Feature Selection by User Specific Feature Mask on a Biometric Hash Algorithm for Dynamic Handwriting

Karl Kümmel, Tobias Scheidat, Christian Arndt, Claus Vielhauer

Feature Selection by User Specific Feature Mask on a Biometric Hash Algorithm for Dynamic Handwriting

Karl Kümmel, Tobias Scheidat, Christian Arndt and Claus Vielhauer

Brandenburg University of Applied Sciences, PO box 2132,
14737 Brandenburg, Germany
{kuemmel, scheidat, arndtch, claus.vielhauer}@fh-brandenburg.de

Abstract. One of the most important requirements on a biometric verification system, beside others (e.g. biometric template protection), is a high user authentication performance. During the last years a lot of research is done in different domains to improve user authentication performance. In this work we suggest a user specific feature mask vector *MV* applied on a biometric hash algorithm for dynamic handwriting to improve user authentication and hash generation performance. *MV* is generated using an additional set of reference data in order to select/deselect certain features used during the verification process. Therefore, this method is considered as a simple feature selection strategy and is applied for every user within the system. In our first experiments we evaluate 5850 raw data samples captured from 39 users for five different semantics. Semantics are alternative written content to conceal the real identity of a user. First results show a noticeable decrease of the equal error rate by approximately *three* percentage points for each semantic. Lowest equal error rate (*5.77%*) is achieved by semantic *symbol*. In the context of biometric hash generation, the reproduction rates (RR) increases by an average of approx. *26%*, whereas the highest RR (*88.46%*) is obtained by semantic *symbol* along with a collision rate (CR) of *5.11%*. The minimal amount of selected features during the evaluation is *81* and the maximum amount is *131* (all available features).

Keywords: biometrics, dynamic handwriting, biometric hashing, user bitmask, feature selection

1 Introduction

Today, biometric user authentication is an important field in IT security. It relies on individual biological or behavioral characteristics of a person. The purpose of a generic biometric system is to identify and/or verify a person's identity based on at least one biometric modality (i.e. fingerprint, iris, voice). For all biometric systems, it is crucial to protect the biometric reference data (template) in order to avoid misuse of individual and personal data. However, biometric templates cannot be easily protected by common cryptographic hash algorithms, like they are used in ordinary password authentication systems. The biometric intra-class variability has to be taken into account to ensure reproducibility and protection of a template. The problem of biometric template protection is a frequently discussed issue in biometrics [1]. One

possibility to ensure reproducibility and simple template protection is for example the Biometric Hash algorithm for dynamic handwriting introduced in [2]. The aim of this method is to transform intra-subject biometric data, into stable and individual hash vector values; an overview is given in section 2. In this work we focus on the authentication performance and robustness of this particular biometric hash algorithm.

During the last years a lot of research in almost every biometric authentication algorithm and modality is done to improve user authentication performance. Hollingsworth et al. introduce in [4] a method where potential fragile iris code bits are masked to increase the separation between the match and non-match distributions in iris based authentication systems. Fratric et al. propose in [5] a method of feature extraction from face images to improve recognition accuracy. They use a so-called local binary linear discriminant analysis (LBLDA), which combines the good characteristics of both LDA and local feature extraction methods. Biometric fusion is another technique to improve user authentication performance. Rathgeb et al. describe in [6] a generic fusion technique for iris recognition at bit-level (called Selective Bit Fusion) to improve accuracy and processing time.

Another method, besides many others, is the improvement of the authentication performance by determination of useful features during a feature selection process. In this context, useful features are features which positively affect the user authentication and biometric hash generation performance. Kumar et al. show in [7] that an evaluation and selection of useful biometric features can improve the recognition accuracy. They used a correlation based feature selection (CFS) for bimodal biometric systems and analyzed the classification performance. Makrushin et al. compare in [8] different feature selection strategies to determine sophisticated features. It has been shown that forward and backward selection algorithms have always better results than considered heuristics.

