Vulnerabilities in biometric encryption systems

Andy Adler

School of Information Technology and Engineering,
University of Ottawa, Ontario, Canada
adler@site.uOttawa.ca [**]

Abstract. The goal of a biometric encryption system is to embed a secret into a biometric template in a way that can only be decrypted with a biometric image from the enroled person. This paper describes a potential vulnerability in such systems that allows a less-than-brute force regeneration of the secret and an estimate of the enrolled image. This vulnerability requires the biometric comparison to "leak" some information from which an analogue for a match score may be calculated. Using this match score value, a "hill-climbing" attack is performed against the algorithm to calculate an estimate of the enrolled image, which is then used to decrypt the code. Results are shown against a simplified implementation of the algorithm of Soutar et al. (1998).

1 Introduction

Traditional biometric technology tests for a match between a new image of an individual and the key biometric features of an original image stored in a biometric template. If the biometric software detects a match, further processing in a security system is activated. This often involves the release of security tokens or password codes to enable other applications. There are several potential concerns with such systems; in this paper we consider the concern that all the information needed to relase the codes must somehow be available to the software. It is therefore theoretically possible to compromise any traditional biometric system in order to gain secure access without presenting a biometric image [10]. At the same time, it may be possible to get information about the enrolled person from their biometric template [2][16].

Biometric encryption is designed to avoid these problems by embedding the secret code into the template, in a way that can be decrypted only with an image of the enrolled individual. [5][14]. Since the secret code is bound to the biometric template, an attacker should not be able to determine either the enrolled biometric image or secret code, even if they have access to the biometric software and hardware.

While such biometric encryption systems are not widely deployed, they appear to offer some compelling benefits for many applications [19]. The benefit of biometric encryption is perhaps most important for mobile applications of

[**] This work is supported by NSERC Canada

biometrics, such as for cell phones or laptop computers, or in biometric-based identity cards, such as those designed into many new national passports. Another important application of biometric encryption is for control of access to digital content, with the primary interest being in preventing copyright infringement. Digital documents encoded with the biometric of the user(s) with approved access will presumably be subject to attacks, especially since both the documents and the software to access them will be widely distributed [10]. Finally, biometric encryption promises to help address the privacy concerns of biometric technology [17][19].

The primary difficulty in designing biometric encryption systems is the variability in the biometric image between data measurements. For example, a fingerprint image changes with applied pressure, temperature, moisture, sweat, oil, dirt on the skin, cuts and other damage, changes in body fat, and with many other factors. In the case of biometric encryption, this means that the presented biometric image cannot itself be treated as a code, since it varies with each presentation. For biometric encryption systems, this variability becomes especially difficult. An algorithm must be designed which allows an image from the enrolled person, with significant differences from the original, to decode the complete secret code. At the same time, an image from another person – which may only be slightly more different from the enrolled image – must not only not decode the secret, it must not be allowed to decode (or "leak") any information at all.

This paper develops one approach to attack biometric encryption algorithms, based on using any "leaked" information to attempt a "hill-climbing" of the biometric template. We show that this approach can successfully reconstruct a face image from a biometric encryption scheme based on [14][15]. We then discuss recent work in this area and some possible improvements to this attack.

2 Image reconstruction from biometric templates

As discussed in [7], a biometric encryption system must have error tolerance, such that, for an enrolled image IM_{enroll}, it must be possible to perform the decryption for an input IM' which is sufficiently close (in which "close" is defined in some distance space appropriate to the biometric modality). For an IM' further from IM_{enroll} than some threshold, it must not only be infeasible to decrypt, but it must be impossible to obtain any statistical information about IM_{enroll}. The essence of the proposed attack on biometric encryption is to use this type of "leaked" information to iteratively improve an estimate of the enrolled biometric, which is then used to decrypt the secret code. Unfortunately, it is difficult to design an encryption algorithm to give complete information for a "close" answer, but no information for a slightly less accurate one [4][7][11][19].

In order to use the "leaked" information, it is necessary to construct a measurement which functions as a *match score*, ie. a measure which increases with the similarity of IM' to IM_{enroll}. Several authors have shown that, given access to match score data, it is possible to reconstruct a good estimate of an unknown enrolled image [16] from a fingerprint [9][20] or face recognition template [2].

