# Feature Selection on Handwriting Biometrics: Security Aspects of Artificial Forgeries

Karl Kümmel[1,2], Tobias Scheidat[1,2], Claus Vielhauer[1] and Jana Dittmann[2]

[1]Brandenburg University of Applied Sciences, Germany
{karl.kuemmel,tobias.scheidat,claus.vielhauer}@fh-brandenburg.de
[2]Otto-von-Guericke University Magdeburg, Germany
{tobias.scheidat,jana.dittmann}@iti.cs.uni-magdeburg.de

**Abstract.** A lot of improvements were introduced lately in order to increase the verification performance of biometric user authentication systems. One method, besides many others, is the selection of specific features for each user during the verification process. In this paper we present a security analysis of a user specific bit mask vector, which was originally introduced to improve verification performance on a Biometric Hash algorithm for dynamic handwriting. Therefore, we use a reverse engineering attack method to generate artificial handwriting data and calculate error rates to examine the impact on the verification performance. Our goal is to study the effect of a feature selection by a mask vector on artificial data in comparison to genuine handwriting data. Our first experimental results show an average decrease of the equal error rate, generate by the artificial data, by approx. *64%*. In comparison, equal error rates of random attacks, using verification data of another user, decreases by an average of approx. *27%*.

**Keywords:** Biometrics, dynamic handwriting, feature selection, security analysis, reverse engineering

## 1 Introduction and Motivation

Biometric user authentication is an important field in IT security today. It relies on personal physiological or behavioral characteristics of a person. The purpose of a generic biometric system is to determine and/or verify a person's identity based on at least one biometric modality (i.e. fingerprint, iris, voice etc.). Like in every other authentication system, i.e. knowledge based (password or PIN), it is crucial to protect the reference data (templates) from being misused. A variety of biometric template protection methods were introduced during the last years to prevent the misuse of biometric data. Jain et al. categorize in [1] a selection of template protection schemes for several biometric modalities and describe their advantages and disadvantages. Besides security issues, authentication performance is also a key requirement to biometric authentication systems. During the last years a lot of research in almost every biometric authentication algorithm and modality is done to improve user authentication performance. Many of which rely on the concept of feature selection. For exam-

ple, Fratric et al. propose in [2] a novel 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 methods LDA and local feature extraction. Hollingsworth et al. introduce in [3] 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.

A further technique to improve user authentication performance is biometric fusion. Rathgeb et al. describe in [4] a generic fusion technique for iris recognition at bit-level (called Selective Bit Fusion) to improve accuracy and processing time. Nageshkumar et al. propose in [5] an authentication method for a multimodal biometric identification system using the two traits face and palmprint.

Specific feature selection is, besides many others, another method to improve authentication performance, whereby useful features are determined during a feature selection process. In this context, useful features are those which positively affect the user authentication and biometric hash generation performance. In [6] Kumar et al. show 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 [7] 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.

We introduced in [8] a much simpler feature selection method which leads to similar findings compared to [7]. A user specific bit mask is generated during the enrollment process to enable/disable certain features within the verification process.

In this work we focus on the security perspective of this user specific bit mask applied on a Biometric Hash algorithm for dynamic handwriting [10] with respect to a specific attack scenario. We analyze whether a potential attack gains any advantages or disadvantages when a user mask vector is used during the verification process. In order to perform a security analysis, we use a reverse engineering attack method introduced in [11] and [12] to check the security affects of a user specific bit mask vector. Since we already observed in [11] and [12] that computer generated handwriting samples of this attack method are not as good as human forgeries, we like to examine if a selective feature approach may affect the false acceptance characteristics of synthetically generated data. Secondly, we like to compare the behavior of the system's verification performance using artificial verification data similar to genuine verification data under feature selection approach.

The structure of the paper is as follows. In section 2 the Biometric Hash algorithm for dynamic handwriting is shortly presented. The user specific feature mask and its generation are described in section 3. A reverse engineering attack method based on a spline interpolation technique is introduced in section 4. Experimental results are shown in section 5 and a conclusion and possible future prospects are given in the last section.

## 2 Biometric Hash Algorithm

The Biometric Hash algorithm for dynamic handwriting (hereafter BioHash) is initially introduced by Vielhauer et al. in [9] and enhanced in [10] in order to generate stable individual biometric hash values as well as to perform biometric verification based on the hashes. Generally, 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). The data consist of five values per sample point: pen tip positions x(t) and y(t), pen tip pressure p(t) and pen orientation angles altitude Φ(t) and azimuth Θ(t).

