On the Vulnerability of Face Verification Systems to Hill-Climbing Attacks

†J. Galbally, C. McCool, †J. Fierrez, S. Marcel, and †J. Ortega-Garcia

IDIAP Research Institute

†Biometric Recognition Group–ATVS, EPS, Universidad Autonoma de Madrid

{christopher.mccool,sebastien.marcel}@idiap.ch

{javier.galbally,julian.fierrez,javier.ortega}@uam.es

Abstract. In this paper, we use a general hill-climbing attack algorithm based on Bayesian adaption to test the vulnerability of two face recognition systems to indirect attacks. The attacking technique uses the scores provided by the matcher to adapt a global distribution computed from an independent set of users, to the local specificities of the client being attacked. The proposed attack is evaluated on an Eigenface-based and a Parts-base face verification system using the XM2VTS database. Experimental results demonstrate that the hill-climbing algorithm is very efficient and is able to bypass over 85% of the attacked accounts (for both face recognition systems). The security flaws flaws of the analyzed system are pointed out and possible countermeasures to avoid them are also proposed.

1 Introduction

Automatic access of persons to services is becoming increasingly important in the information era. This has resulted in the establishment of a new research and technology area known as biometric recognition, or simply biometrics [1]. The basic aim of biometrics is to discriminate automatically between subjects -in a reliable way and according to some target application- based on one or more signals derived from physical or behavioral traits, such as fingerprint, face, iris, voice, hand, or written signature.

Biometric technology presents several advantages over classical security methods that are based on a pass-phrase (Personal Identification Number or Password) or on a physical key (or access card) [2, 3]. A major disadvantage of traditional authentication systems is that they cannot discriminate between impostors who have illegally acquired the privileges to access a system and the genuine user. Furthermore, in biometric systems there is no need for the user to remember difficult PIN codes that could be easily forgotten or to carry a key that could be lost or stolen.

Despite their advantages, biometric systems are still vulnerable to external attacks which could decrease their level of security. Thus, it is of utmost importance to analyze the vulnerabilities of biometric systems, in order to find their limitations and to develop useful countermeasures for foreseeable attacks.

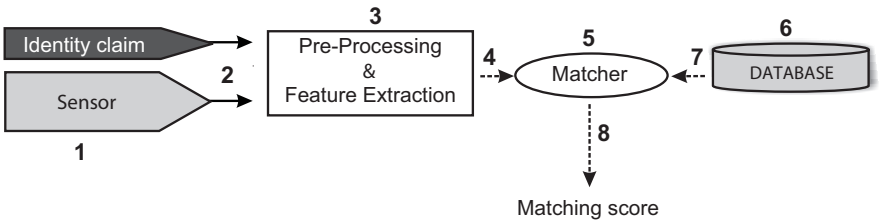

Fig. 1. Architecture of an automated biometric verification system. Possible attack points are numbered from 1 to 8.

In [4] Ratha identified and classified eight possible attack points for biometric recognition systems. These vulnerability points, depicted in Fig. 1, can be broadly divided into two groups:

- **Direct attacks**. In [4] the possibility to generate synthetic biometric samples (for instance, speech, fingerprints or face images) in order to illegally access a system was defined as the first vulnerability point in a biometric security system. These attacks at the sensor level are referred to as *direct attacks* and require no specific knowledge about the system; no knowledge of the matching algorithm, feature extraction process or feature vector format is required. Furthermore, the attack is carried out in the analog domain, outside the digital limits of the system, so digital protection mechanisms (digital signature or watermarking) cannot be used.
- **Indirect attacks**. This group includes all the remaining seven points of attack identified in Fig. 1. Attacks 3 and 5 might be carried out using a Trojan Horse that bypasses the feature extractor and the matcher respectively. In attack 6 the system database is manipulated (a template is changed, added or deleted) in order to gain access to the application. The remaining points of attack (2, 4, 7 and 8) are thought to exploit possible weak points in the communication channels of the system by extracting, adding or changing information from them. In this case the intruder needs to have some additional information about the internal working of the recognition system and, in most cases, physical access to some of the application components (feature extractor, matcher or database) is required.

Previous work has studied the robustness of biometric systems to direct attacks, specifically finger- and iris-based system [5–7]. Some efforts have also been made in the study of indirect attacks to biometric systems. Most of these works use some type of variant of the hill-climbing algorithm presented in [8]. Some examples include an indirect attack to a face-based system in [9], and to standard and Match-on-Card minutiae-based fingerprint verification systems in [10] and [11] respectively. These attacks, which belong to types 2 or 4 in Fig. 1, take advantage of the score given by the matcher to iteratively change a synthetically created template until the similarity score exceeds a fixed decision threshold and thereby gain access to the system.

Most of the hill-climbing approaches are all highly dependent on the technology used, only being usable for a very specific type of matcher. However, in [12] a general hill-climbing algorithm based on Bayesian adaptation was presented and tested using an on-line signature verification system. In the present contribution this attack is successfully applied to two automatic face recognition systems thus proving its biometric independency and its ability to adapt to different matchers which use fixed length feature vectors.

Two case studies are presented in this work where several aspects of the attack are investigated. The first one examines the effectiveness of the technique on an Eigenface-based verification system while the second uses a more advanced Gaussian Mixture Model (GMM) Parts-based approach. For both case studies the experiments are conducted on the XM2VTS database [13] and it is shown that the attack is able to bypass over 85% of the accounts attacked for the best configuration of the algorithm found. Furthermore, the hill-climbing approach is shown to be faster than a brute-force attack for all the operating points evaluated.

The paper is structured as follows. The hill-climbing attack algorithm used in the experiments is described in Sect. 2, while the two attacked systems are presented in Sect. 3. The database and experimental protocol followed are described in Sect 4. The results on the Eigenface-based system and the GMM system are detailed in Sect. 5.1 and Sect. 5.2 respectively. Conclusions are finally drawn in Sect. 6.

2 Bayesian Hill-Climbing Algorithm

Problem statement. Consider the problem of finding a K-dimensional vector $\mathbf{y}^*$ which, compared to an unknown template $\mathcal{C}$ (in our case related to a specific client), produces a similarity score bigger than a certain threshold δ, according to some matching function J, i.e.: $J(\mathcal{C}, \mathbf{y}^*) > \delta$. The template can be another K-dimensional vector or a generative model of K-dimensional vectors.

Assumptions. Let us assume:

- That there exists a statistical model G (K-variate Gaussian with mean $\boldsymbol{\mu}_G$ and diagonal covariance matrix $\boldsymbol{\Sigma}_G$, with $\boldsymbol{\sigma}_G^2 = \text{diag}(\boldsymbol{\Sigma}_G)$), in our case related to a background set of users, overlapping to some extent with $\mathcal{C}$.
- That we have access to the evaluation of the matching function $J(\mathcal{C}, \mathbf{y})$ for several trials of $\mathbf{y}$.

