1-1-2005

Cancelable key-based fingerprint templates

Russell Ang
University of Wollongong

Reihaneh Safavi-Naini
University of Wollongong, rei@uow.edu.au

Luke F. McAven
University of Wollongong, lukemc@uow.edu.au

Cancelable key-based fingerprint templates

Abstract

Biometric based authentication can provide strong security guarantee about the identity of users. Security of biometric data is particularly important as compromise of the data will be permanent. Cancelable biometrics store a non-invertible transformed version of the biometric data and so if the storage is compromised the biometric data remains safe. Cancelable biometrics also provide a higher level of privacy by allowing many templates for the same biometric data and hence non-linkability of user's data stored in different databases. We define how to measure the success of a particular transformation and matching algorithm for fingerprints. We consider a key-dependent geometric transform that is applied to the features extracted from a fingerprint, to generate a key-dependent cancelable template for the fingerprint. We investigate performance of an authentication system that uses this cancelable fingerprint when a fingerprint matching algorithm is used for detection. We evaluate performance of the system and show the challenges of achieving good performance if the matching algorithm is not modified.

Keywords

Cancelable, key, based, fingerprint, templates

Disciplines

Physical Sciences and Mathematics

Publication Details

Ang, R., Safavi-Naini, R. & McAven, L. F. (2005). Cancelable key-based fingerprint templates. In C. Boyd & J. Gonzalez Nieto (Eds.), Australasian Conference on Information Security and Privacy (pp. 242-252). Germany: Springer.

Cancelable key–based fingerprint templates

Russell Ang, Rei Safavi–Naini and Luke McAven

School of Information Technology and Computer Science, University of Wollongong, Northfields Avenue, NSW 2522, Australia.
[rja02,rei,lukemc]@uow.edu.au

Abstract. Biometric based authentication can provide strong security guarantee about the identity of users. Security of biometric data is particularly important as compromise of the data will be permanent. Cancelable biometrics store a non–invertible transformed version of the biometric data and so if the storage is compromised the biometric data remains safe. Cancelable biometrics also provide a higher level of privacy by allowing many templates for the same biometric data and hence non-linkability of user's data stored in different databases. We define how to measure the success of a particular transformation and matching algorithm for fingerprints. We consider a key–dependent geometric transform that is applied to the features extracted from a fingerprint, to generate a key–dependent cancelable template for the fingerprint. We investigate performance of an authentication system that uses this cancelable fingerprint when a fingerprint matching algorithm is used for detection. We evaluate performance of the system and show the challenges of achieving good performance if the matching algorithm is not modified.

1 Introduction

User authentication is becoming increasingly important. Integrity of data and transactions in various applications relies on authenticity of participants' identities.

The three basic forms of user authentication that can be used independently or in combination with others, are *knowledge based* which rely on a secret such as a password held by the user, *token based* that rely on possession of a 'token', such as a physical key or smartcard and *biometric based* that uses unique characteristics of individuals, such as fingerprints or voice prints.

While knowledge can be forgotten, and tokens stolen or lost, biometrics do not suffer from these deficiencies, and can provide the security of long passwords without sacrificing the ease of memorizing short ones.

Biometric authentication has two phases, *enrolment* and *authentication* (or *verification*). Enrolment involves measuring an individuals biometric data to construct a *template* for storage. Authentication involves a measurement of the same data and comparison with the stored template.

Biometric readings are rarely identical and readings are environment and device dependent. A template provides an approximate version of the biometric data, and verification succeeds if the second reading is close to the stored template.

Biometric characteristics are largely immutable and as such their compromise is permanent. For fingerprints, template compromise may allow the construction of artificial fingerprints [5]. Fingerprint images can be artificially generated [1], and there exists commercial software which is able to generate synthetic fingerprints [9].

Even without database compromise, biometric possession by government or medical organisations provides the potential for information misuse as the data belonging to the same user can be easily linked. Cancelable biometrics store a transformed version of the biometric data. The transformation is one way and so knowledge of a transformed biometric does not leak information about the actual biometric data. Moreover, by using different cancelable templates, data belonging to the same user cannot be linked. In this paper, we consider cancelable biometrics that are generated through a keyed transformation and investigate performance of the system when the verification method stays the same as the one used for the original fingerprint template.