In this work we suggest a much simpler way of feature selection as described in [8]: We apply a user specific feature mask on a biometric hash algorithm for dynamic handwriting to select and/or deselect specific features in order to improve the authentication performance as well as the generation of stable individual biometric hashes.

The structure of the paper is composed as follows. In section 2 we give an overview on the Biometric Hash algorithm for dynamic handwriting. A simple user specific feature mask generation method is introduced in section 3. Experimental results are shown and discussed in section 4. In the last section we present a conclusion and our future work based on the findings.

2 Biometric Hash Algorithm

The Biometric Hash algorithm for dynamic handwriting (hereafter BioHash) is initially introduced by Vielhauer et al. in [2] and enhanced in [3]. During the enrollment process the BioHash algorithm generates a so-called Interval Matrix *IM* for each user. The *IM* is based on raw data of the writer and several parameters. The raw data of each dynamic handwriting sample consists of a time dependent sequence of physical values derived from a digitizer device (e.g. Tablet PC, signature tablet).

Generally, there are five values per sample point: pen tip positions $x(t)$ and $y(t)$, pen tip pressure $p(t)$ and pen orientation angles altitude $\Phi(t)$ and azimuth $\Theta(t)$. From each raw data sample derived from a person during the enrollment process, a statistical feature vector (static and dynamic features) is calculated with a dimensionality of k ($k=131$ in the implementation used in this paper). The *IM* consists of a vector containing the length of a mapping interval for each feature and an offset vector. Both vectors are calculated based on an analysis of intra-class variability of the user using his/her statistical feature vectors.

There are two possibilities to parameterize the hash generation by scaling the mapping intervals stored in the *IM*: Tolerance Vector *TV* and Tolerance Factor *TF*. The aim of the *TV* is to provide a scaling of the mapping interval of each statistical feature separately. Thus, the dimensionality of *TV* is also *k*. *TV* can be calculated individually for each user or globally by a group of users, e.g. either based on all or a selection of enrolled persons, but also on a disjoint group. In contrast to the Tolerance Vector, the Tolerance Factor *TF* is a global hash generation parameter, which is a scalar value. Using the *TF*, it is possible to scale the mapping intervals for all features globally by the same factor.

Based on one statistical feature vector derived from the enrollment data and the users' individual *IM* the so-called interval mapping function determines the reference hash vector b_{ref} of a user. Therefore, the feature dependent interval lengths and offsets provided by *IM* are used to map each of the *k* statistical features to the corresponding hash value. Each further biometric hash is calculated in the same manner, independently if it is used for biometric verification or hash generation application. For verification, the hash vector *b* derived from the currently presented handwriting sample is compared against the reference hash vector b_{ref} by some distance measurement. For more details of the single calculation steps, the interested reader is referred to reference [2].

3 User Specific Feature Mask Generation

In addition to the reference BioHash vector b_{ref} and the corresponding Interval Matrix *IM*, we generate a *k* dimensional ($k=131$) feature mask vector *MV* for each user. *MV* is created during the feature selection process after the enrollment. The main idea of creating a feature mask vector is to select or deselect specific features. If a bit is set to *1*, the represented feature is considered during the verification process and if it is set to *0*, it is not taken into account. This method allows an uncomplicated user specific enabling or disabling of used features.

3.1 Selection Strategy

A user specific feature mask vector *MV* is generated using the reference BioHash vector b_{ref} and corresponding *IM*. Furthermore, raw data samples $s_0, s_1, \ldots, s_n$, which are not used during the enrollment process, are required. The identifier *n* represents the maximum amount of used samples. The feature selection (*MV* generation) is done in three steps. Firstly, the *k*-dimensional feature vectors $fv_0, fv_1, \ldots, fv_n$ are determined

