Attacks on Biometric Systems: A Case Study in Fingerprints

Umut Uludag*, Anil K. Jain*
Department of Computer Science and Engineering,
Michigan State University, East Lansing, MI, USA 48824

ABSTRACT

In spite of numerous advantages of biometrics-based personal authentication systems over traditional security systems based on token or knowledge, they are vulnerable to attacks that can decrease their security considerably. In this paper, we analyze these attacks in the realm of a fingerprint biometric system. We propose an attack system that uses a hill climbing procedure to synthesize the target minutia templates and evaluate its feasibility with extensive experimental results conducted on a large fingerprint database. Several measures that can be utilized to decrease the probability of such attacks and their ramifications are also presented.

Keywords: Biometrics, fingerprint, minutiae, security, template, attack, synthesis

1. INTRODUCTION

Biometrics-based personal authentication systems that use physiological (e.g., fingerprint, face) or behavioral (e.g., speech, handwriting) traits are becoming increasingly popular, compared to traditional systems that are based on tokens (e.g., key) or knowledge (e.g., password) [1]. Traditional authentication systems cannot discriminate between an impostor who fraudulently obtains the access privileges (e.g., key, password) of a genuine user and the genuine user herself. Furthermore, biometric authentication systems can be more convenient for the users since there is no password to be forgotten or key to be lost and a single biometric trait (e.g., fingerprint) can be used to access several accounts without the burden of remembering passwords.

In spite their numerous advantages, biometric systems are vulnerable to attacks, which can decrease their security. Ratha et al. [2] analyzed these attacks, and grouped them into eight classes. Fig. 1 shows these attacks along with the components of a typical biometric system that can be compromised. Type 1 attack involves presenting a fake biometric (e.g., synthetic fingerprint, face, iris) to the sensor. Submitting a previously intercepted biometric data constitutes the second type of attack (replay). In the third type of attack, the feature extractor module is compromised to produce feature values selected by the attacker. Genuine feature values are replaced with the ones selected by the attacker in the fourth type of attack. Matcher can be modified to output an artificially high matching score in the fifth type of attack. The attack on the template database (e.g., adding a new template, modifying an existing template, removing templates, etc.) constitutes the sixth type of attack. The transmission medium between the template database and matcher is attacked in the seventh type of attack, resulting in the alteration of the transmitted templates. Finally, the matcher result (accept or reject) can be overridden by the attacker.

Schneier [3] compares traditional security systems with biometric systems. The lack of secrecy (e.g., leaving fingerprint impressions on the surfaces we touch), and non-replaceability (e.g., once the biometric data is compromised, there is no way to return to a secure situation, unlike replacing a key or password) are identified as the main problems of biometric systems.

Maltoni et al. [4] describe typical threats for a generic authentication application, which may result in quite different effects for traditional and biometrics-based systems. In *Denial of Service (DoS)*, an attacker corrupts the authentication system so that legitimate users cannot use it. For a biometric authentication system, an online authentication server that processes access requests (via retrieving templates from a database and performing matching with the transferred

*{uludagum, jain}@cse.msu.edu; phone: 1 517 355-9319; fax: 1 517 432-1061; http://biometrics.cse.msu.edu

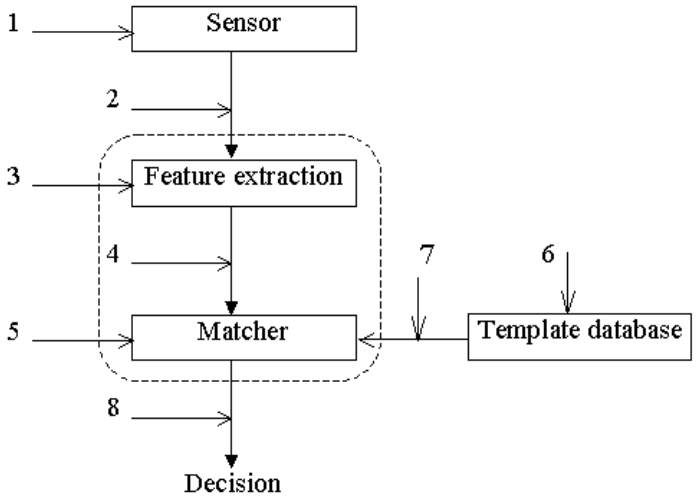


Fig. 1. Eight different attack points in a biometric authentication system (adapted from [2]).

biometric data) can be bombarded with many bogus access requests, to a point where the server's computational resources cannot handle valid requests any more. In *circumvention*, an attacker gains access to the system protected by the authentication application. This threat can be cast as a privacy attack, where the attacker accesses the data that she was not authorized (e.g., accessing the medical records of another user) or, as a subversive attack, where the attacker manipulates the system (e.g., changing those records, submitting bogus insurance claims, etc.). In *repudiation,* the attacker denies accessing the system. For example, a corrupt bank clerk who modifies some financial records illegally may claim that her biometric data was "stolen", or she can argue that the False Accept Rate (FAR) phenomenon associated with any biometric may have been the cause of the problem.