Algorithm. The problem of finding $\mathbf{y}^*$ can be solved by adapting the global distribution G to the local specificities of template $\mathcal{C}$, through the following iterative strategy:

1. Take N samples ($\mathbf{y}_i$) of the global distribution G, and compute the similarity scores $J(\mathcal{C}, \mathbf{y}_i)$, with $i = 1, \ldots, N$.
2. Select the M points (with $M < N$) which have generated highest scores.

3. Compute the local distribution $L(\boldsymbol{\mu}_L, \boldsymbol{\sigma}_L)$, also K-variate Gaussian, based on the M selected points.
4. Compute an adapted distribution $A(\boldsymbol{\mu}_A, \boldsymbol{\sigma}_A)$, also K-variate Gaussian, which trades off the general knowledge provided by $G(\boldsymbol{\mu}_G, \boldsymbol{\sigma}_G)$ and the local information given by $L(\boldsymbol{\mu}_L, \boldsymbol{\sigma}_L)$. This is achieved by adapting the sufficient statistics as follows:

$$\boldsymbol{\mu}_A = \alpha\boldsymbol{\mu}_L + (1-\alpha)\boldsymbol{\mu}_G \tag{1}$$

$$\boldsymbol{\sigma}_A^2 = \alpha(\boldsymbol{\sigma}_L^2 + \boldsymbol{\mu}_L^2) + (1-\alpha)(\boldsymbol{\sigma}_G^2 + \boldsymbol{\mu}_G^2) - \boldsymbol{\mu}_A^2 \tag{2}$$

5. Redefine $G = A$ and return to step 1.

In Eq. (1) and (2), $\boldsymbol{\mu}^2$ is defined as $\boldsymbol{\mu}^2 = \text{diag}(\boldsymbol{\mu}\boldsymbol{\mu}^T)$, and α is an adaptation coefficient in the range [0,1]. The algorithm finishes either when one of the N similarity scores computed in step 2 exceeds the given threshold δ, or when the maximum number of iterations is reached.

In the above algorithm there are two key concepts not to be confused, namely: *i*) number of *iterations* (n_{it}), which refers to the number of times that the statistical distribution G is adapted, and *ii*) number of *comparisons* (n_{comp}), which denotes the total number of matchings carried out through the algorithm. Both numbers are related through the parameter N, being $n_{comp} = N \cdot n_{it}$.

3 Face Verification Systems Attacked

The described Bayesian hill-climbing algorithm is used to test the robustness against this type of attacks of two different face verification system. The first system is based on the Eigenfaces technique [14] and the second system is based on the GMM Parts-based approach [15]:

- **Eigenface-based system**. An Eigenfaces-based system is used as it is a well known technique within the face verification community. It was first introduced by Turk and Pentland in [14] and it was used to present initial results for the recent Face Recognition Grand Challenge evaluation [16].
  The evaluated Eigenfaces-based system uses cropped face images of size 64×80. These images are used to train a PCA vector space where 80% of the variance is retained. This leads to a system where the original image space of 5120 dimensions is reduced to 91 dimensions (or eigenvectors).
  Similarity scores are computed in this PCA vector space using the standard correlation metric. The standard correlation metric, $d(\mathbf{x}, \mathbf{y}) = 1 - [(\mathbf{x} - \mu_\mathbf{x}) \cdot (\mathbf{y} - \mu_\mathbf{y})]/\sigma_\mathbf{x}\sigma_\mathbf{y}$, is used as it showed the best performance out of the tested similarity measures.
- **GMM Parts-based system**. The GMM Parts-based system used in the evaluation tesselates the 64×80 images into 8×8 blocks with a horizontal and vertical overlap of 4 pixels. This tessalation process results in 285 blocks and from each block a feature vector is obtained by applying the Discrete Cosine Transform (DCT); from the possible 64 DCT coefficients only the

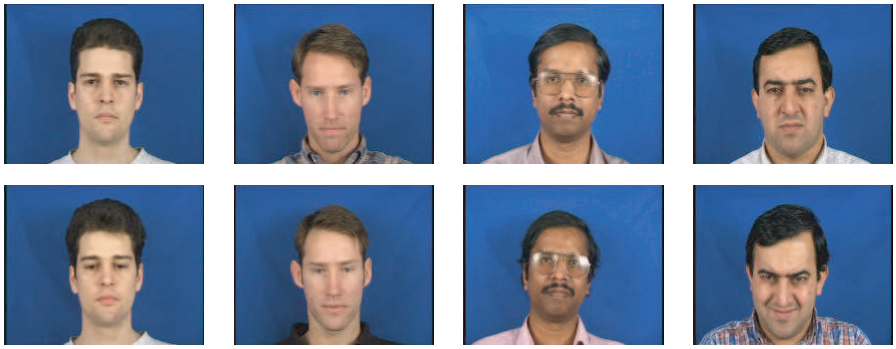

Fig. 2. Examples of the images that can be found in XM2VTS.

first 15 coefficients are retained [15]. The blocks are used to derive a world GMM ω_w and a client GMM ω_c. Experimentation found that using a 512 mixture component GMM gave optimal results.
When performing a query, or match, the average score of the 285 blocks from the input image are used. The DCT feature vector from each block v_i (where $i = [1...285]$) is matched to both Ω_w and Ω_c to produce a log-likelihood score. These scores are then combined using the log-likelihood ratio, $S_{llr,j} = log[P(\Omega_c|v_j)] - log[P(\Omega_w|v_j)]$, and the average of these scores is used as the final score, $S_{GMM} = \frac{1}{285}\sum_{j=1}^{285} S_{llr,j}$. This means that the query template can be considered to be a feature matrix of size 285×15.

4 Experimental Protocol

The experiments are carried out on the XM2VTS face database [13], comprising 295 users. This database was acquired in four time-spaced capture sessions in which two different face images of each client were taken under controlled conditions (pose and illumination) to complete the total $295 \times 8 = 2,360$ samples of the database. Two evaluation protocols are defined for this database, the Lausanne Protocol (LP) 1 and 2 [13]. In Fig. 2 some example images from the XM2VTS database are shown.

4.1 Performance Evaluation

The performance of the evaluated systems is computed based on the LP2 protocol. This protocol is chosen as it provided the most number of samples for training, however, due to the limited number of samples the protocol was changed slightly to maximise the number of samples to estiamte G.