A so-called Interval Matrix *IM* is determined by the BioHash algorithm during the enrollment process for each user separately. The *IM* calculation is based on raw data of the writer and the parameters Tolerance Factor and Tolerance Vector. From each raw data sample derived from each person during the enrollment process, a statistical feature vector (static and dynamic features) is calculated with a dimensionality of *k* (*k=131* in the reference implementation used in this paper). The *IM* consists of a vector containing the length of a mapping interval for each feature and an offset value vector. Both vectors are calculated based on an analysis of intra-class variability of the user using its statistical feature vectors acquired during enrollment session.

The Biometric Hash algorithm provides two possibilities to parameterize the hash generation by scaling the mapping intervals stored in the *IM*: Tolerance Factor *TF* and Tolerance Vector *TV*. 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 by one global factor. In contrast to the *TF*, the aim of the Tolerance Vector *TV* is to provide an individual 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 a disjoint group of users, but also on all or a selection of enrolled persons.

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 Hamming distance measurement. For more details of the single calculation steps, the interested reader is referred to reference [10].

## 3 Feature Mask Vector

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.

To create a user specific feature mask vector *MV* raw data samples $s_0, s_1, \ldots, s_n$, which are not used during the enrollment process, are required. The identifier *n* indicates the maximum number of used samples. Three steps have to be executed to generate *MV*. Firstly, feature vectors $fv_0, fv_1, \ldots, fv_n$ are determined based on the raw data samples $s_0, s_1, \ldots, s_n$. Secondly, feature vector $fv_0, fv_1, \ldots, fv_n$ of each user are mapped to BioHash vectors $b_0, b_1, \ldots, b_n$ using the corresponding Interval Matrix *IM* of the user. Within the last step, one feature mask vector *MV* for each user are determined by an element-wise comparison of each BioHash vector $b_0, b_1, \ldots, b_n$ and reference BioHash $b_{ref}$. 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*. The result is a *k*-dimensional feature mask vector *MV*. This vector is a new part of the reference data of each user and therefore stored together with corresponding Interval Matrix *IM* and BioHash $b_{ref}$, for example in a database or on a Smart Card. During the verification process only selected features, which are marked by "ones" within *MV*, are considered.

This method allows a simplistic user specific enabling or disabling of used features. We come in [8] to the first conclusion that the application of feature mask vector *MV* leads to improved recognition accuracy. In our tests, the equal error rates (EER see section 5.1) decreases 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 see section 5.1) is approx. *26%*. 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.

## 4 Reverse Engineering based Attack Method

In previous work ([11] and [12]) we introduced a method for constructing biometric raw data from given reference data. This method is based on the following conditions. A potential attacker has compromised a biometric based verification system and has access and knowledge to username, reference BioHash $b_{Ref}$, and Interval Matrix (*IM*) for each registered individual. The operating principle of the BioHash algorithm is openly published and is accessible for everyone who is interested in (Kerkhoff principles). The attacker's aim is to generate synthetic raw data that produces a BioHash $b_{att}$, which is almost identical to the reference BioHash $b_{ref}$. He determines the differences by calculating the Hamming Distance between them. Consequently, he tries to provoke a false-acceptance using his artificially generated raw data.

In [11] we determined the following vulnerability of the Biometric Hash algorithm. When BioHash $b_{ref}$ and corresponding Interval Matrix *IM* are given, a reverse mapping to create a feature vector $fv_{calc}$ can be performed. Due to the fact that $fv_{calc}$ is determined from $b_{ref}$ and corresponding *IM*, it can be mapped with help of *IM* to $b_{ref}$ again and therefore be used to reconstruct raw data, based on it. If an attacker takes

advantage of this vulnerability (reverse mapping) he can reduce his work on reconstructing raw data based on that calculated feature vector $fv_{calc}$, i.e. in feature space rather than on the BioHash.