In *contamination (covert acquisition),* an attacker can surreptitiously obtain biometric data of legitimate users (e.g., lifting a latent fingerprint and constructing a three-dimensional mold) and use it to access the system. Further, the biometric data associated with a specific application can be used in another unintended application (e.g., using a fingerprint for accessing medical records instead of the intended use of office door access control). This becomes especially important for biometric systems since we have a limited number of useful biometric traits, compared to practically unlimited number of traditional access identities (e.g., keys and passwords). Cross-application usage of biometric data becomes more probable with the growing number of applications using biometrics (e.g., opening car or office doors, accessing bank accounts, accessing medical records, locking computer screens, gaining travel authorization, etc.). In *collusion,* a legitimate user with wide access privileges (e.g., system administrator) is the attacker who illegally modifies the system. In *coercion,* attackers force the legitimate users to access the system (e.g., using a fingerprint to access ATM accounts at a gunpoint) [4].

The problems that may arise from the above mentioned attacks on biometric systems are raising concerns as more and more biometric systems are being deployed both commercially and in government applications [5]. This, along with the increase in the size of the population using these systems and the expanding application areas (visa, border control, health care, welfare distribution, e-commerce, etc.) may lead to possible finance, privacy, and security related breaches.

In this paper, we analyze these attacks in the realm of a fingerprint-based biometric system. Fingerprint-based systems are among the most frequently deployed biometric systems, due to their accuracy, size, cost, performance and proven track record. Hence, we choose to use a fingerprint-based system in this study. In Section 2, previous related studies are summarized. In Section 3, we propose a system that can attack a minutia-based fingerprint matcher. Section 4 contains the experimental results pertaining to the feasibility of such attacks. Section 5 summarizes several measures that can be used to counter such attacks. Conclusions and future research directions are given in Section 6.

2. PREVIOUS WORK

In this section, we summarize several studies that show the vulnerability of biometric systems and provide solutions to some of the attacks presented in Section 1.

Fake biometric submission to the sensor (type 1 attack) is shown to be quite successful by several researchers. Note that this attack does not need anything more than a fake biometric; hence the feasibility of it compared to the other attacks can be high. For example, neither a knowledge of the matcher or template specifications nor template database access privileges (generally limited to system administrators) are necessary. Also, since it operates in the *analog* domain, outside the digital limits of the biometric system, the digital protection mechanisms such as encryption, digital signature, hashing etc. are not applicable.

Putte and Keuning [6] tested several fingerprint sensors to check whether they accept an artificially created (dummy) finger instead of a real finger. The authors describe methods to create dummy fingers with and without the cooperation of the real owner of the biometric (say, Alice). When the owner cooperates (namely, Alice is helping the attackers), obviously, the quality of the produced dummy fingers can be higher than those produced without cooperation (namely, Alice is a victim of the attackers). In the former case, after creating the plaster cast of the finger, liquid silicon rubber is filled inside the cast to create a wafer-thin dummy that can be attached to a finger, without being noticed at all. This operation is said to take only a few hours. In the latter case, more time (nearly eight hours) and more skill are needed: first, a fine powder is used to enhance the latent fingerprints left on a glass or scanner surface. Then, a photo of the print is taken which is used to transfer the print to a PCB (Printed Circuit Board). UV light exposure and acid etching leaves the profile of the print on the board, which is used for producing the silicon cement dummy. In both the cases, the authors used cheap and easily accessible material for the creation of the dummy finger. Five out of six sensors (that included both optical and solid state sensors) tested by the authors accepted a dummy finger created by the above methods as a real finger in the first attempt; the remaining sensor accepted the dummy finger in the second attempt. The authors argue that the properties (e.g., temperature, conductivity, heartbeat, dielectric constant, etc.) claimed to be used by the scanner manufacturers to distinguish a dummy finger from a real finger, may not perform well since the detection margins of the system need to be adjusted to operate in different environments (e.g., indoor vs. outdoor), different environmental conditions (e.g., hot summer vs. cold winter), etc. Wafer thin silicon dummy fingers may lead to changes that are still within the detection margins of the systems.

Matsumoto et al. [7] attacked 11 different fingerprint verification systems with artificially created gummy (gelatin) fingers. For a cooperative owner, her finger is pressed to a plastic mold, and gelatine leaf is used to create the gummy finger. The operation is said to take lass than an hour. It was found that the gummy fingers could be enrolled in all of the 11 systems, and they were accepted with a probability of 68-100%. When the owner does not cooperate, a residual fingerprint from a glass plate is enhanced with a cyanoacrylate adhesive. After capturing an image of the print, PCB based processing similar to the operation described above is used to create the gummy fingers. All of the 11 systems enrolled the gummy fingers and they accepted the gummy fingers with more than 67% probability.

To overcome such fake biometric attacks, Derakhshani et al. [8] proposed two software-based methods (not based on sensors that measure temperature, conductivity, etc.) for fingerprint liveness detection. They used a commercially available capacitive sensor and the sole input to the liveness detection module is a 5-second video of the fingerprints. In their static method, the periodicity of sweat pores along the ridges is used for liveness detection. In the dynamic method, sweat diffusion pattern over time along the ridges is measured. Live fingers, fingers from cadavers, and dummy fingers made up of play dough are used in the experiments. A back propagation neural network (BPNN) based classifier is used to distinguish live fingers from cadaver/dummy fingers. The static method leads to an EER of nearly 10%; the dynamic method leads to an EER in the range of 11-39%, where a false accept event is a cadaver/dummy finger being classified as live, and a false reject event is a live finger being classified as a cadaver/dummy.

We can see that fake biometric attacks can be quite successful in fooling the existing systems, and no perfect (either hardware or software) solution is currently available. As noted previously, this attack aims at a point in the biometric system that is very *close* to the end user (in the sense that a physical replica is used) and this may hinder the utilization

of some protection mechanisms. One other problem associated with this attack is that the means to *detect* an attack are limited.