from all raw data samples s_0, s_1, …, s_n. Secondly, feature vectors of each user are mapped to the biometric hash vectors b_0, b_1, …, b_n using the reference Interval Matrix *IM* of this specific user. In the last step the feature mask vector *MV* is generated for each user. Therefore a element-wise comparison is done using the reference BioHash b_{ref} and BioHashes b_0, b_1, …, b_n of the specific user. If a certain number of values at position *i* is equal, the corresponding *i*-th bit of *MV* is set to *1*; otherwise it is set to *0*. We define a so-called similarity threshold th_s as the maximum number of allowed differences between the *i*-th element of the hashes. If th_s is set to *0*, all values have to be equal; if th_s set to *1*, only one different value is allowed and so on. In the end, the result of the *MV* generation is a *k*-dimensional feature mask vector *MV*. The *MV* is a new part of the reference data (template) and is therefore stored together with the corresponding Interval Matrix *IM* and BioHash b_{ref}, for example in a database or on a Smart Card. In our first implementation we only generate the *MV* once for each user during the enrollment process.

Figure 1 exemplarily shows the *MV* generation using only three short BioHashes (b_1, b_2 and b_3) to demonstrate the procedure. In this example we use a threshold th_s of *0* and the character "∧" (logical conjunction) represents the element-wise comparison between the three vectors.

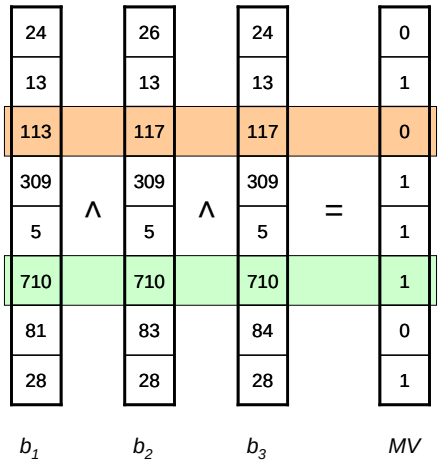

Fig. 1. Example of MV generation during the enrollment process

3.2 Feature Mask Vector and Verification

During a regular verification without feature mask vector consideration, the hash vector b_{cur} derived from the currently presented handwriting sample is compared against the reference hash vector b_{ref} by some distance measurement. In the current configuration, we use the Hamming Distance combined with the feature mask vector as measurement. If two *k*-dimensional BioHashes b_{ref} and b_{cur} are compared using the Hamming Distance, an intermediate result in terms of a *k*-dimensional bit vector *HV* is calculated. The number of all *ones* inside this vector is equal to the Hamming

Distance of the two *k*-dimensional vectors b_{ref} and b_{cur}. In order to include the feature selection results stored inside the feature mask vector *MV*, a *k*-dimensional vector HV_{MV} is calculated by determining the result of the AND (logical conjunction) operation of the two vectors *MV* and *HV*. In the end, the Hamming Distance of b_{ref} and b_{cur}, considering the feature mask vector *MV*, is the sum of all *ones* of the HV_{MV} vector. Therefore, the maximum Hamming Distance value depends not only on the dimensionality of a vector but also on the number of *ones* of the feature mask vector. Figure 2 shows the effect of the *MV* during the verification process; only short BioHashes are used to demonstrate the procedure. The result of this simple example, calculating the Hamming Distance of b_{ref} and b_{cur} using the *MV*, is $HD_{MV} = 1$.