There are two data sets which could be used to estimate the initial distribution G, either the development data or test data. The test data has almost three times the number of impostor samples (when compared to the development

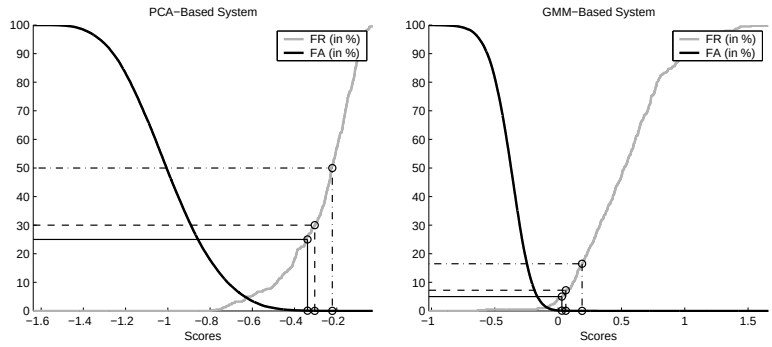

Fig. 3. FAR and FRR curves for the Eigenface-based system (left) and the GMM-based system (right).

data) and consequently the test data set is used to estimate G. The remainder of the protocol is the same, as the enrollment samples are drawn from the training data. Further details on the Attack Protocol are presented in Sect. 4.2.

As a result of using the same subjects for PCA training and client enrollment, the system performance is optimistically biased, and therefore harder to attack than in a practical situation (in which the enrolled clients may not have been used for PCA training). This means that the results presented in this paper are a conservative estimate of the attack's success rate. In Fig. 4 a general diagram showing the LP2 evaluation protocol is given.

The resulting experimental protocol leads to $p = 4$ enrollment for each client. Therefore, the final score produced by Eigenfaces-based system is the ystem is the average of the p scores obtained after matching the input vector to the p templates of the attacked client model $\mathcal{C}$. while in the GMM system the p templates are used to estimate the GMM client model.

The performance of the two face recognition systems is presented in Fig. 3. The system performance is presented in terms of the False Acceptance Rate (FAR) and the False Rejection Rate (FRR) curves for the Eigenfaces-based system (left) and for the GMM Parts-based system (right). The Eigenface-based system has an Equal Error Rate (EER) of 4.74%, while the GMM system shows a better performance with an EER of 1.24%.

Three operating points are used to evaluate the hill climbing algorithm. These points correspond to FAR=0.1%, FAR=0.05% and FAR=0.01% which correspond to low, medium, and high security applications respectively [17].

4.2 Experimental Protocol for the Attacks

Following the LP2 protocol Attack Protocol is as follows:

– the Training set of LP2 is used to compute the PCA transformation matrix, the GMM world model and to enrol clients, and

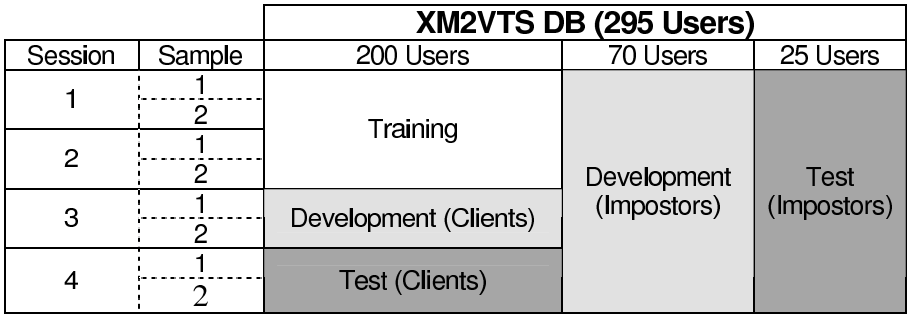

Fig. 4. Diagram showing the partitioning of the XM2VTS database according to the LP2 protocol (which was used in the performance evaluation of the present work).

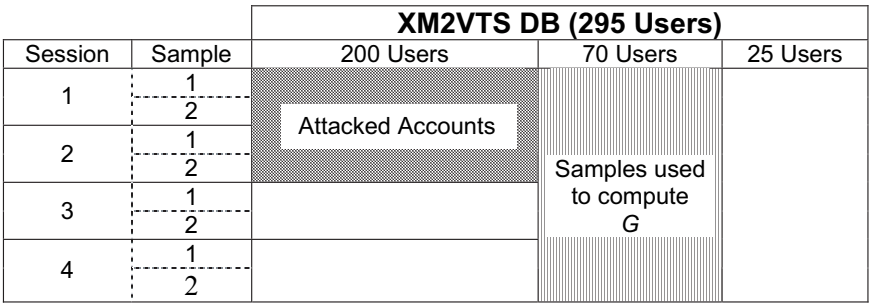

Fig. 5. Diagram showing the partitioning of the XM2VTS database followed in the attacks protocol.

– the Test set of impostors are used to estiamte G.

The Training set of LP2 corresponds to the first two sessions of 200 users and all of this information is used to enrol the user, as well as calculating the PCA transformation matrix and GMM world model. The Test set of impostors corresponds to four sessions of 70 user and this data is used to calculate G.

The initial K-variate distribution G of the algorithm is estimated using the first session of all the impostors of the Test set (70 users) defined in LP2. This ensures there is no overlap between the attacked set of users (200 accounts) and the subjects used to initialize the algorithm; if there was overlap there could be a bias introduced into the success rate (SR) of the attack.

The SR is defined as the number of accounts broken (A_b) by the attack divided by the total number of accounts attacked (A_T), $SR = A_b/A_T$. An account is considered to be broken when the hill-climbing scheme reaches the decision threshold δ and for these experiments there are a total of $A_T = 200$ accounts attacked for these experiments. In Fig. 5 the partitioning of the database used for the attacks is shown.

5 Experiments

The goal of these experiments is to study the vulnerability of automatic face recognition systems to hill-climbing attacks. This is achieved by examinig the effectiveness of the Bayesian-based hill-climbing algorithm in attacking two differet face recognition systems at several operating points. By performing these attacks it will also be demonstrated that the Bayesian-based hill-climbing algorithm can be applied to other biometric traits; it was already shown to be successful in attack an on-line signature verification system [12].

Two case studies are presented for the attacks on the two separate face verification systems. The first case study examines the effectiveness of the Bayesian-based hill-climbing attack on an Eigenface-based system. The second study uses the previously found optimal configuration to attack a GMM Parts-based system. By using the same optimal configuration between studies we can determine if the performance of the attack is highly dependent on the values of the parameters selected.

5.1 Case study 1: Attacking an Eigenface-Based Face Verification System

In the first set of experiments, we study the effect of varying the three parameters of the algorithm (N, M, and α) on the success rate. The attack is performed on the Eigenface-based system (described in Sect. 3) in order to find the optimal configuration by maximizing the number of broken accounts while minimizing the average number of comparisons (n_{comp}) needed to reach the fixed threshold δ.