The remaining attacks are feasible only if some knowledge about the biometric authentication system and/or some access privileges are available to the attacker. This fact may decrease their applicability compared to type 1 attacks. On the other hand, it may also increase their applicability since no physical production (that is still more costly and time consuming compared to *digital* production) such as plastic molding, is necessary. Further, in the digital domain, the attacks can be executed in relatively less time.

For eliminating type 2 attacks, where a previously intercepted biometric is replayed, Ratha et al. [9] proposed a challenge/response based system. A pseudo-random challenge is presented to the sensor by a secure transaction server. At that time, the sensor acquires the current biometric signal and computes the response corresponding to the challenge (for example, pixel values at locations indicated in the challenge). The acquired signal and the corresponding response are sent to the transaction server where the response is checked against the received signal for consistency. An inconsistency reveals the possibility of the resubmission attack.

Soutar [10] proposed a "hill-climbing" attack for a simple image recognition system based on filter-based correlation. Synthetic templates are gradually input to a biometric authentication system; using the scores returned by the matching system, Soutar showed the system could be compromised till the point of incorrect positive identification. Outputting only the quantized matching scores, not absolute scores, is proposed as a way to increase the time needed for an incorrect positive identification, thereby decreasing the practicality of this attack. This hill climbing attack can be cast as either type 2 or type 4 attack. As an example of the former, Adler [11] proposed an attack on a face recognition system where the account of a specific user enrolled in the system is attacked via synthetically generated face images. An initial face image is selected. Using the matching scores returned from the matcher that were generated for each of the successive face images, this initial image is modified. At each step, several eigen-images (that can be generated from public domain face databases) are multiplied with a weight and added to the current candidate face image. The modified image that leads to the highest matching score is input as the new candidate image. These iterations are repeated until no improvement in matching score is observed. Experimental results on three commercial face recognition systems show that after about 4000 iterations, a sufficiently large matching score is obtained, which corresponds to a very high (~99.9%) confidence of matching scores. The author calculated the confidence as a sigmoidal function of the matching scores.

When hill climbing is applied as a type 2 attack (before the feature extractor), the information about the template format (which is essential for a type 4 attack) is not necessary. Synthetic images are input to the matching algorithm, which in turn handles conversion of the images into any suitable representation before matching. But, for a fingerprint-based biometric system, such an approach presents challenges not found in a face-based system: the discriminating information in fingerprints is not tied to specific geometrical relationships, as it is in face-based systems (e.g., between eyes, nose, mouth, etc.) and methods that are inherently linked to the correct registration of image pixels (e.g., eigenimage analysis used in [11]) seem unsuitable.

A study that is related to the template database security (type 6 attack) is given in [12]. Using a commercial fingerprint matcher, the minutiae template data is reverse engineered by the author and the corresponding synthetic fingerprint images are generated. Although the generated images are not very realistic and few experimental results are provided, the possibility of this *masquerading* may imply that raw biometric templates need to be secured, using, for example, techniques such as encryption. Another method to protect templates from fraudulent usage involves using a distorted (but noninvertible) version of the biometric signal or the feature vector [9]; if a specific representation of template is compromised, the distortion transform can be replaced with another one from a transform database. Every application can use a different transform (e.g., health care, visa, e-commerce) so that the privacy concerns of subjects related to database sharing between institutions can be addressed. Data hiding and watermarking techniques have also been proposed as means of increasing the security of fingerprint images, by detecting modifications [13], by hiding one biometric into another [14] and by hiding messages (authentication stamps such as personal ID information) in the compressed domain [9]. Linnartz and Tuyls [15] proposed delta-contracting and epsilon-revealing functions as preprocessors to construct helper data that is used in a way that no information about user templates is released to unauthorized parties.

In the following section, we propose a system that uses hill climbing as a type 4 attack that bypasses the feature extractor and uses synthetically generated feature sets in the realm of a minutiae-based fingerprint matcher. Note that the format of the feature template used by the system should be known in advance to launch such an attack.

3. SYSTEM ARCHITECTURE

We have designed an attack system for a minutiae-based fingerprint authentication system. While there exist other representation methods for fingerprints (e.g., FingerCode [16]), we chose a minutiae-based system as our test bed since they are used in most of the commercial fingerprint authentication systems.

In typical minutiae-based fingerprint authentication systems, minutiae points consist of ridge endings and ridge bifurcations. Generally, all of the minutiae based systems use the location (c,r) of the minutiae and the orientation θ associated with the minutiae as the attributes; but some systems use additional information such as ridge flow around the minutiae [17]. For keeping our attack system more general, we simply use (c,r,θ) attributes for each minutia. This is also consistent with the proposed minutiae template exchange format [18] that excludes proprietary features, and encompasses only the location, orientation and type for each minutia.

Our attack system inputs synthetic minutiae sets to the matcher with the aim of gaining access to the system in place of a genuine user. Note that the user's template information is unknown to the attack system. Using the scores returned by the matcher and the characteristics of these minutiae sets, the attack system tries to generate a minutia set that results in a sufficiently high matching score to achieve positive identification. The block diagram of the proposed system is given in Fig. 2.