These algorithms use a “hill-climbing” strategy. A test image is presented to a biometric algorithm and compared to an unknown enrolled image to obtain a match score. Then, iteratively, modifications are made to the input, and those that increase the match score are retained. Eventually, a best-match image is generated, which resembles the essential features of the unknown enrolled image, and is able to compare to it at high match score. In order to protect against this attack, the BioAPI [3] specifies that match scores should be quantized. However, recently, we have shown that the hill-climbing attack can be modified to overcome the effects of quantization [1] (for reasonable levels of quantization, ie. where one quantization level corresponds to a 10% change in match confidence).

Tests in this paper show that the modified hill-climbing algorithm is required for attacks against the biometric encryption algorithm. This appears to be because match scores calculated from biometric encryption algorithms are not easily related to traditional biometric match score values, and often it is only possible to calculate a quantized value. For example, with an error correcting code, the match score may be the number of bits that require correction, resulting in a heavily quantized score.

2.1 Quantized Hill-climbing

This section describes the quantized hill climbing algorithm used to the attack the biometric encryption technique [1]. It has been shown to work successfully for face recognition systems; however, recent work [9][19] suggests that it is extensible to fingerprint biometrics. The algorithm has the ability to obtain match scores (MS) of the target compared to an arbitrarily chosen image (IM). We represent this function as:

$$MS = \mathsf{compare}(IM, IM_{enroll}) \tag{1}$$

A schematic diagram of this algorithm is shown in Fig. 1. It is implemented as follows:

1. *Local database preparation*: A local database of frontal pose face images is obtained. Images are rotated, scaled, cropped, and histogram equalized.
2. *Eigenface calculation*: Use a principle components analysis (PCA) decomposition to calculate an set of eigenimages (or eigenfaces) from the local image database [18], using the method of Grother [8]. Divide each image into four quadrants (Fig. 1, left). Quadrant eigenimages ($EF_{i,quadrant}$) are then defined to be equal to EF_i within the quadrant and zero elsewhere. The edge of each quadrant is then smoothed to provide a gradual transition over 10% of the image width and height.
3. *Initial image selection*: Choose an initial estimate (IM_0), which is subsequently iteratively improved in the next step. The selected image could be random, or could be the one with the largest MS.
4. *Iterative estimate improvement*: Iterate for step number i. Repeat iterations until MS is maximum, or there is no more improvement in MS.

(a) Randomly select a quadrant Q. The diametrically opposite quadrant is referred to as OQ.
(b) Randomly select an eigenimage, k; the component in Q is $EF_{k,Q}$
(c) Generate an image RN, consisting of random Gaussian noise in OQ and zero elsewhere.
(d) Calculate the amount of RN which reduces the quantized match score by one quantization level. Using a bisection search, calculate a minimum value n such that

$$\mathsf{compare}(IM_i, IM_{enroll}) > \mathsf{compare}(IM_i + nRN, IM_{enroll}) \quad (2)$$

(e) Iterate for j for a small range of values c_j

$$MS_j = \mathsf{compare}(IM_i + nRN + c_j EF_{k,Q}, IM_{enroll}) \quad (3)$$

(f) Select j_{max} as the value of j for the largest MS_j.
(g) Calculate

$$IM_{i+1} = IM_i + c_{j_{max}} EF_{k,Q} \quad (4)$$

(h) Truncate values to image limits (ie. 0 to 255) if any pixel values of IM_{i+1} exceed these limits.