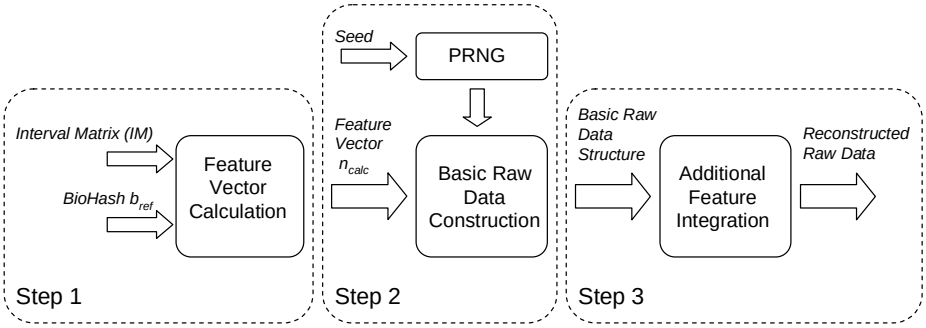


**Fig. 1.** Illustration of steps 1 to 3 of the raw data generation approach

Figure 1 illustrates the three main steps of the attack method to generate raw data. Step 1 implies the calculation of $fv_{calc}$ using reference BioHash $b_{ref}$ and corresponding *IM*. Within step 2 a spline interpolation function generates a basic raw data structure. This is done based on specific feature elements of $fv_{calc}$, which contains the amount of maxima and minima for horizontal pen movement signal X respectively vertical pen movement signal Y. All splines are set randomly using a pseudo random number generator (PRNG), which requires a seed value. The implementation of additional feature into the basic raw data structure is done in step 3. These additional features represent minimum, maximum and average of pressure and angle values. The algorithm simply sets an arbitrary chosen sample point and marks it with the maximum or minimum value. All other sample points are set in a way that the sum matches the average value. This procedure can be done for pressure and both angle values. The result of all three steps is a set of synthetic raw data of an artificial handwriting signal. Due to the reverse engineering algorithm the shape of genuine handwriting signals and artificial handwriting signals do not look similar at all (see figure 2). A detailed description on the algorithm is described in [11] and [12].

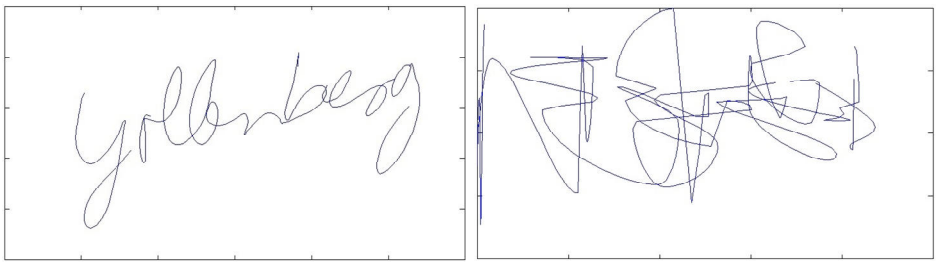

**Fig. 2.** Original genuine handwriting signal (left) and artificial handwriting signal (right)

## 5 Experimental Evaluation

In this section we describe our methodology and setup for the experimental evaluation and show first results on the verification and security performance of the user specific feature mask vector *MV* in context of reverse engineering based attacks.

### 5.1 Methodology

In order to demonstrate the improved verification performance of an applied user specific feature mask vector *MV* on the BioHash algorithm, we compare the verification performance with and without applied *MV*. Afterwards we use the raw data, which is generated by the attack method described in section 4, to test the security impact of an applied *MV*. Therefore, biometric error rates FRR, FAR and EER are calculated for both settings. The ratio between the number of false rejections of authentic persons and the total number of tests is described by the false rejection rate (FRR). The FAR (false acceptance rate) is the ratio between number of false acceptances of non-authentic persons and the entire number of authentication attempts. The equal error rate (EER) is a common measurement in biometrics for a comparative analysis of the verification performance. EER denotes the point in error characteristics, where FRR and FAR yield identical value. Furthermore, we calculate the false acceptance rate caused by the reverse engineering attack method and corresponding equal error rate ($EER_{re}$). The $FAR_{re}$ is the ratio between number of false acceptance of artificially generated imposter data (attack raw data) and the entire number of authentication attempts. Consequently, $EER_{re}$ donates the point in error characteristics where FRR and $FAR_{re}$ yield identical value.

We also evaluate the reproducibility rate (RR) and collision rate (CR) for both settings including the attack data; these values are related sums of identical reproduced hashes in genuine and both imposter trials (see [12]). 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, [12]) is selected as a hash generation quality criterion. The CRR is defined in the following equation, whereas CR and RR are equally weighted.