Our notation in the remainder of the paper will be as follows:

- D_i : The database template corresponding to user i, $i = 1,2,3,....N$, where N is the total number of users registered in the system. It is assumed that the attacking system knows the format of this template, but it cannot access the template itself.
- n_i : The total number of minutiae in D_i. Note that the attacking system does not know this value.
- T_i^j : The j^{th} synthetic template generated by the attacking system for user i. This template has the same format as database templates; it can be represented as

$$T_i^j = \begin{bmatrix} {}^1c_i^j & {}^1r_i^j & {}^1\theta_i^j \\ {}^2c_i^j & {}^2r_i^j & {}^2\theta_i^j \\ \vdots & \vdots & \vdots \\ {}^{n_{ij}}c_i^j & {}^{n_{ij}}r_i^j & {}^{n_{ij}}\theta_i^j \end{bmatrix}, \tag{1}$$

where each row represents column index, row index and orientation associated with a minutia; upper left hand subscript denotes the minutiae index, so the total number of minutiae in T_i^j is n_{ij}.

- $S(D_i, T_i^j)$: The matching score between D_i and T_i^j.
- $S_{threshold}$: The decision threshold used by the matcher. Note that the attacking system does not know this value.

For attacking a specific user's (i) account, the attacking system follows the following five steps:

- Step 1 (Initial guessing): Generate a fixed number of synthetic templates. In the current implementation, 100 random minutia templates ($T_i^1, T_i^2, T_i^3, \ldots, T_i^{100}$) are created.

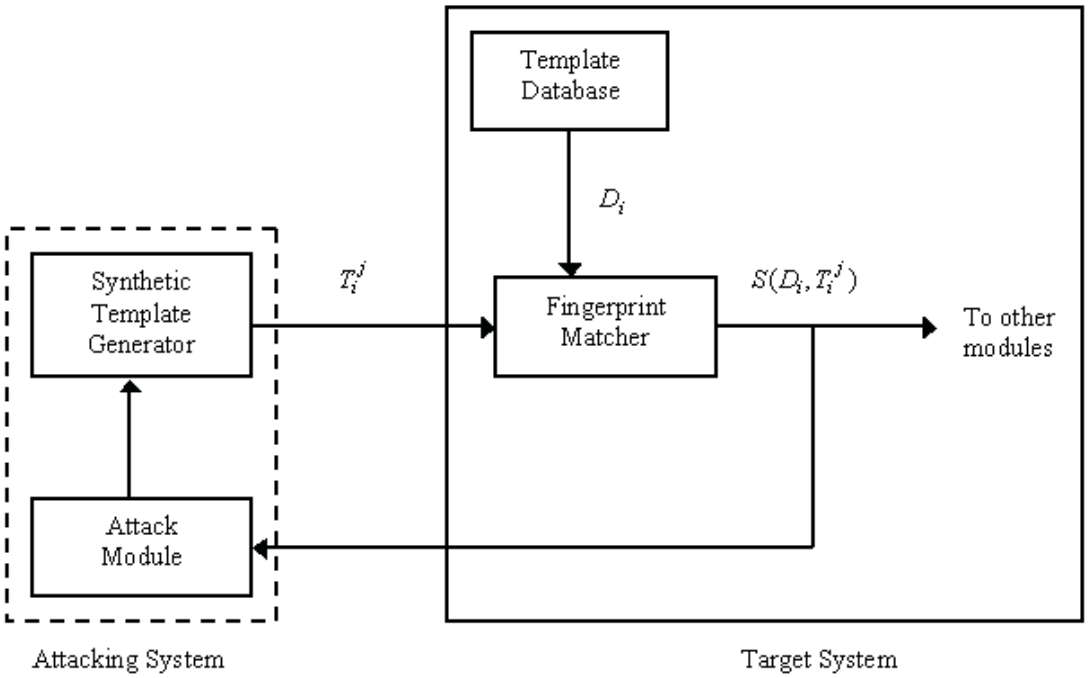


Fig. 2. Overview of the attack system.

- Step 2 (Try initial guesses): Attack user i account with the templates generated in Step 1; accumulate the corresponding matching scores ($S(D_i, T_i^1), S(D_i, T_i^2), S(D_i, T_i^3), ..., S(D_i, T_i^{100})$).
- Step 3 (Pick the best initial guess): Declare the best guess (T_i^{best}) to be the template resulting in the highest matching score. Declare the best score ($S^{best}(D_i)$) to be the highest matching score.
- Step 4 (Try modification set): Modify T_i^{best} by (i) perturbing an existing minutia, (ii) adding a new minutia, (iii) replacing an existing minutia, and (iv) deleting an existing minutia. If for any one of these attempts, the matching score is larger than $S^{best}(D_i)$, declare the modified template as T_i^{best}, and update $S^{best}(D_i)$ accordingly. Else, do not change the parameters of T_i^{best}.
- Step 5 (Obtaining result): If the current best score is accepted by the matcher (namely, $S^{best}(D_i) > S_{Threshold}$), stop the attack; else, go to Step 4.

We assume that the resolution (in dpi) and size (in pixels) of the images generating the original templates is known to the attacking system. This is a valid assumption since these values are generally announced by sensor manufacturers. For the current implementation, we used MSU-VERIDICOM fingerprint image database, with 500 dpi, 300x300 images. The image size is used for generating the location of synthetic minutiae; the resolution determines the inter-ridge distance (9 pixels for 500 dpi) associated with the fingerprints.

For eliminating the generation of minutiae too close to each other, we first create a rectangular grid (where each cell size is set to be the same as the inter-ridge distance). Then, the 2D location of a minutia, (c, r), is created to be in the center of those cells, where the cells to be occupied are selected randomly. Hence, the attacking program does not create minutiae that are closer than the inter-ridge distance. This helps is creating dispersed minutiae sets.