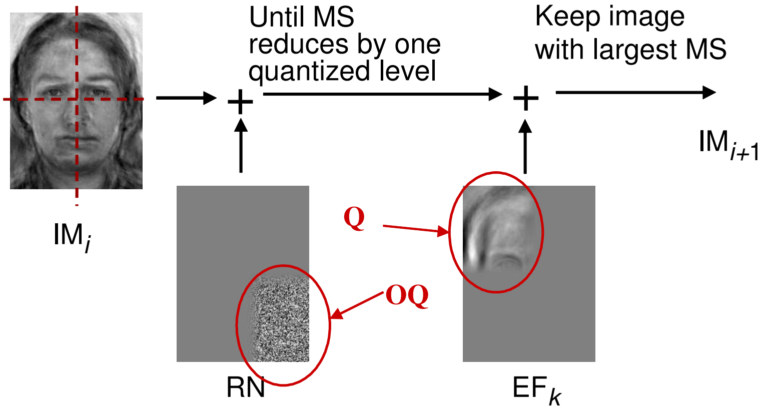

Fig. 1. Schematic diagram of the hill-climbing algorithm for quantized match scores. In each iteration, the candidate image is first "worsened" with the addition of random noise to a quadrant, until the match score is below a quantized level. Then a component of an eigenimage is added to the opposite quadrant, and the maximum match score output is retained.

Because the quantized match score will not normally give information to allow hill climbing, a carefully chosen level of noise is introduced into the opposite image quadrant, in order to force the quantized score into a range where its information can once again be used. The local database does not need to resemble the target image, and may be one of the many freely available face image databases (for example [12][13]).

3 Biometric Encryption

This paper considers the fingerprint biometric encryption algorithm of Soutar et al. [14]. This algorithm was chosen because it represents a concrete system which has been implemented and for which the details are well described. Bioscypt Inc. (the employer of Soutar) has indicated that significant enhancements were made to this algorithm after the published version. However, this paper simply presents a framework for an attack, and not necessarily a break of a specific, implemented, algorithm. For a review of other recent biometric encryption systems, refer to [7][19].

Enrollment requires several sample images, and a secret code, and creates a template binding the code to the images. This differs for some other systems, such as that of Davida et al. [5][6], in which the biometric image forms a unique key. The system under consideration [14] calculates a template related to the input image by frequency domain correlations. We describe a simplified operation of this system, using slight variations in notation from [14]. During enrollment, an average image f_0 is obtained (with 2D Fourier transform $F_0(u)$) from multiple samples of the input fingerprint, after suitable alignment. In order to encode the secret, a random code is chosen and encoded as a phase-only function $R_0(u)$ such that the amplitude is one and the phase is $e^{\pm\pi/2}$ (selected randomly). Using F_0 and R_0, a filter function $H(u)$ is calculated based on a Wiener inverse filter, as

$$H_0 = \frac{F_0^* R_0^*}{F_0^* F_0 + N^2} \tag{5}$$

where * denotes the complex conjugate, and N^2 the image noise power. For this algorithm, N encodes the expected variability between images. As N increases, an image more dissimilar from the one enrolled can decrypt the code, at the expense of a smaller secret.

In order for biometric encryption to allow for variability in the input image, the secret code must be robustly encoded, using some sort of error correcting code (ECC) framework. [14] uses a simple ECC based on Hamming distances and majority decision. The secret is encoded by linking it with the sign of the complex component R_0. Each bit of the secret is associated with L locations in R_0 with the same phase angle. These associations are then stored in the template in a “link table”. Majority decision requires that L be odd; [15] appears to recommend $L = 5$. For example, if the 4^{th} bit of the secret is a 1, position 4 of the link table will point to five positions in R_0 with a phase of $e^{+\pi/2}$, while if the bit is 0, position 4 will point to five positions with phase $e^{-\pi/2}$. The template is created containing the following information: H_0, the link table, a cryptographic hash of the secret, and an identifier. The cryptographic hash and identifier are to detect errors in storage and software processing, and do not concern us here.

During *key release*, a new image f_1 is acquired. This image is deconvolved with the filter H_0 to calculate R_1, an estimate of R_0.

$$R_1^* = sign(imag(H_0 F_1)) \tag{6}$$

It appears that the sign of the imaginary component of the phase of R_1 is the most robust feature of this calculation [15]. If F_1 is from the same individual as F_0, then R_1 should be a good estimate of R_0. The link table is used to extract the phase locations into which each bit is encoded. Since $R_1 \neq R_0$, some phase elements will be incorrect; however, if R_1 is sufficiently close, the use of majority decision should allow the correct value of the secret to be obtained.