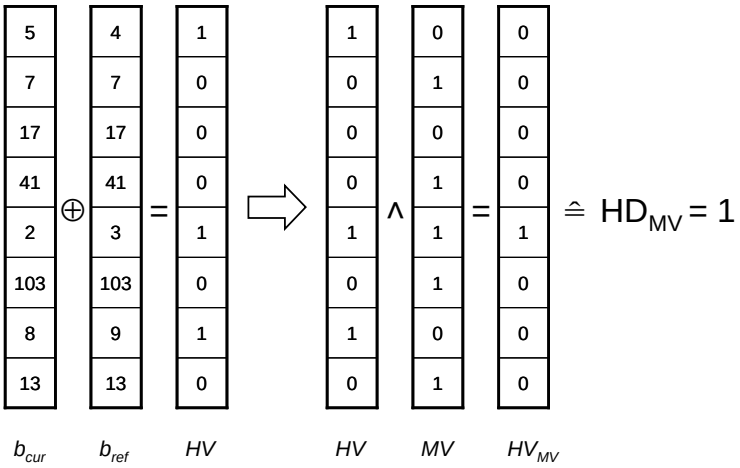

Fig. 2. Example of MV used during the verification process

4 Experimental Evaluation

In this section we show our first experiments on applying a user specific feature mask vector on the Biometric Hash algorithm for dynamic handwriting. Our goal is to compare performance of user authentication and biometric hash generation of the algorithm with and without a feature mask vector. In this section, first we define experimental settings. Secondly, we introduce our methodology to provide a comparative study of the achieved results. Thirdly, experimental results are presented and discussed.

4.1 Experimental Settings

The biometric database of our initial tests consists of *39* subjects, which have donated *30* handwriting samples in three sessions with an interval of at least one month between two sessions. Within a session a user provides *10* handwritten samples for

five different semantics (5850 test samples overall). These semantics are “Free chosen Pseudonym” (*pseudonym*), “Free chosen Symbol” (*symbol*), “Answer to the Question: Where are you from?” (*place*), “Fixed 5 digit PIN: 77993” (*public PIN*) and “Free chosen 5 digit PIN” (*secret PIN*). It has been observed in [2] that semantics produce similar recognition accuracy compared to handwriting signatures, without disclosing the true identity of the writer. All samples were captured under laboratory conditions using a Toshiba M200 Portege tablet PC. The handwriting samples acquired during the first session are used as enrollment data in order to determine the reference BioHash b_{ref} as well as to generate the Interval Matrix *IM*. The samples of the second session are used for tuning of the Tolerance Factor *TF* and feature selection in terms of feature mask vector calculation. Finally, the data collected within the third session are used for evaluation. An attempt of one user to be verified as another one is considered as an imposter trial. Each test implies *390* genuine trials, where reference data of a user is matched against its own verification data (*39* user times *10* test samples) and *14,820* imposter trials, where reference data of a user is matched against all other verification data except its own (*38* user claims times *39* actual users times *10* test samples). Within the feature extraction process of the BioHash algorithm *131* features are calculated based on the handwritten samples.

4.2 Methodology

In our first test we compare the performance of the BioHash algorithm with and without a user specific feature mask vector. Therefore, biometric error rates FRR/FAR, and EER are calculated for both settings. The false rejection rate (FRR) describes the ratio between the number of false rejections of authentic persons and the total number of tests. The FAR (false acceptance rate) is the ratio between number of false acceptances of non-authentic persons and the entire number of authentication attempts. For a comparative analysis of the verification performance, the equal error rate (EER) is a common measurement in biometrics. EER denotes the point in error characteristics, where FRR and FAR yield identical value.

We also evaluate the reproducibility rate (RR) and collision rate (CR) for both settings; these values are related sums of identical reproduced hashes in genuine and imposter trials (see [9]). Because of the reciprocal effect of RR and CR, a tuning of the system to improve RR leads to a degradation of CR and vice versa. Therefore, the collision reproduction rate (CRR, [8]) is selected as a hash generation quality criterion. The CRR is defined in the following equation, whereas CR and RR are weighted equally.

$$CRR = \frac{1}{2}(CR + (1 - RR)) \qquad (\mathbf{1})$$

The tolerance vector *TV* is set to (*1*, …, *1*) since all features are considered equally. Thus, the tolerance factor (*TF*) is the main parameter for controlling CR and RR. In previous work [8] we already determined tolerance factor values of the same evaluation data for two scenarios, lowest EER (EER mode) and highest RR (CRR mode), in all semantics. According to these results of the previous test, based on all *131* features, the *TF* values are set as shown in table 1.