The angle value associated with a minutia is generated randomly as a quantized value in the range [0, 360). For the current implementation, we quantized this interval into 16 equally spaced intervals.

In Steps 1-3, the aim is to find a good initial guess and concentrate on modifying it. Note that if the initial guess is bad, the algorithm may need more iterations to break into an account. Initial templates all have the same number of minutiae ($n_{ij} = 25,\ \forall i,\ j = 1,2,...,100$). Even though the actual number of minutiae in the target template (n_i) is not known, this value of 25 is selected to be a typical number. Further, the algorithm modifies the template set so that the number of minutia can increase or decrease, based on the returned matching scores, as explained below.

Step 4 is composed of 4 iterations in each loop of the process. At the first iteration, an existing minutia is randomly picked, and its location or angle are modified slightly, both with a probability of 0.5. Here, the aim is to change either the location or angle of a minutia and see the effect on the matching score. The location in each direction is perturbed with a distance equal to inter-ridge distance (note that a minutia has at most 8 neighboring cells to go to); the angle is perturbed so that it is increased or decreased to the next angle quantum (hence, it changes ±22.5 degrees in the current implementation). At the second iteration, a new randomly created minutia is added to the current template. At the third iteration, an existing minutia is randomly picked and it is replaced with a randomly created minutia. At the fourth iteration, an existing minutia is randomly picked and it is deleted from the current template.

After each iteration, if the matching score improves, we replace the current template with the new template; otherwise we do not change it. Hence, the algorithm "hill climbs" to increase the matching score.

4. EXPERIMENTAL RESULTS

The minutiae-based fingerprint matcher used in this study is based on the system described in [17]. Jain et al. [17] use the ridge information, in addition to (c, r, θ) triplets associated with each minutia. We eliminated the ridge information in the matcher used in this study.

We used the right-index finger segment of MSU-VERIDICOM fingerprint image database (160 users, 4 impressions/finger, obtained with a VERIDICOM solid state sensor, 500 dpi 300x300 images) in our experiments. The minutiae for each of these 640 images are extracted [17] and the matcher is run on these minutia feature sets. A total of 1,600 genuine and 203,520 imposter scores are obtained (scores are in the range [0, 100]). Fig. 3 shows the associated ROC (Receiver Operating Characteristics) curve. Fig. 4 shows the plots of FAR (False Accept Rate) and FRR (False Reject Rate) vs. the decision threshold.

As an example operating point for the fingerprint verification system, we consider FAR = 0.1%, for which GAR=87.6% and the decision threshold is found to be 12.22. We chose the 0.1% FAR value since it is a typical value used by system administrators. Note that this specific decision threshold is not known to the attacking program: it just submits synthetic feature sets to the matcher and obtains the corresponding scores. Obviously, if the attacking program can break into a specific account, it gets this information and stops the attack; otherwise it continues. In fact, if the underlying verification system uses the same decision threshold for every account (and this threshold does not change over time), after the attacking program breaks the first account, it will have a partial knowledge of the decision threshold. However, for the following experiments, we did not make use of this fact, to keep the system more general.

Selecting this specific threshold (FAR=0.1%) implies that, on the average, 1 in 1000 imposter attempts will be accepted as a genuine match. Below, we show that our attack algorithm can break into accounts (namely, generate a synthetic minutiae template set T_i^* for which $S(D_i, T_i^*) > S_{threshold}$) by trying a fewer number of attempts.

After setting $S_{threshold}$ to 12.22, we attacked all of the 160 user accounts in the system (the first impression associated with every finger is targeted) using the attack methodology explained in Section 3. The attacking program broke all of the 160 accounts with less than 1,000 attempts for each account. The minimum, mean, and the maximum number of attack attempts that is required for breaking into all accounts are found to be 132, 271, and 871, respectively. Fig. 5 shows the histogram of the number of required attempts at which the account is broken. Since, on the average, the attacking program needed only 271 iterations for breaking into an account, we can conclude that hill climbing procedure as explained previously performs very effectively.

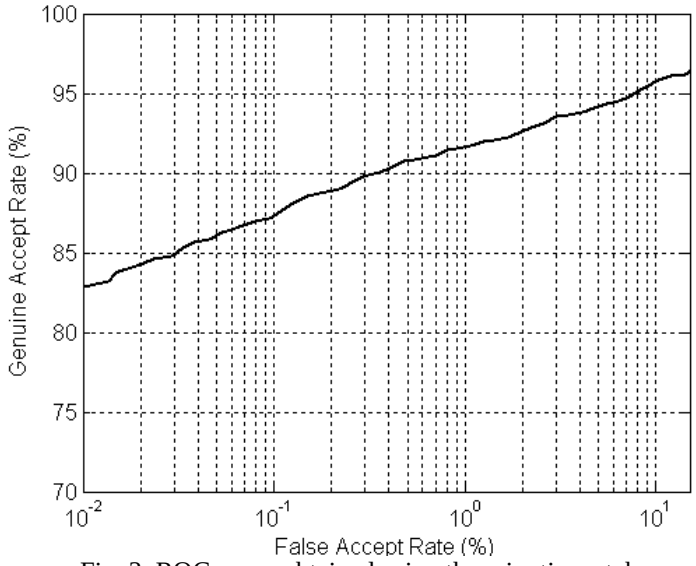


Fig. 3. ROC curve obtained using the minutia matcher.