The parameters optimised in these experiments are:

- $N = [10, 25, 50, 100, 200]$ the number of sampled points of the adapted distribution at a given iteration,
- $M = [3, 5, 10, 25, 50, 100]$ the number of top ranked samples used at each iteration to adapt the global distribution, and
- $\alpha = [0.0, 0.1, \ldots, 1.0]$ the adaptation coefficient.

The importance of the initial distribution G is also examined by evaluating the attack performance when a smaller number of samples is used to compute G, the case where G is randomly selected is also examined.

When presenting results the brute-force approach is used to provide a baseline to compare with the hill-climbing algorithm. We compare n_{comp} the number of matchings necessary for a successful brute-force attack at the operating point under consideration ($n_{bf} = 1/\text{FAR}$). However, the proposed hill-climbing algorithm and a brute-force are not fully comparable because a successful brute-force attack requires much greater resources, for instance a database of thousands of samples is needed.

Table 1. Success Rate (in %) of the hill-climbing attack for increasing values of N (number of sampled points) and M (best ranked points). The maximum number of iterations allowed is given in brackets. The Success Rate (in %) appears in plain text, while the average number of iterations needed to break an account appears in **bold**. The best configuration of parameters N and M is highlighted in grey.

		N				
		10 (2500)	25 (1000)	50 (500)	100 (250)	200 (125)
M	3	84.5 **5,162**	86.0 **4,413**	86.0 **4,669**	86.0 **5,226**	86.0 **6,296**
	5	81.5 **5,796**	86.0 **4,275**	86.0 **4,512**	86.0 **5,022**	86.0 **5,988**
	10		85.5 **4,534**	86.0 **4,540**	86.0 **5,019**	86.0 **5,941**
	25			86.0 **5,213**	86.0 **5,379**	86.0 **6,256**
	50				86.0 **6,455**	86.0 **6,934**
	100					86.0 **8,954**

Analysis of N and M (sampled and retained points). For the initial evaluation of the algorithm an operating point of (FRR=50%, FAR=0.01%) was fixed. This FA implies that an eventual brute-force attack would be successful, on average, after 10,000 comparisons. Given this threshold the algorithm was executed for different values of N and M (fixing $\alpha = 0.5$) and the results are given in Table 1. The maximum number of iterations (n_{it}) allowed for the algorithm appears in brackets. This value changes according to N in order to maintain constant the maximum number of comparisons permitted ($n_{comp} = N \cdot n_{it}$). In plain text we show the success rate of the attack (in % over the total 200 accounts tested), while the average number of comparisons needed for a successful attack is represented in **bold**.

Examining Table 1, the optimal configuration is $[N = 25, M = 5]$. For this point, the number of accounts broken is maximized (86%) and n_{comp} is minimized (4,275). This minimum represents less than half of the expected number of matchings required for a successful brute-force attack ($n_{bf} = 1/\text{FAR} = 10,000$).

Further analysis of Table 1 provides two interesting results. The first is that optimising N is more important than optimising M, this is because N is the number of scores produced at each iteration and consequently has a direct impact on the number of comparisons performed n_{comp}. For instance once N becomes large (¿50) then the number of comparisons increases significantly. The second is that varying M has an impact on performance and that choosing a value such that $M < N$ provides significantly fewer comparisons than if $M \simeq N$. However,

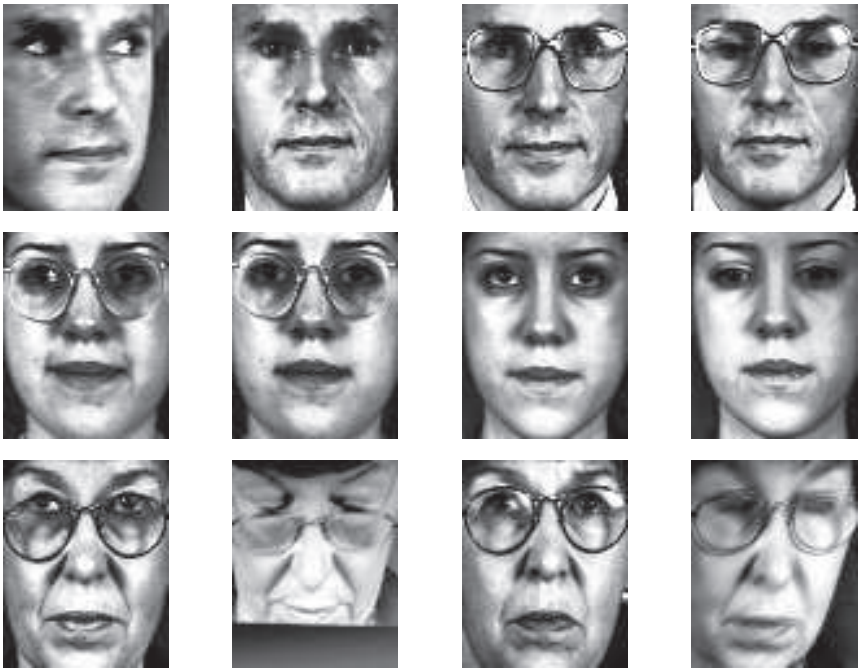

Fig. 6. The four enrollment images (columns) constituting the model of three of the unbroken accounts (rows).

this effect cannot be observed in the success rate of the attack, which is 86% for most of the configurations evaluated (172 broken accounts out of a total of 200.

The 28 clients who remain robust to the attack are the same in all cases. To search for an explanation, the 28 unbroken client models (comprising the four images of the first two database sessions) were matched to the other four images of the user (those corresponding to sessions three and four). None of the client models produced a score high enough to enter the system, which means that these 28 clients would not be suitable for face recognition under the considered system working at the selected operating point. We can then conclude that the attack successfully broke all the models that would be used in a real application. In Fig. 6 the enrollment images which form three of the resistant accounts are shown. In all cases we can observe a great variance among the samples of a given model (glasses/not glasses, different poses, and blurred images).

Analysis of α (adaptation coefficient). For the optimal configuration of N and M the effect of varying α is studied. This value is varied from 0 (only the global distribution G is taken into account) to 1 (only the local distribution L affects the adaptation stage). The results are presented in Table. 2 where the

Table 2. Success Rate (in %) of the hill-climbing attack for increasing values of α and for $[N, M] = [25, 5]$. The Success Rate (in %) appears in plain text, while the average number of iterations needed to break an account appears in **bold**.