4 Results

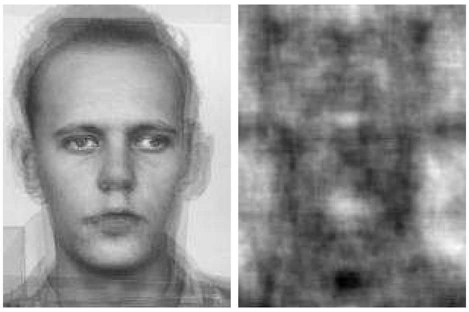

Fig. 2. Sample images for an implementation of the biometric encryption technique of [14] applied for a face recognition. *Left*: Image f_0 averaged from five samples. *Right*: Template h_0 including the random phase encoded elements.

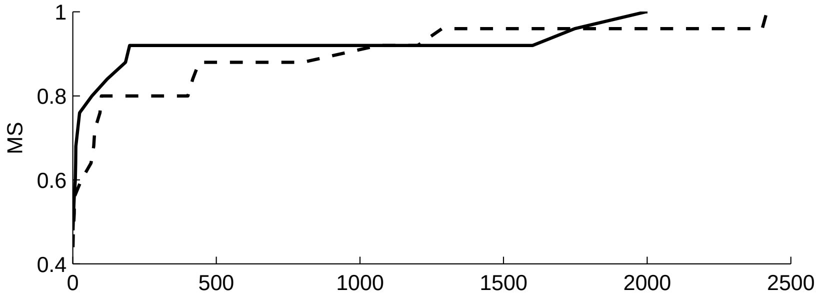

Fig. 3. Match score MS versus iteration number. The match score is calculated as the number of bit positions matching in the template. A MS of 1.0 indicates a perfect match. Solid and dashed line corresponds to top and bottom images in 4, respectively.

In order to apply the attack of section 2.1, it is necessary to create a match score from the template. For the biometric encryption system of [14] this is

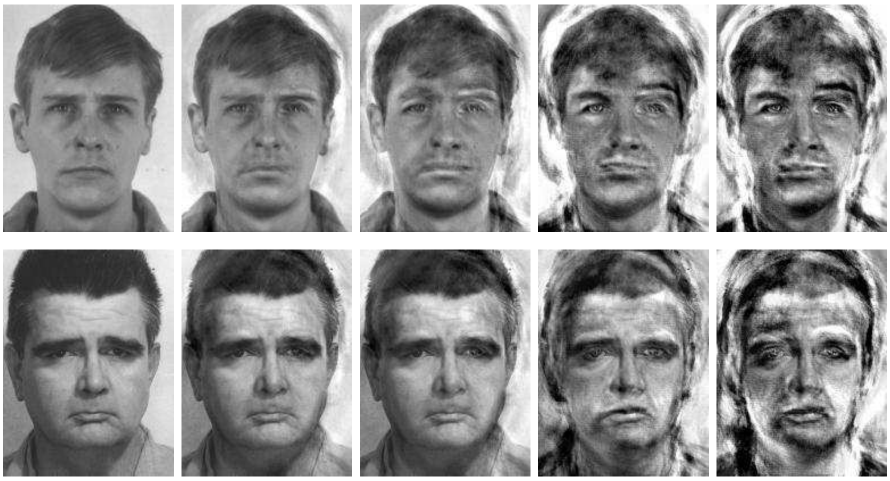

Fig. 4. Sample images of IM_k for as a function of iteration for two different initial images (top and bottom row). Left image is IM_0 and right image yields $MS = 1.0$.

relatively straightforward. If $R_1 = R_0$, then all phases corresponding to each bit position in the link table will be equal, while for a random image, approximately half of the elements will match. We thus create a match score MS from the R_1 based on the difference between the number of ones and zeros in the link table, as

$$MS = \frac{1}{LB} \sum_{i=1}^{B} \left| \sum_{j=1}^{L} (LT_{ij} = 0) - \sum_{j=1}^{L} (LT_{ij} = 1) \right| \tag{7}$$

where LT_{ij} is the value of the link table entry for the j^{th} element of bit i, and B is the number of bits of secret. The maximum MS is 1; the minimum possible MS is $\frac{1}{L}$, and statistical considerations show a random image will typically give $MS = \frac{3}{8}$ of the maximum for $L = 5$.