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

$CR_{re}$ and $CRR_{re}$ describe collision rate and collision reproduction rate of the attack data, which is generated by the reverse engineering attack method.

### 5.2 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 [10] that semantics produce similar recognition accuracy compared to handwriting signatures, without disclosing the true identity of the writer. All samples were captured under similar 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. Ten raw data samples are generated by the attack method for each user based on its reference data BioHash $b_{ref}$ and *IM* (39 users times 10 test samples). These test samples are used to calculate $EER_{re}$, $CR_{re}$, and $CRR_{re}$.

In addition, an attempt of one user to be verified as another one is considered as an imposter trial (random attack). 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 (random attacks), 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.

**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 |

Since all features are considered equally, the tolerance vector *TV* is set to (*1, …, 1*). Thus, the tolerance factor (*TF*) is the main parameter for controlling hash generation as well as verification performance. In previous work [7] 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.

Feature mask vectors are generated for each user in all semantic classes separately, as described in section 3, using the evaluation data of the second session. During the *MV* generation, only if all values at a specific position *i* of all BioHash vectors are equal, then $MV_i$ is set to *1*. 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.

### 5.3 Experimental Results

Table 2 shows equal error rates (EER and $EER_{re}$) of all semantics with and without applied *MV*. By comparing the first columns of each section, where the EER is presented for all semantics, a decrease of all EER is clearly noticeable. The highest drop of *4.67* percentage points (relative drop of *42.61%*) is caused by the semantic pseudonym. Similar results are recorded by the $EER_{re}$. The highest drop of 2.2 percentage points (relative drop of *46.8%*) is achieved by semantic *pseudonym*. Semantic *public PIN* even reaches an $EER_{re}$ of *0%*. In one case a slightly increase of the $EER_{re}$ of 0.2 percentage points occurs (semantic *place*).

Table 3 shows reproduction rates, collision rates and collision reproduction rates of all semantic classes with and without applied specific feature mask vector *MV*. If a MV is used all reproduction rates increases significantly by an average of approximately 19%, whereas the collision rates also increase by an average of *31%*.

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

| Semantic | No MV | | MV | |
|---|---|---|---|---|
| | EER | $EER_{re}$ | EER | $EER_{re}$ |
| Public PIN | 17.46 | 2.56 | 13.25 | 0.00 |
| Secret PIN | 12.71 | 1.41 | 11.54 | 0.25 |
| Pseudonym | 10.96 | **4.70** | **6.29** | 2.50 |
| Symbol | 9.45 | 1.86 | 6.44 | 1.35 |
| Place | 9.79 | 2.30 | 7.09 | 2.50 |

The largest reproduction rate increase is obtained by the semantic *public PIN* (*51.79% up to 72.54%*) and the highest reproduction rate was achieved by semantic symbol (*94.35%*). Collision rates of associated attack test samples are in almost every semantic *zero*. Within semantic symbol a slightly collision rate of *0.76%* is recorded during the experimental tests.

**Table 3.** Collision reproduction rates ($CRR/CRR_{re}$), reproduction rates (RR) and collision rates ($CR/CR_{re}$) of all semantic classes (in %) with and without user specific feature mask vector *MV*.

| Semantic | No MV | | | | | MV | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | RR | CR | $CR_{re}$ | CRR | $CRR_{re}$ | RR | CR | $CR_{re}$ | CRR | $CRR_{re}$ |
| Public PIN | 51.79 | 5.10 | 0.00 | 26.65 | 24.10 | 72.54 | 8.34 | 0.00 | 17.88 | 13.71 |
| Secret PIN | 60.00 | 4.85 | 0.00 | 22.42 | 20.00 | 78.20 | 9.17 | 0.00 | 15.48 | 10.89 |
| Pseudonym | 71.02 | 4.33 | 0.00 | 16.65 | 14.48 | 84.61 | 5.84 | 0.00 | 10.61 | 7.69 |
| Symbol | 86.92 | 5.40 | 0.00 | 9.24 | 6.53 | **94.35** | 6.72 | 0.76 | **6.18** | **3.20** |
| Place | 71.02 | 4.06 | 0.00 | 16.52 | 14.48 | 87.17 | 5.57 | 0.00 | 9.19 | 6.41 |

