations on a theme. *SIAM Journal on Mathematical Analysis*, 24(2):499–519, 1993.

[12] R. D. Findling, M. Muaaz, D. Hintze, and R. Mayrhofer. Shakeunlock: Securely transfer authentication states between mobile devices. *IEEE Transactions on Mobile Computing (TMC)*, 2016. *IEEE early access*.

[13] D. Gafurov. *Performance and Security Analysis of Gait-based User Authentication*. PhD thesis, Faculty of Mathematics and Natural Sciences at the University of Oslo, 2008.

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

[15] D. Gafurov, E. Snekkenes, and P. Bours. Gait authentication and identification using wearable accelerometer sensor. In *Automatic Identification Advanced Technologies*, pages 220–225, June 2007.

[16] 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*, 43(3):1027–1038, 2010.

[17] 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*, pages 845–853. Springer, Berlin, Heidelberg, 2003.

[18] M. Govan and T. Buggy. A computationally efficient fingerprint matching algorithm for implementation on smartcards. In *Biometrics: Theory, Applications, and Systems (BTAS) 2007*, pages 1–6, Sept. 2007.

[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). Technical report, Information Access Division, National Institute of Standards and Technology (NIST), May 2009.

[20] M. R. Hestbek, C. Nickel, and C. Busch. Biometric gait recognition for mobile devices using wavelet transform and support vector machines. In *Proc. Systems, Signals and Image Processing (IWSSIP)*, pages 205–210, Apr. 2012.

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

[22] R. K. Ibrahim, E. Ambikairajah, B. Celler, N. H. Lovell, and L. Kilmartin. Gait patterns classification using spectral features. In *Signals and Systems Conference (ISSC) 2008*, pages 98–102, June 2008.

[23] ISO. *Identification cards – Integrated circuit cards – Part 4: Organization, security and commands for interchange*. 2005.

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

[25] A. K. Jain, B. F. Klare, and A. Ross. Guidelines for best practices in biometrics research. In *International Conference on Biometrics (ICB)*, volume 8, Phuket, Thailand, May 2015.

[26] A. K. Jain and K. Nandakumar. Biometric authentication: System security and user privacy. *IEEE Computer*, 45(11):87–92, 2012.

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

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

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

[30] P. Kocher, J. Jaffe, and B. Jun. Differential power analysis. In *Advances in Cryptology CRYPTO99*, pages 388–397, 1999.

[31] J. R. Kwapisz, G. M. Weiss, and S. A. Moore. Cell phone-based biometric identification. In *Biometrics: Theory Applications and Systems (BTAS) 2010*, pages 1–7, Sept. 2010.

[32] K. Lee and H. Byun. A new face authentication system for memory-constrained devices. *IEEE Consumer Electronics*, 49(4):1214–1222, Nov. 2003.

[33] A. Mannini and A. M. Sabatini. Machine learning methods for classifying human physical activity from on-body accelerometers. *Sensors*, 10(2):1154–1175, 2010.

[34] J. Mäntyjärvi, M. Lindholm, E. Vildjiounaite, S.-M. Mäkelä, and H. Ailisto. Identifying users of portable devices from gait pattern with accelerometers. In *Proc. IEEE Acoustics, Speech, and Signal Processing (ICASSP) 2005*, volume 2, pages ii–973, 2005.

[35] M. Martinez-Diaz, J. Fierrez-Aguilar, F. Alonso-Fernandez, J. Ortega-Garcia, and J. Siguenza. Hill-climbing and brute-force attacks on biometric systems: A case study in match-on-card fingerprint verification. In *Proc. IEEE Security Technology*, pages 151–159, Oct. 2006.

[36] L. Middleton, A. A. Buss, A. Bazin, and M. S. Nixon. A floor sensor system for gait recognition. In *Proc. IEEE Automatic Identification Advanced Technologies (AutoID) 2005*, pages 171–176, Oct. 2005.

[37] 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*, pages 691–698, Dec. 2010.

[38] 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. Information Security and Assurance (ISA) 2008*, pages 307–311, Apr. 2008.

[39] M. Muaaz and R. Mayrhofer. An analysis of different approaches to gait recognition using cell phone based accelerometers. In *Proc. MoMM 2013*, pages 293:293–293:300, NY, USA, 2013. ACM.

[40] M. Muaaz and R. Mayrhofer. Orientation independent cell phone based gait authentication. In *Proc. MoMM 2014*, pages 161–164, NY, USA, 2014. ACM.

[41] M. Muaaz and R. Mayrhofer. Cross pocket gait authentication using mobile phone based accelerometer sensor. In *Proc. Computer Aided Systems Theory (EUROCAST) 2015*, pages 731–738, Las Palmas de Gran Canaria, Spain, Feb. 2015. Springer.

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

[43] C. Nickel. *Accelerometer-based Biometric Gait Recognition for Authentication on Smartphones*. PhD thesis, Technische Universität Darmstadt, 2012.

[44] V. Niennattrakul and C. A. Ratanamahatana. Learning DTW global constraint for time series classification. *CoRR*, abs/0903.0041, 2009.

[45] 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*, 49(2):453–459, May 2003.

[46] S. Prabhakar, S. Pankanti, and A. Jain. Biometric recognition: security and privacy concerns. *IEEE Security Privacy*, 1(2):33–42, Mar. 2003.

[47] S. Rahati, R. Moravejian, and F. M. Kazemi. Gait recognition using wavelet transform. In *Proc. Information Technology: New Generations (ITNG) 2008*, pages 932–936, Apr. 2008.

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

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

[50] H. Sakoe and S. Chiba. Dynamic programming algorithm optimization for spoken word recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 26(1):43–49, Feb. 1978.

[51] 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 Transactions on Pattern Analysis and Machine Intelligence*, 27(2):162–177, Feb. 2005.

[52] A. Savitzky and M. J. E. Golay. Smoothing and differentiation of data by simplified least squares procedures. *Analytical Chemistry*, 36(8):1627–1639, 1964.

[53] 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*, volume 5, pages 3007–3010, July 2005.

[54] 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*, 5(11):369–378, Nov. 2009.

[55] M. Tamviruzzaman, S. I. Ahamed, C. S. Hasan, and C. O'brien. ePet: When cellular phone learns to recognize its owner. In *Proc. Assurable and Usable Security Configuration (SafeConfig) 2009*, pages 13–18, NY, USA, 2009. ACM.

[56] M. Tistarelli and E. Grosso. Active vision-based face authentication. *Image and Vision Computing*, 18(4):299–314, 2000.

[57] U. Uludag and A. K. Jain. Attacks on biometric systems: a case study in fingerprints. In *Security, Steganography, and Watermarking of Multimedia Contents VI*, volume 5306 of *Society of Photo-Optical Instrumentation Engineers (SPIE)*, pages 622–633, June 2004.

[58] 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*, pages 10:1–10:14, NY, USA, 2013. ACM.

[59] D. Vermoen, M. Witteman, and G. N. Gaydadjiev. Reverse engineering java card applets using power analysis. In *Proc. Information Security Theory and Practices (IFIP) 2007*, pages 138–149. Springer, 2007.

[60] B. Vibert, C. Rosenberger, and A. Ninassi. Security and performance evaluation platform of biometric match on card. In *2013 World Congress on Computer and Information Technology (WCCIT)*, pages 1–6, June 2013.

[61] X. Wang, Y. Li, and F. Qiao. Gait authentication based on multi-criterion model of acceleration features. In *Proc. Modelling, Identification and Control (ICMIC) 2010*, pages 664–669, July 2010.

[62] D. Winter. *Biomechanics and Motor Control of Human Movement*. Wiley, 2004.