α	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
SR(%)	0	84.5	86.0	86.0	86.0	86.0	86.0	81.0	71.5	51	20.0
n_{comp}	**25,000**	**6,468**	**5,121**	**4,617**	**4,381**	**4,273**	**4,380**	**4,990**	**7,901**	**10,404**	**14,154**

Table 3. Success Rate (in %) of the hill-climbing attack for increasing number of samples used to compute the initial distribution G. N,M, and α are set to 25, 5, and 0.5 respectively. The Success Rate (in %) appears in plain text, while the average number of iterations needed to break an account appears in **bold**.

Number of real samples used to compute G							
5	10	35	70	140	280	560	Random (μ=0,σ=1)
86.0	86.0	86.0	86.0	86.0	86.0	86.0	86.0
4353	**4307**	**4287**	**4283**	**4,279**	**4,285**	**4,281**	**4,492**

success rate of the attack appears in plain text (%), while the average number of comparisons needed for a successful attack is shown in **bold**.

From Table. 2 it can be seen that the optimal point is $\alpha = 0.5$. This corresponds to the case where both the global and local distribution are given approximately the same importance. As in the previous experiment, it can be noticed that 14% percent of the accounts (the same 28 clients as in the previous experiments) is never bypassed as a consequence of the large user intra-variability.

Analysis of the initial distribution *G*. In the previous experiments the K-variate initial distribution G was computed using the first two images from the first session of the 70 impostors comprised in the development set. In this section the effect of estimating G using:

- fewer impostors to estimate G and
- a random initialisation of G

are both explored.

In Table 3 we show how the performance of the attack varies depending on the number of samples used to estimate this distribution G, for the best configuration of the attack $[N, M, \alpha] = [25, 5, 0.5]$. Because there are 70 impostors when the number of images which is equal to or smaller than 70 then one image per impostor is used. For larger numbers 2, 4 and 8 samples from each impostor are

Table 4. Results of the attack for different points of operation and the best configuration found of the attacking algorithm ($N = 25$, $M = 5$, $\alpha = 0.5$). The Success Rate is given in plain text (over a total of 200 accounts), and n_{comp} in **bold**. The average number of matchings needed for a successful brute-force attack (n_{bf}) is also given for reference.

	Operating points (in %)		
	FAR=0.1,FRR=25	FAR=0.05,FRR=30	FAR=0.01,FRR=50
Success Rate (in %)	99.0	98.5	86.0
n_{comp}	**840**	**1,068**	**4,492**
n_{bf}	1,000	2,000	10,000

used. In all cases, the resulting multivariate gaussian G results in $[-0.8 < \mu_i < 0.5]$ and $[0.2 < \sigma_i < 18]$, where μ_i and σ_i are respectively the mean and variance of the i-th dimension, with $i = 1 \ldots 91$. When using a random initialisation of G no real samples are used and the multivariate Gaussian is set to zero mean and unit variance.

From the results shown in Table 3 we can see that the number of samples used to compute the initial distribution G has little effect on the performance of the attack. In fact, the experiment proves that the algorithm can be successfully run starting from a general initial distribution G of zero mean and unit variance. This means that an attacker does not need to have any real face images to carry out the attack, this is in startk contrast to a brute force attack which requires a large database to perform a successful attack.

Analysis of different operating points. The attacks conducted at two additional operating points of $FAR = 0.05\%$ and $FAR = 0.1\%$. The evaluation was conducted using the optimal configuration $[N, M, \alpha] = [25, 5, 0.5]$ and with a general initial distribution G (with zero mean and unit variance). The operating point of $FAR = 0.05\%$ results in $FRR = 30\%$ and implies $n_{bf} = 2,000$ while the operating point of $FAR = 0.1\%$ results in $FRR = 25\%$ and implies $n_{bf} = 1,000$. The smaller values of the FAR imply a larger value for the threshold δ and this causes a rise in the average number of iterations required for a successful attack.

The results in Table 4 demonstrate that this technique is effective across mulitple operating points. In all cases it can be seen that the number of comparisons needed to break the system (using the Bayesian hill-climbing attack) is lower than that of a brute force attack. The Bayesian hill-climbing attack has the added advantage that it does not need any real face images to begin (initialise) the attack.

In Figs. 7 and 8 two examples of broken and non-broken accounts (corresponding to two of the users presented in Fig. 6) are shown. For each of the examples, the evolution of the score through the iterations of the algorithm is

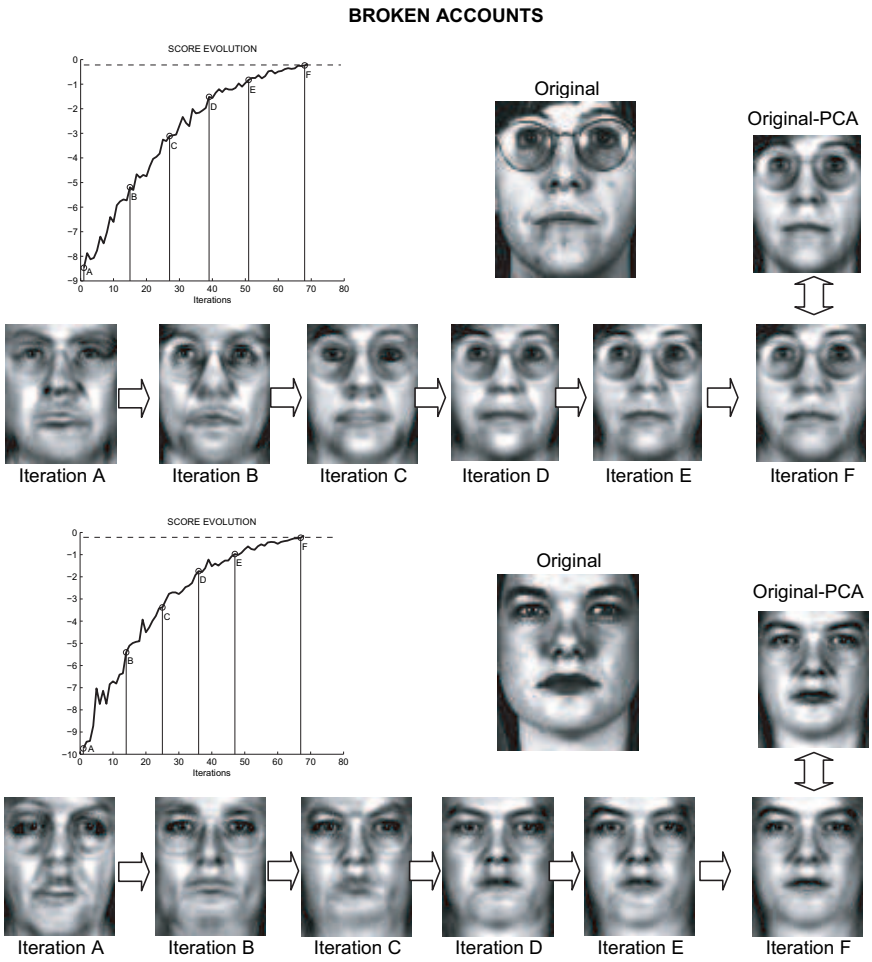

Fig. 7. Examples of the evolution of the score and the synthetic eigenfaces through the iterations of the attack for broken and accounts. The dashed line represents the objective threshold.