Table 1. Tolerance factor (*TF*) values used during the evaluation

Semantic	TF in CRR mode	TF in EER mode
Public PIN	1.50	1.00
Secret PIN	1.75	1.00
Pseudonym	2.50	1.25
Symbol	3.50	1.50
Place	2.50	1.25

Feature mask vectors are generated for each user in all semantic classes separately, as described in section 3.2, using the evaluation data of the second session. During the *MV* generation, only if all values at a specific position *i* are equal, then MV_i is set to *1*. Therefore the similarity threshold th_s is set to *0*. The minimal, average and maximal amounts of selected features are determined to show how many features are actually used during the verification or hash generation process. Note that the evaluation protocol leads to a realistic scenario since the reference data has already undergone an aging of at least 2 month compared to the evaluation data.

In our first evaluation we do not consider the slightly increased computational effort which is caused by the *MV* calculation during the enrollment (*MV* creation) and verification. Compared to the feature extraction processing time it is negligible.

4.3 Experimental Results

Table 2 shows EERs of all semantics with and without use of the user specific feature mask vector *MV*. The results point out that the EER decreases approximately by *three* percentage points in all semantic classes when the user specific feature mask vector *MV* is applied. The lowest EER is obtained by the semantic *symbol* and is marked in bold letters (*5.77%*). Semantic *public PIN* obtains the highest EER (*12.86%*), which might be caused due to the same written content of all users.

Table 2. Equal error rates (EER) of all semantic classes with and without applied *MV*.

	Public PIN	Secret PIN	Pseudonym	Symbol	Place
No MV	16.55 %	13.42 %	10.96 %	8.30 %	9.79 %
MV	12.86 %	10.86 %	6.58 %	**5.77 %**	7.09 %

Collision reproduction rates (CRR), reproduction rates (RR) and collision rates (CR) for all semantics (with and without *MV*) are shown in table 3. First results indicate that for all semantics the CRR decreases when a *MV* is applied. Therefore, the reproduction rate increases as well as the collision rate. A maximum reproduction rate (RR) of *88.46%* is obtained by the semantic *symbol* with a collision rate (CR) of *5.11%*. An average RR increase of approx. 26% is observed for all semantics, whereas the largest increase is obtained by semantic *public PIN* (from *48.72%* up to *71.03%*).

Table 3. Collision reproduction rates (CRR), reproduction rates (RR) and collision rates (CR) of all semantic classes with and without user specific feature mask vector *MV*.

Semantic	No MV			MV		
	CRR	RR	CR	CRR	RR	CR
Public PIN	27.81%	48.72%	4.33%	18.15%	71.03%	7.33%
Secret PIN	24.27%	55.64%	4.18%	16.76%	73.33%	6.84%
Pseudonym	18.57%	66.67%	3.79%	10.59%	83.85%	5.03%
Symbol	11.19%	82.31%	4.70%	**8.33%**	**88.46%**	5.11%
Place	19.15%	65.38%	3.68%	10.22%	84.87%	5.31%

Table 4 shows the minimal, average and maximal amount of selected features represented by the feature mask vector in each semantic class for both scenarios (verification and hash generation mode). The minimal amount (*81*) of features used during a verification process is obtained by semantic *secret PIN* within the EER mode. In CRR mode the number of used features is always higher than in EER mode. The average amount of selected features over all semantics in EER mode is *122* and in CRR mode *128*.