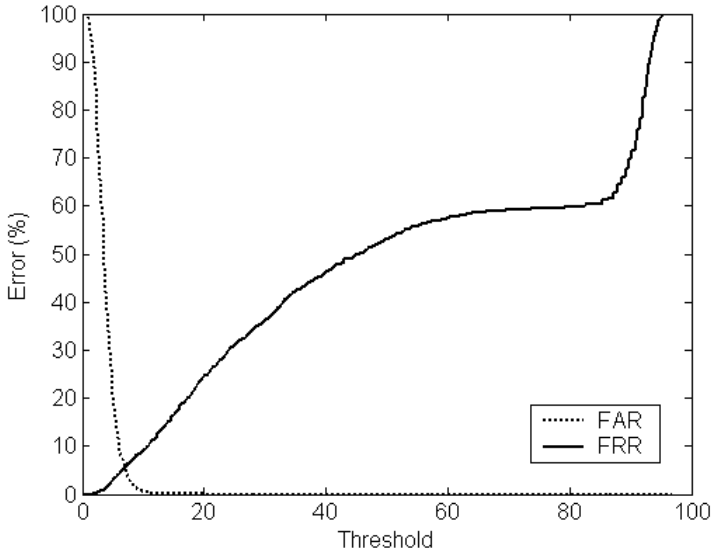


Fig. 4. FAR and FRR vs. decision threshold plots.

For analyzing the progression of matching scores for specific accounts, we selected three accounts that were broken at the 132nd, 271st, and 871st attempts (for simulating an “easy” account, a “medium” account, and a “hard” account, respectively, in terms of the number of access attempts necessary to break them). The matching scores for the final synthetic templates that broke these accounts were 12.5, 16.4, and 13.3, respectively. Fig. 6 shows the progression of matching scores for these accounts. Fig. 7 shows the original fingerprint image (with overlaid) minutiae and the minutiae of the original and synthetic templates, for these three accounts.

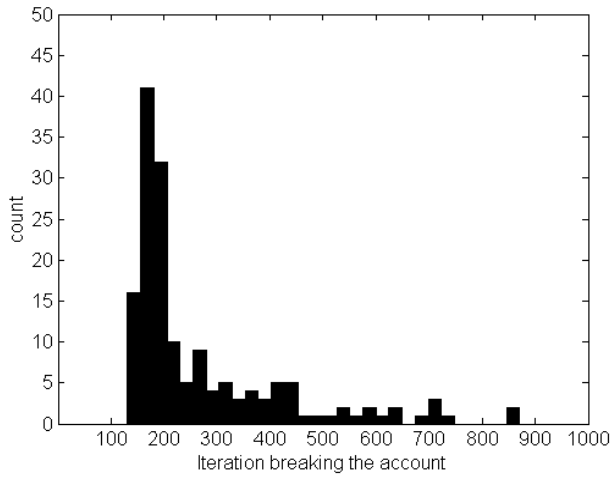
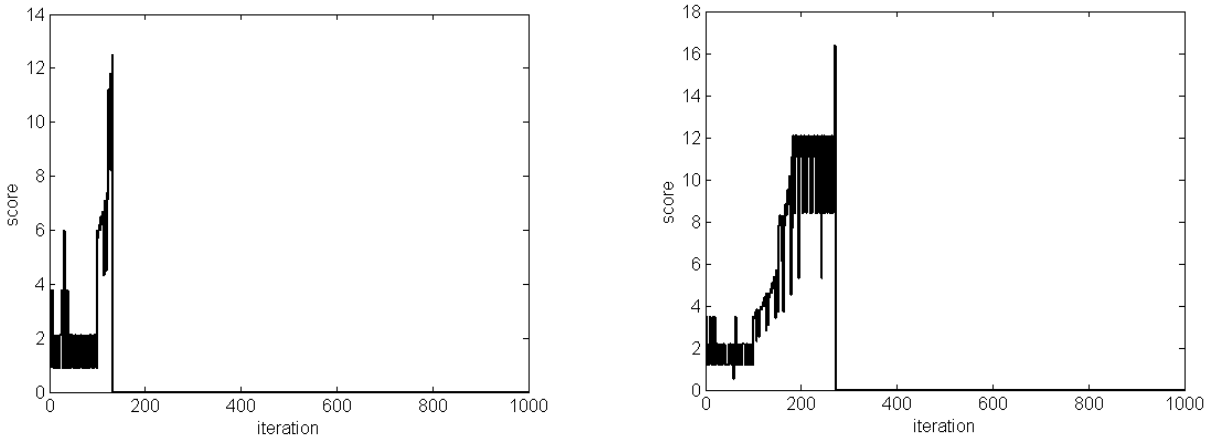
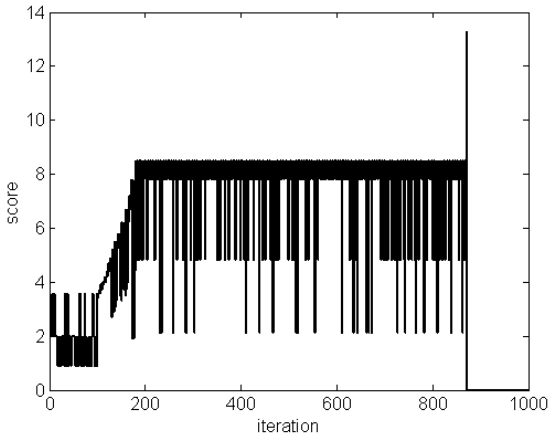


Fig. 5. Histogram of number of attempts at which the accounts are broken.