depicted, together with six points (including the first and the last one) of the iterative process (marked with letters A to F). The dashed line represents the objective value to be reached (i.e., the threshold δ). The two upper faces correspond to one of the original images of the attacked user and its representation in the PCA space (where part of the information has been lost). The sequence of the six faces below correspond to the feature vectors that produced each of the six scores marked with A to F, including the first one A which is produced by randomly sampling the estimated general distribution $\mathcal{G}$ and the last one F which is able to break the system.

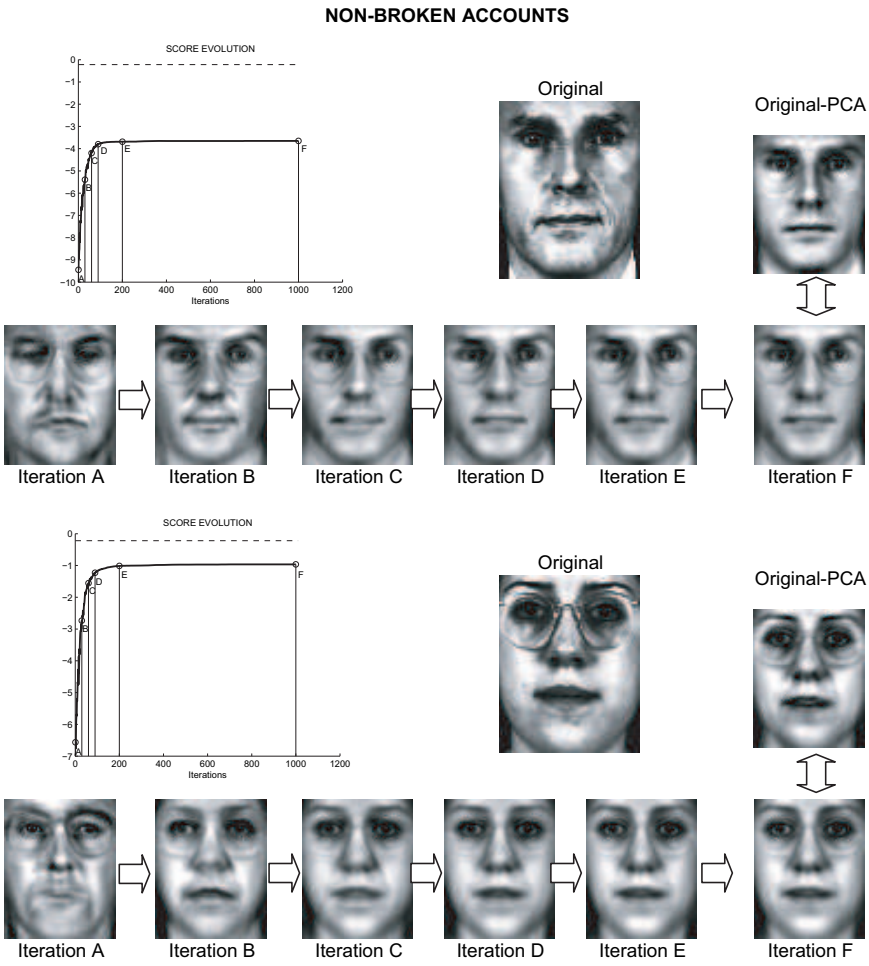

Fig. 8. Examples of the evolution of the score and the synthetic eigenfaces through the iterations of the attack for non-broken and accounts. The dashed line represents the objective threshold.

In Figs. 7 and 8 we can observe that the hill-climbing algorithm starts from a totally random face which is iteratively modified to make it resemble as much as possible to the PCA projection of the attacked users face (labeled as "Original-PCA"). In both cases (broken and non-broken accounts) the attack successfully finds a final image which is very similar to the objective face, however, in the case of the accounts resistant to the attack, the threshold is not reached as a consequence of the large user intra-variability, which leads to low scores even when compared with images of the same client.

Table 5. Success Rate (in %) of the hill-climbing attack when performing a single (top) and a multiple (bottom) block search, for increasing number of real samples used to compute the initial distribution G. The Success Rate (in %) appears in plain text, while the average number of iterations needed to break an account appears in **bold**.

	Number of real samples used to compute G						
	5	10	35	70	140	280	560
Sing. Block Search	100 **25**	100 **25**	100 **25**	100 **25**	100 **25**	100 **25**	100 **25**
Mult. Block Search	100 **1,031**	100 **1,025**	100 **1,631**	100 **1,514**	99.5 **1,328**	100 **1,293**	100 **1,254**

5.2 Case study 2: Attacking a GMM Face Verification System

In order to attack the GMM-based system, the best configuration of the algorithm found in the previous experiments was used. Using the optimal parameters ($N = 25$, $M = 5$, and $\alpha = 0.5$) from the previous case study meant that we could see if the attack configuration is highly dependent on the matcher tested, or if, on the contrary, a good set of parameter values can perform successfully on different systems. If it is not specified, the operating point selected to attack the system corresponds to FAR=0.01% (this means that a brute force attack would need on average to be successful $n_{bf} = 10,000$ matchings), and FRR=16%.

Two different approaches to the problem of attacking the GMM system are tested in these experiments:

- **Single block search**. This attack searches for one block to break the client's account. As explained in Sect. 3, the client score Sc is computed by taking the average score from all the blocks, therefore, if we are able to find one good matching block and replicate it for all the other blocks we should be able to produce a score high enough to be granted access. With these premises, this attack uses the Bayesian adaptation to search for one 15 dimensional vector which is repeated 285 times in order to produce the final synthetic template capable of breaking the system.
- **Multiple block search**. In this case we search for a unique set vectors which are capable of breaking into the client's account. Like the single block search this attack undertakes a search for a 15 dimensional vector, however, in this search 285 appropriate points which are all different (all unique) are searched for. This makes the multiple block search more difficult to accomplish and also more difficult to detect.