We implemented the algorithm of section 3 for use with face recognition biometrics; the only modification required was to test which part of the Fourier transformed image F_0 produced reliable phase values to be encoded in the link table. The 13 × 13 low frequency 2D Fourier components appeared to be the most reliable for this application. The advantage of this implementation is that the framework and software previously developed for hill-climbing for face recognition in [1] would be applicable. On the other hand, such an algorithm is not realistic. Because face recognition data is not very distinctive, it would not be possible to encode many bits of a key (our initial results would suggest a maximum of about 20 bits). A template was created using 5 images from the NIST Mugshot Identification Database [12], and 20 secret bits were encoded using $L = 5$. In order to illustrate the power of the algorithm, an initial image intentionally different from the template was chosen. Fig. 2 shows an image of the averaged enrollment images from the template (f_0), and the encoded template (h_0). All images were scaled and rotated to have a common size and eye locations.

Results show that the template recreation algorithm is quickly able to attain a perfect match to F_0 ($MS = 1$), even though the resulting images are not very similar to the enrolled image. This is significantly larger than match values for other images of the enrolled individual (which were typically accurate to $MS = 0.82 - 0.86$). Fig. 3 shows the graph of MS versus iteration number for $L = 5$, while Fig. 4 shows a selection of images IM_k of the progress of the algorithm for $L = 5$ for two different initial images. There is an initial rapid increase in MS after which the algorithm shows a more gradual improvement. It is interesting to note that IM begins to show some similar features to f_0 as iteration progresses. For example, the position of eyebrows, and shape of eyes, nose and chin and outline of the face begin to show a resemblance. One interesting aspect is that the hill-climbing algorithm does not seem to terminate with a final good estimate of the template image. Perhaps biometric encryption allows several possible variants of the enrolled image to match.

5 Discussion

This paper presents an approach to attack biometric encryption algorithms in order to extract the secret code with less than brute force effort. A successful result was obtained for a simplified version of the biometric encryption algorithm of [14]. Essentially, this attack requires that the some information be "leaked" from the biometric match for sample images very dissimilar from the enrolled one. This leaked information is used to construct a match score, which is subsequently used to iteratively improve an estimate.

While this work was implemented against a specific algorithm [14], several more recent systems have been proposed, which appear to be somewhat less susceptible to this vulnerability. For example, the fingerprint algorithm of [4], encodes the secret as the coefficients of a Galois field polynomial. Minutiae points are encoded as pairs (x_i, y_i) where x_i is a minutiae point, and y_i is a point on the polynomial. Additionally, numerous "chaff" points are encoded, in which the value of y_i is random. During key release, the minutiae of the new fingerprint image are calculated, and the points x_i closest to the minutiae are chosen. The y_i corresponding to these points are used to estimate the polynomial, using a Reed-Solomon error correcting code framework. If enough legitimate points are taken, the correct polynomial will be obtained and the correct secret decrypted. This encryption technique is based on the "fuzzy vault" technique of [11]. An interesting generalization of this scheme is given by the "secure sketches" of [7]. We believe that it may be possible to use the attacks of this paper against the biometric encryption technique of [4], even though Juels and Sudan [11] were able to give a proof of security. A key assumption for security proof is that the data held in the "fuzzy vault" are random. The data of [4], however, are not. Firstly, biometric data is inherently structured – otherwise hill-climbing wouldn't be possible. Secondly, the need to carefully place chaff minutiae points sufficiently far from legitimate ones is another source of non-randomness. However, at this time, we are not able to demonstrate an attack against this technique.

In their analysis, Uludag et al. [19] note that most proposed biometric encryption systems only appear to account for a "limited amount of variability in the biometric representation." In order to quantify this notion, experiments were conducted by them to estimate the variability in fingerprint minutiae. Matched fingerprint pairs were imaged and minutiae locations identified by a human expert, which was assumed to give an upper bound on system performance. Using these data, the algorithm of [4] was analyzed to estimate the *FMR*/*FNMR* tradeoff curve during key generation and key release. Results were surprisingly poor; an equal error rate of 6.3% can be estimated from the results, although the authors note that there are a limited number of feasible operating points. This means that such systems could be feasibly attacked by successively presenting biometric samples from a representative population.