Table 4. Minimal, average and maximal amount of selected features for each semantic in both scenarios (verification and hash generation mode)

	Public PIN		Secret PIN		Pseudonym		Symbol		Place	
Mode	EER	CRR	EER	CRR	EER	CRR	EER	CRR	EER	CRR
Min.	86	96	**81**	100	103	116	103	116	103	121
Avg.	120	125	120	128	122	128	125	129	122	128
Max.	130	131	130	131	131	131	131	131	131	131

5 Conclusion and Future Work

In this work we introduce a simple feature selection method applied on a biometric hash algorithm for dynamic handwriting. A generated user specific feature mask vector *MV* is used to switch specific features on or off, which are used during the verification or hash generation process. By analyzing the results, we come to a first conclusion that the application of feature mask vector *MV* leads to improved recognition accuracy. In our tests, the equal error rates (EER) decreases in all semantics noticeable by approximately *three* percentage points. Furthermore, the reproducibility of generated biometric hashes increases in all tests considerable. The average increase of the reproduction rate (RR) is approx. *26%*, whereas the highest RR was achieved by the semantic *symbol* (*88.46%*) and the highest rise of the RR (from *48.72%* up to *71.03%*) was reached by the semantic *public PIN*. These results show that a simple feature selection strategy is able to substantial increase the biometric hash generation as well as the user authentication performance.

In future work we will verify our first results by using additional test subjects and study the effects on a non-binary *MV*. A dynamic adaption of the *MV* is also considered in future works, where the *MV* is adapted after each successful verification

attempt. Due to the reduction of relevant features to a specific user, caused by the feature mask vector *MV*; we will investigate the security issue on this side effect. Especially the advantages an attacker gains if he/she is in possession of a *MV* and reference data will be studied. The side effect also leads to reduced entropy and therefore to a potential reduction of a cryptographic key length, if the generated biometric hash is used as a basis for cryptographic key generation.

Acknowledgments. This work is supported by the German Federal Ministry of Education and Research (BMBF), project "OptiBioHashEmbedded" under grant number 17N3109. The content of this document is under the sole responsibility of the authors. We also like to thank Prof. Jana Dittmann of the Otto-von-Guericke University Magdeburg and the StepOver GmbH for supporting the project "OptiBioHashEmbedded".

References

1. Jain, A. K., Nandakumar, K., Nagar, A.: Biometric Template Security, in EURASIP Journal on Advances in Signal Processing, Article ID 579416 (2008)
2. Vielhauer, C.: Biometric User Authentication for IT Security: From Fundamentals to Handwriting, Springer, New York (2006)
3. Vielhauer, C., Steinmetz, R., Mayerhöfer, A., "Biometric Hash based on Statistical Features of Online Signature", Proc. of the Intern. Conf. on Pattern Recognition (ICPR), Quebec City, Canada, Vol. 1 (2002)
4. Hollingsworth, K.P., Bowyer, K.W., Flynn, P.J.: The best bits in an iris code. IEEE Trans. on Pattern Analysis and Machine Intelligence 31(6), 964-973 (2009)
5. Fratric, I., Ribaric S.: Local Binary LDA for Face Recognition. In. Proceedings of the 3rd European Workshop on Biometrics and Identity Management (BioID2011), pp. 144-155, Germay, Brandenburg (2011)
6. Rathgeb, C., Uhl, A., Wild, P.: Combining Selective Best Bits of Iris-Codes. In. Proceedings of the 3rd European Workshop on Biometrics and Identity Management (BioID2011), pp. 127-137, Germany, Brandenburg (2011)
7. Kumar, A., Zhang, D.: Biometric Recognition using Feature Selection and Combination, in LNCS 3546, pp. 813-822 (2005)
8. Makrushin, A., Scheidat, T., Vielhauer, C.: Handwriting Biometrics: Feature Selection based Improvements in Authentication and Hash Generation Accuracy. In. Proceedings of the 3rd European Workshop on Biometrics and Identity Management (BioID2011), pp. 37-48, Germany, Brandenburg (2011)
9. Scheidat, T., Vielhauer, C., Dittmann, J.: Advanced Studies on Reproducibility of Biometric Hashes. In: Proceedings of First Workshop on Biometrics and Identity Management (BIOID 2008), pp. 150-159, Roskilde University, Denmark (2008)