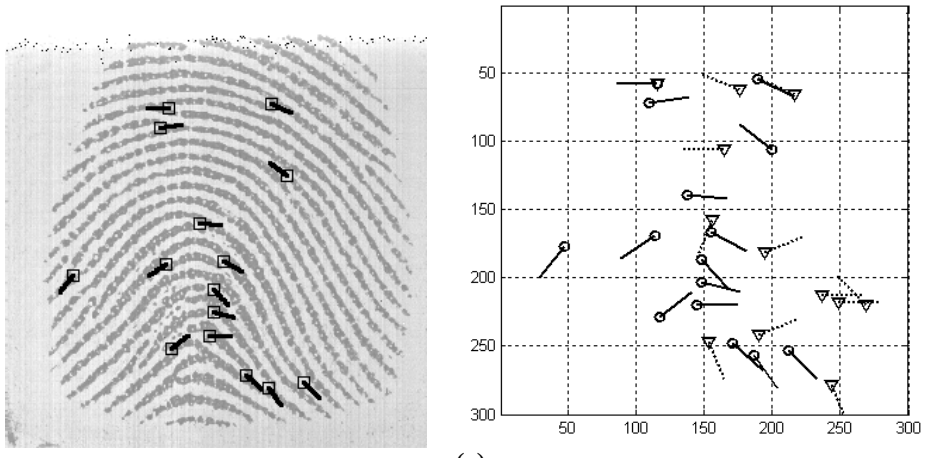


(a) (b)

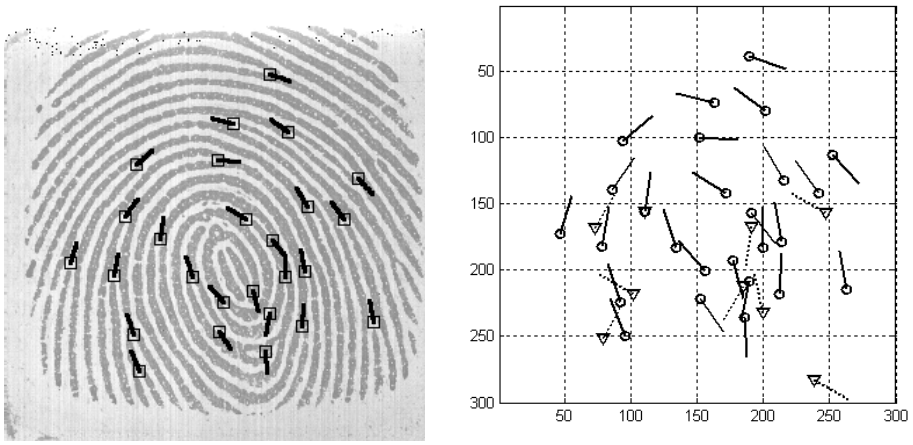


(c)

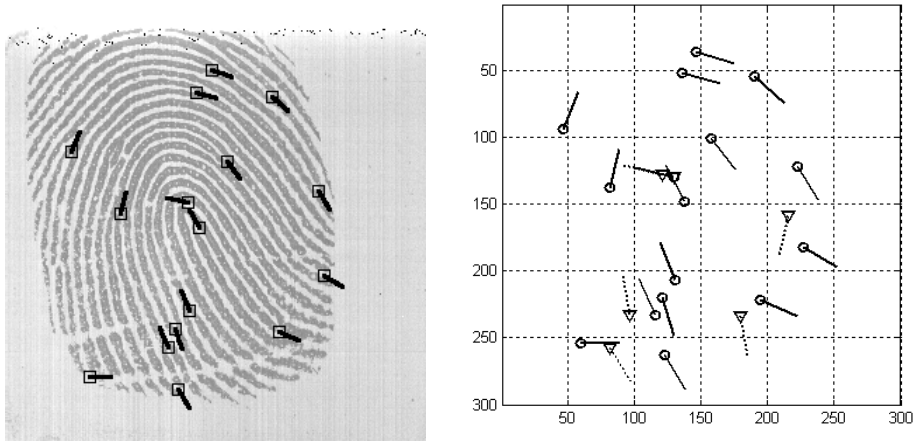
Fig. 6. Progression of matching scores for three accounts: (a) an easy to crack account, (b) a medium level account, (c) a hard to crack account.



(a)



(b)



(c)

Fig.7. Fingerprint image and minutiae comparison: (a) an easy account to crack (original template has 15 minutia, synthetic template has 17 minutia, final matching score is 12.5), (b) a medium difficulty account (original template has 26 minutia, synthetic template has 10 minutia, final matching score is 16.4), (c) a hard account to crack (original template has 16 minutia, synthetic template has 10 minutia, final matching score is 13.3). Solid lines with circle (-o) indicate the original minutiae, dotted lines with triangles ($\cdots\nabla$) indicate the synthetic minutiae.

5. SAFEGUARDS AGAINST ATTACKS

The system proposed in this paper, and other studies cited above, use the matching score as the driver for the attack systems. The information leaked from the matcher (via revealing the matching score) is used to reach a positive identification faster, e.g. compared to trying all possible input combinations (either image or feature). The trivial solution of not revealing the matching score and just outputting the accept/reject decision may not be suitable for certain biometric systems, where the matching score is necessary outside the matcher, e.g., for multibiometric systems that need matching scores from different matchers to arrive at a decision. In [10], quantizing the revealed matching scores is shown to increase the time needed for positive identification, hence decreasing the feasibility of the attack.