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 (verifi-

cation and hash generation mode). The minimal amount (*87*) of features used during a verification process is obtained by semantic *public 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. | **87** | 96 | 89 | 107 | 103 | 120 | 113 | 126 | 107 | 123 |
| Avg. | 120 | 126 | 120 | 127 | 123 | 128 | 125 | 130 | 122 | 128 |
| Max. | 129 | 131 | 130 | 131 | 131 | 131 | 131 | 131 | 131 | 131 |

## 6 Conclusion and Future Work

In this work we study the security impact of an applied user specific feature mask vector *MV* introduced in [8] on a Biometric Hash algorithm for dynamic handwriting [10]. Therefore, we use a reverse engineering attack method introduced in [11] and [12] to generate attack test samples for each user. Our goal was to see if the feature mask vector *MV* has any affect on artificial data and if so, are the results similar to genuine verification data. Within the experimental tests, equal error rates for both settings, with and without applied *MV*, were evaluated. First results indicate that an applied feature mask vector reduces the false acceptance rates caused by the attack test samples significantly. Consequently, this leads to a decrease of $EER_{re}$ in almost every semantic class by an average of approx. *64%*. Compared to the decrease caused by the *MV* on the EER of random attacks (average drop by 27%), artificial data are more effected. It seems that an applied feature mask vector has even a greater security impact on artificially generated data then on genuine data and random attacks. Collision rates caused by the artificial generated data (in almost every semantic *0%*) support this assumption. These results point out, that an applied specific feature mask vector improves not only the verification but also the security performance of the Biometric Hash algorithm by reducing error rates imposed by artificial samples. Nevertheless, a slightly increase of an $EER_{re}$ of *0.2* percentage points and a minor collision rate of *0.76%* during an applied *MV* needs to be studied further. In order to substantiate the first experimental results further tests, using more individuals and a greater amount of attack test samples, have to be carried out. Also the reverse engineering method can be improved to generate more efficient imposter data in order to execute a more sophisticated security performance test. One possible research direction here could be a composition of handwriting signals from sets of base letter structure components with the pseudorandom spline function. This method may also lead to more realistic looking artificial handwriting data then those which were generated by the actual reverse engineering method.

**Acknowledgements.** 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 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. 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, Germany, Brandenburg (2011)
3. Hollingsworth, K.P., Bowyer, K.W., Flynn, P.J.: The Best Bits in an Iris Code. IEEE Transactions on Pattern Analysis and Machine Intelligence Volume 31 Issue 6, 964-973 (2009)
4. 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)
5. Nageshkumar, M.; Mahesh, P., Swamy, M. S.: An Efficient Secure Multimodal Biometric Fusion Using Palmprint and Face Image. International Journal of Computer Science Issues, IJCSI, Volume 2, pp. 49-53 (2009)
6. Kumar, A., Zhang, D.: Biometric Recognition using Feature Selection and Combination. In: Audio- and Video-Based Biometric Person Authentication, LNCS 3546, pp. 813-822 (2005)
7. 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)
8. Kümmel, K.; Scheidat, T.; Arndt, C., Vielhauer, C.: Feature Selection by User Specific Feature Mask on a Biometric Hash Algorithm for Dynamic Handwriting. In Proceedings of the 12th Joint IFIP TC6 and TC11 Conference on Communications and Multimedia Security, pp. 85-93 (2011)
9. Vielhauer, C., Steinmetz, R., Mayerhöfer, A., "Biometric Hash based on Statistical Features of Online Signature". In Proceedings of the International Conference on Pattern Recognition (ICPR), Quebec City, Canada, Vol. 1 (2002)
10. Vielhauer, C.: Biometric User Authentication for IT Security: From Fundamentals to Handwriting, Springer, New York (2006)
11. Kümmel, K., Vielhauer, C.: Potentielle Rückführbarkeit eines biometrischen Hashes für Handschriften. In Proceedings of the D-A-CH Security Conference, pp. 66-77, Klagenfurt, Germany (2010)
12. Kümmel, K., Vielhauer, C., Scheidat, T., Franke, D., Dittmann, J.: Handwriting Biometric Hash Attack: A Genetic Algorithm with User Interaction for Raw Data Reconstruction. In Proceedings of the 11th Joint IFIP TC6/TC11 International Conference on Communications and Multimedia Security, pp. 178-190, Austria (2010)
13. Scheidat, T., Vielhauer, C., Dittmann, J.: Advanced Studies on Reproducibility of Biometric Hashes. In Proceedings of the First Workshop on Biometrics and Identity Management (BIOID 2008), pp. 150-159, Roskilde University, Denmark (2008)