Experiments starting from an average initial distribution G. For these experiments we computed an initial distribution G representing the average block

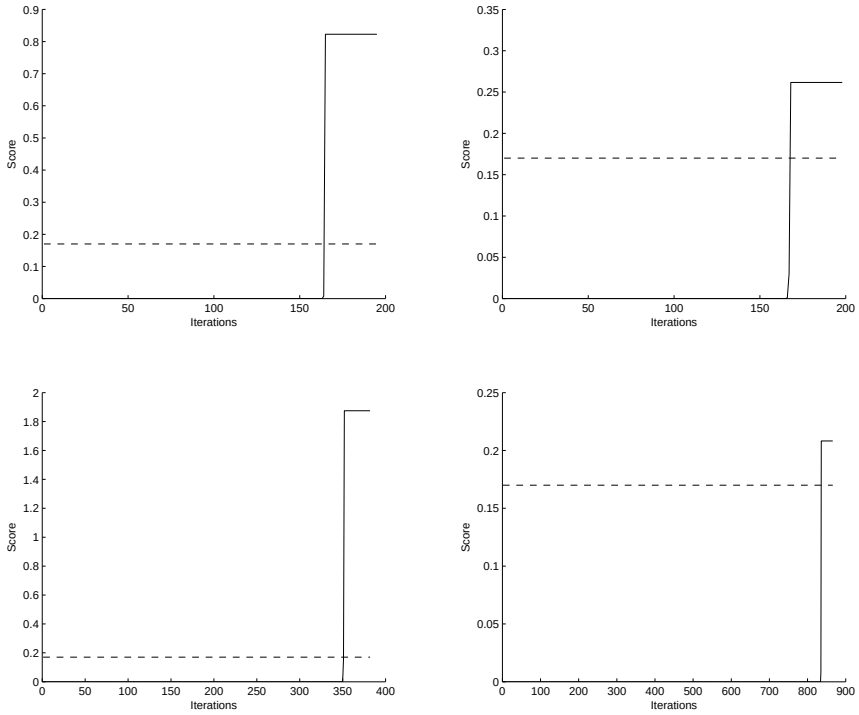

Fig. 9. Evolution of the score for four of the broken accounts using the single block search approach on the GMM-based face verification system. The dashed line represents the objective threshold.

(i.e., mean and average of the 15 dimensional vectors found in an image). The distribution was computed using a different number of images selected from the development set defined in LP2, the protocol used for previous case study 5.1 is reproduced here; for numbers of images smaller than 70, one sample per impostor (randomly selected) is picked, while for larger numbers 140, 280, and 560 then 2, 4, and 8 samples per impostor are selected respectively. In Table 5 the results for the single and multiple block search approaches are shown.

For the single block search all the accounts are broken at the first iteration of the attack. This means that the Bayesian adaptation hill-climbing algorithm is not necessary and that the system can be broken using synthetic templates built replicating 285 times a random average block computed using as few as 5 images; since each iteration consists of 25 comparisons. This is a serious security flaw, however, it can easily be countermeasured by checking if all the blocks in the template trying to access the system are different.

The multiple block search attack has at least a 99.5% success rate. This success rate is regardless of the number of images used to compute the initial

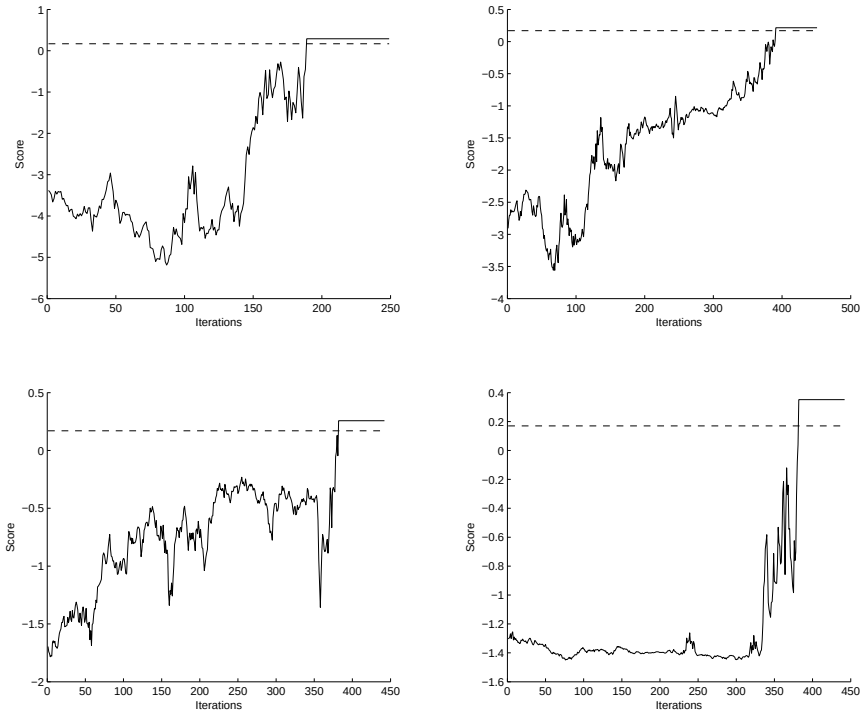

Fig. 10. Evolution of the score for four of the broken accounts using the multiple block search approach on the GMM-based face verification system. The dashed line represents the objective threshold.

distribution G. However, for this attack there are, on average, around 1,200 comparisons (corresponding to around 55 iterations of the attack) to break the system. This represents less than one sixth of the matchings required by a successful brute force attack ($n_{bf}10,000$) with the added advantage that just 5 real face images are needed to perform the hill-climbing attack. Although the multiple block search is slower than the single block search approach, in this case countermeasuring the attack is significantly more difficult as all the vectors, which form the synthetic template, are different amongst themselves.

Experiments starting from a random initial distribution G. The GMM-based system was also attacked starting from a random initial distribution G with zero mean and unit variance. For the single block search approach 98% of the accounts (out of the total 200 tested) were bypassed, and the average number of matchings needed to enter the system was 1,102. Although that success rate is very high, we can observe in Fig. 9 that the hill-climbing is not working properly as the score remains unaltered and equal to zero throughout the iterations (there

is no increasing or *hill-climbing* effect) until at one point it very rapidly (two or three iterations) reaches the objective value (shown with a dashed line).

This behaviour can be explained by the fact that the score given by the system is the substraction of the client and the world scores (see Sect. 3). As the synthetic templates are built duplicating a block randomly selected from a general distribution G, their appearance is completely different to that of a face and so both similarity scores (those obtained from the world and client model) are the same, leading to a zero final score. As the final score obtained by all the synthetic templates is the same (zero), we have no feedback as about the local distribution L (representing those templates which are more similar to the attacked one). Therefore, the algorithm ends up doing a random search until at some point one of the templates produces (by chance) a non-zero score.