In conclusion, this paper has presented a scheme that appears to show vulnerabilities in biometric encryption systems. The attacker can regenerate an estimate of the enrolled biometric image and use it to release the stored secret. The attacker considered here, who has access to biometric templates and authentication software, is quite plausible, as such biometric templates may be stored in standardized formats on identity documents or portable devices.

References

1. Adler, A.: "Images can be regenerated from quantized biometric match score data", *Proc. Can. Conf. Elec. Comp. Eng.* 469–472 (2004)
2. Adler, A.: "Sample images can be independently restored from face recognition templates" *Proc. Can. Conf. Elec. Comp. Eng.* 1163–1166 (2003)
3. BioAPI Consortium: *BioAPI Specification* `http://www.bioapi.org/BIOAPI1.1.pdf` 1163–1166 (2001)
4. Clancy, T.C., Kiyavash, N., Lin, D.J.: "Secure smartcard-based fingerprint authentication" *Proc. ACMSIGMM 2003 Multimedia, Biometrics Methods and Applications Workshop* 45–52. (2003)
5. Davida, G.I., Frankel, Y., Matt, B.J.: "On enabling secure applications through off-line biometric identification" *Proc. IEEE Symp. Privacy and Security* 148–157 (1998)
6. Davida, G.I., Frankel, Y., Matt, B.J., Peralta, R.: "On the relation of error correction and cryptography to an offline biometric based identification scheme" *Proc. Conf. Workshop Coding and Cryptography (WCC99)* 129–138.
7. Dodis, Y., Reyzin, L., Smith, A.: "Fuzzy Extractors and Cryptography, or How to Use Your Fingerprints", Proc. Eurocrypt'04, (2004) `http://eprint.iacr.org/2003/235/`
8. Grother, P.: "Software Tools for an Eigenface Implementation" National Institute of Standards and Technology, (2000) `http://www.nist.gov/humanid/feret/`
9. Hill, C.J.: *Risk of Masquerade Arising from the Storage of Biometrics* B.S. Thesis, Australian National University, 2001 `http://chris.fornax.net/biometrics.html`
10. Kundur, D., Lin, C.-Y., Macq, B., Yu, H.: "Special Issue on Enabling Security Technologies for Digital Rights Management" Proc. IEEE **92** 879-882, (2004).

11. Juels, A., Sudan, M.: “A fuzzy vault scheme” *Proc. IEEE Int. Symp. Information Theory* 408 (2002)
12. National Institute of Standards and Technology (NIST): *NIST Special Database 18: Mugshot Identification Database (MID)* `http://www.nist.gov/srd/nistsd18.htm`
13. Phillips, P.J., Moon, H., Rauss, P.J., Rizvi, S.: “The FERET evaluation methodology for face recognition algorithms” *IEEE Trans. Pat. Analysis Machine Int.* **22** 1090–1104 (2000)
14. Soutar, C., Roberge, D., Stoianov, A., Gilroy, R., Vijaya, B.: “Biometric Encryption using image processing”, Proc. SPIE Int. Soc. Opt. Eng., **3314** 178-188 (1998)
15. Soutar, C., Roberge, D., Stoianov, A., Gilroy, R., Vijaya, B.: “Biometric Encryption: enrollment and verification procedures”, Proc. SPIE Int. Soc. Opt. Eng., **3386** 24-35 (1998)
16. Soutar, C., Gilroy, R., Stoianov, A.: “Biometric System Performance and Security”, Conf. IEEE Auto. Identification Advanced Technol., (1999). `http://www.bioscrypt.com/assets/security_soutar.pdf`
17. Tomko G.: “Privacy Implications of Biometrics - A Solution in Biometric Encryption”, 8th Ann. Conf. Computers, Freedom and Privacy, Austin, TX, USA, (1998).
18. Turk, M.A., Pentland, A.P.: “Eigenfaces for recognition” *J. Cognitive Neuroscience* **3** 71–86 (1991)
19. Uludag, U., Pankanti, S., Prabhakar, S., Jain, A.K.: “Biometric Cryptosystems: Issues and Challenges”, *Proc. IEEE* **92** 948–960 (2004)
20. Uludag, U.: “Finger minutiae attack system” *Proc. Biometrics Conference*, Washington, D.C. USA. Sept. (2004)