Another solution is to reveal the matching scores after they pass a masking procedure: with the constraint of not altering the matching result (accept or reject), outputting randomly generated scores outside the matcher breaks the correlation between the attack data and the scores, hence resulting in attack algorithm wandering around in the search space and not reaching the portion of it that guarantees positive identification. But, this may eliminate utilization of masked matching scores in a matching-score based multibiometric system, e.g., the system outlined in [19].

Another simple, but effective solution is to block matching attempts if there are too many false matches in a given period of time (e.g., it is highly unlikely that a legitimate user can provide more than, say, 20 false matches per day). The successive attempts may indicate an attack by a computer program, as proposed in this paper, for a specific template. If the attacker has enough time, even this measure may not be very effective. For example, the attacker can accumulate the results from multiple days: if 1,000 iterations are necessary for breaking into an account, the attacker can mount an attack that lasts 50 days (with 20 iterations/day) and still manage to break into the account.

6. CONCLUSIONS

After analyzing the feasibility of attacks against fingerprint-based biometric systems, we have shown that the proposed attacking system is quite effective when breaking into accounts protected with templates composed of minutiae location and angle information. The system was able to synthesize templates that guarantee positive identification in a relatively small number of attempts (271 on the average). Even though we proposed several measures to counter such attacks, each has its own limitations, especially for multimodal biometric systems. We are currently working on modified attack systems with the aim of decreasing the number of attempts even further.

REFERENCES

1. A.K. Jain, R. Bolle, and S. Pankanti, (Eds.), *Biometrics: Personal Identification in Networked Society*, Kluwer Academic Publishers, 1999.
2. N.K. Ratha, J.H. Connell, and R.M. Bolle, "An analysis of minutiae matching strength", *Proc. AVBPA 2001, Third International Conference on Audio- and Video-Based Biometric Person Authentication*, pp. 223-228, 2001.
3. B. Schneier, "The uses and abuses of biometrics", *Comm. ACM*, vol. 42, no. 8, pp. 136, Aug. 1999.
4. D. Maltoni, D. Maio, A.K. Jain, and S. Prabhakar, *Handbook of Fingerprint Recognition*, Springer, 2003.
5. Enhanced Border Security and Visa Entry Reform Act of 2002, Congress of the United States of America, http://unitedstatesvisas.gov/pdfs/Enhanced_Border_SecurityandVisa_Entry.pdf
6. T. Putte and J. Keuning, "Biometrical fingerprint recognition: don't get your fingers burned", *Proc. IFIP TC8/WG8.8, Fourth Working Conf. Smart Card Research and Adv. App.*, pp. 289-303, 2000.
7. T. Matsumoto, H. Matsumoto, K. Yamada, and S. Hoshino, "Impact of Artificial Gummy Fingers on Fingerprint Systems", *Proc. of SPIE, Optical Security and Counterfeit Deterrence Techniques IV*, vol. 4677, pp. 275-289, 2002.
8. R. Derakhshani, S.A.C. Schuckers, L.A. Hornak, and L.O. Gorman, "Determination of vitality from a non-invasive biomedical measurement for use in fingerprint scanners", *Pattern Recognition*, vol. 36, pp. 383-396, 2003.
9. N.K. Ratha, J.H. Connell, and R.M. Bolle, "Enhancing security and privacy in biometrics-based authentication systems", *IBM Systems Journal*, vol. 40, no. 3, pp. 614-634, 2001.
10. C. Soutar, "Biometric system security", http://www.bioscrypt.com/assets/security_soutar.pdf

11. A. Adler, “Sample images can be independently restored from face recognition templates”, http://www.site.uottawa.ca/~adler/publications/2003/adler-2003-fr-templates.pdf
12. C.J. Hill, “Risk of masquerade arising from the storage of biometrics”, B.S. Thesis, http://chris.fornax.net/biometrics.html
13. S. Pankanti and M.M. Yeung, “Verification watermarks on fingerprint recognition and retrieval”, *Proc. SPIE EI*, vol. 3657, pp. 66-78, 1999.
14. A. K. Jain and U. Uludag, “Hiding biometric data”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 11, pp. 1494-1498, November 2003.
15. J-P. Linnartz and P. Tuyls, “New shielding functions to enhance privacy and prevent misuse of biometric templates”, *Proc. AVBPA 2003, 4. International Conference on Audio- and Video-Based Biometric Person Authentication*, pp. 393-402, Guildford, UK, 2003.
16. A.K. Jain, S. Prabhakar, L. Hong, and S. Pankanti, “FingerCode: A filterbank for fingerprint representation and matching”, *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, vol. 2, pp. 187-193, 1999.
17. A.K. Jain, L. Hong, S. Pankanti, and R. Bolle, “An identity authentication system using fingerprints”, *Proc. IEEE*, vol. 85, no. 9, Sept. 1997, pp. 1365-1388.
18. R.M. Bolle, N.K. Ratha, A. Senior, and S. Pankanti, “Minutiae template exchange format”, *Proc. AutoID 1999, IEEE Workshop on Automatic Identification Advanced Technologies*, pp. 74-77, 1999.
19. M. Indovina, U. Uludag, R. Snelick, A. Mink and A. Jain, “Multimodal Biometric Authentication Methods: A COTS Approach”, *Proc. MMUA 2003, Workshop on Multimodal User Authentication*, pp. 99-106, Santa Barbara, CA, December 11-12, 2003.