Even though this attack is the equivalent of a random search it successfully breaks the system at the first attempt (corresponding to 25 matchings) for 43% of the tested accounts. Therefore, even this security breach should be taken into account when designing countermeasures (e.g., checking that all the blocks of the template are different) for final applications.

The above experiments were repeated using the multiple block search scheme. In this case, all 200 accounts were bypassed and the average number of comparisons needed to break the system was 3,016. In Fig. 10 it can be observed that the hill-climbing algorithm is able to produce the desired increasing effect in the score throughout the iterations. We can see that the synthetic templates produce a negative final score (they get a better matching score from the world model than from the client model, $S = Sc - Sw$) and thus, the algorithm gets the necessary feedback to iteratively improve the estimate of the vector distribution G. Again, this approach is slower than the single block search, but on the other hand it is more difficult to countermeasure as all the image blocks are different amongst themselves.

Analysis of different operating points. The GMM-based system was evaluated at two additional operating points, these being:

- FAR=0.05%, FRR=7% (which implies $n_{bf} = 2,000$), and
- FAR=0.1%, FRR=5% (which implies $n_{bf} = 1,000$).

For these experiments the initial distribution G was chosen as a Gaussian distribution with zero mean and unit variance and the two different attacking approaches (single block search and multiple block search) were tested.

The results indicate that the Bayesian hill-climbing attack is effective for all of the operating points. It is considered to be effective because for all of the operating points the number of comparisons needed to break the system is always lower than that of a brute force attack. Also, the number of broken accounts remains unaltered.

Table 6. Results of the attack for different points of operation and the best configuration found of the attacking algorithm (N=25, M=5, $\alpha = 0.5$). The Success Rate is given in plain text (over a total 200), and n_{comp} in (**bold**). The average number of matchings needed for a successful brute-force attack (n_{bf}) is also given for reference.

	Operating points (in %)		
	FAR=0.1,FRR=5	FAR=0.05,FRR=7	FAR=0.01,FRR=16
Sing. Block Search	100 **123**	100 **413**	98 **1,102**
Mult. Block Search	100 **724**	100 **1,835**	100 **3,016**
n_{bf}	1,000	2,000	10,000

6 Conclusions

The effectiveness of the Bayesian hill-climbing attack to break two different face verification systems was examined. Experimental results show that the two face verification systems studied are highly vulnerable to this type of attack, with over an 85% success rate for all of the attacks; even when no real images were used to initialize the algorithm.

The performance of the Bayesian hill-climbing algorithm was compared to a brute force attack. It was found that the Bayesian hill-climbing attack is more efficient under all tested conditions. In addition, it is worth noting that the resources required by both approaches differ greatly. In order to perform an efficient brute-force attack, the attacker must have a database of more than a thousand real different templates, while the hill-climbing approach does not need any real templates to be successful.

It has also been found that the GMM Parts-based system is very vulnerable to random attacks carried out with templates formed by a replicated random or average block. This important security flaw can be solved by incorporating to the system a mechanism to detect duplicated patterns in the image.

Finally, it has been proven that the Bayesian hill-climbing algorithm can be successfully applied not only to different matchers but also to different biometric traits. In [12] it was shown to be an effective method to attack an on-line signature verification system and for these experiments it has been shown to be very effective at breaking two different face verification systems. Therefore, this threat should be taken into account when designing any biometric security system working with fixed length feature vectors and the necessary countermeasures against the attack should be introduced.

7 Acknowledgements

J. G. is supported by a FPU Fellowship from Spanish MEC and J. F. is supported by a Marie Curie Fellowship from the European Commission. This work was supported by Spanish MEC under project TEC2006-13141-C03-03 and the European NoE Biosecure. J. G. would also like to thank the IDIAP Research Institute for hosting him during the development of the present research work.

References

1. Jain, A.K., Ross, A., Pankanti, S.: Biometrics: a tool for information security. IEEE Trans. on Information Forensics and Security **1**(2) (2006) 125–143
2. Jain, A.K., Flynn, P., Ross, A., eds.: Handbook of biometrics. Springer (2008)
3. Wayman, J., Jain, A., et al.: Biometric systems. Technology, design and performance evaluation. Springer (2005)
4. Ratha, N., Connell, J., Bolle, R.: An analysis of minutiae matching strength. Proc. Audio- and Video-Based Biometric Person Authentication (AVBPA) (2001) 223–228
5. van der Putte, T., Keuning, J.: Biometrical fingerprint recognition: don't get your fingers burned. In: Proc. IFIP Conference on Smart Card Research and Advanced Applications (CARDIS). (2000) 289–303
6. Galbally, J., Fierrez, J., et al.: On the vulnerability of fingerprint verification systems to fake fingerprint attacks. In: Proc. IEEE of International Carnahan Conference on Security Technology (ICCST). (2006) 130–136
7. Pacut, A., Czajka, A.: Aliveness detection for iris biometrics. In: Proc. IEEE of International Carnahan Conference on Security Technology (ICCST). (October 2006) 122–129
8. Soutar, C.: Biometric system security. http://www.bioscrypt.com/assets/security_soutar.pdf
9. Adler, A.: Sample images can be independently restored from face recognition templates. In: Proc. Canadian Conference Electrical and Computing Engineering (CCECE). Volume 2. (2003) 1163–1166
10. Uludag, U., Jain, A.K.: Attacks on biometric systems: a case study in fingerprints. In: Proc. SPIE-IE. Volume 5306. (2004) 622–633
11. Martinez-Diaz, M., Fierrez, J., et al.: Hill-climbing and brute force attacks on biometric systems: a case study in match-on-card fingerprint verification. In: Proc. IEEE of International Carnahan Conference on Security Technology (ICCST). (2006) 151–159
12. Galbally, J., Fierrez, J., Ortega-Garcia, J.: Bayesian hill-climbing attack and its application to signature verification. In: Proc. IAPR International Conference on Biometrics (ICB), Springer LNCS-4642 (2007) 386–395
13. Messer, K., Matas, J., et al.: XM2VTSDB: The extended M2VTS database. In: Proc. IAPR Audio- and Video-Based Biometric Person Authentication (AVBPA). (1999)
14. Turk, M.A., Pentland, A.P.: Face recognition using eigenfaces. In: Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (1991) 586–591
15. Cardinaux, F., Sanderson, C., Marcel, S.: Comparison of MLP and GMM classifiers for face verification on xm2vts. In: Proc. IAPR International Conference on Audio- and Video-based Biometric Person Authentication (AVBPA). (2003)

16. Phillips, J., Flynn, P., et al.: Overview of the face recognition grand challenge. In: Proc. ICCVPR. (2005)
17. : ANSI X9.84-2001, Biometric Information Management and Security.