Related work

Encrypting the template prior to storage can make template compromise harder, but introduces key management problems. Furthermore, key compromise results in the full template being revealed, compromising the biometric. A *fingerprint vault* [3], is a fuzzy vault [7] that uses the feature set of a fingerprint to encode a secret in a polynomial. Biometric readings close to the feature set allows the polynomial to be reconstructed and the secret to be extracted. Implementation of this system using real biometric data requires an appropriate representation for this data. It is shown [3] that finding the secret in the fingerprint vault requires 2^{69} trials, to each valid users trial. However, the system has a high false rejection rate of about 30%, which is unacceptable in practice.

Another approach is to store a "hash" of the biometric data, rather than the biometric data itself. Cryptographic hashes are bit sensitive and not suitable for matching two readings that are slightly different. Biometric hashes have been described which allow 'close' biometric readings to be hashed into the same hash value. For example, [13] describe a biometric hash for handwriting.

Fuzzy commitment schemes [8] commit a secret using fuzzy data x, such as biometric data. An approximate version of x can recover the committed secret. The scheme uses error correcting codes and decommitment requires decoding operations which could be computationally inefficient.

Cancelable biometrics [11] apply non–invertible transforms to the biometric template and store the result. For verification, a biometric reading undergoes the same transformation before comparison with the stored (transformed) template. In a well designed system, two transformed outputs will 'match' if the initial templates 'matched' under the template matching algorithm. That is, the transform will not affect the matching result or the outcome of verification. The biometrics are cancelable in the sense that one cannot derive the original fingerprint from the transformed template, and comprising the stored template doesn't compromise the biometric characteristic of the user. Different cancelable biometrics can be given to different collectors, for example government and health bodies, to ensure that misuse minimises relationship leaking between databases.

1.1 Our contribution

We consider a fingerprint authentication system that uses a key dependent non–invertible transformation applied to readings before storage. Using a key dependent transformation allows us to have different stored template for different applications (databases) so reducing the chance of linkability of information related to a person. The system can be seen as a key–dependent cancelable biometric system [11]. The key–dependent transformation we use is applied on a minutiae based representation of fingerprint data. We use the fingerprint template matching algorithm of [6], to match transformed templates. Although the transformation affects the structure of the template we justify why using the same matching algorithm is meaningful. The algorithm performs well for untransformed fingerprint templates and the aim of our experiment is to investigate suitability of the matching algorithm, or a simple modification thereof, for the transformed template. We find the matching score is affected by the transform, so the resulting system has increased false acceptance and false rejection rates, and so cannot be effectively used in practice. Another shortcoming of the proposed transform is its insensitivity to small changes to the key resulting in a reduced key space. Our work will hopefully motivate further work in this important area.

The rest of this paper is structured as follows. In Section 2 we describe fingerprint structure. Section 3 contains the user authentication procedures, and a description of the matching algorithm used [6].

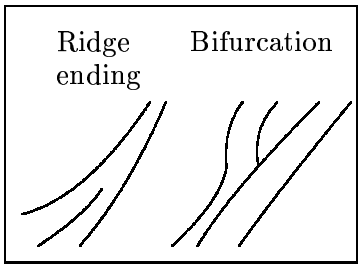
In Section 4 we describe a generic process for implementing key–based transforms for cancelable key–based biometrics. We define a measure of the success of a transform and matching algorithm.

Section 5 contains the transform we adopt as an illustration. In section 6 we analyse the results of implementing this transform. We summarise our results in Section 7.

2 Fingerprints

Fingerprint recognition is probably the oldest method of biometric identification applicable to all persons, apart from the obvious facial recognition used in everyday life. A fingerprint consists of *ridges*, lines across fingerprints, and *valleys*, spaces between ridges. The ridge and valley pattern is unique to each individual. The three basic macroscopic features in ridge flow are arches, loops and whorls. Discontinuities in the ridges are referred to as *minutiae*[9]. Most fingerprints have between 70 and 150 minutiae.

The two major methods of fingerprint matching are *minutiae matching* and *global pattern matching*. The former analyses relative positions and structure of minutiae, while the latter considers macroscopic ridge flows. Most automated systems use minutiae matching. With suitable resources this method is more accurate and can be faster. Performance is very dependent on image quality and the resolution of scanned fingerprints though. It can also be expensive in processor and monetary terms, and potentially requires a large database. We use minutiae based matching, applied locally and globally following [6]. While about 18 minutiae types are recorded, the 2 most frequently occurring, and most frequently used in matching, are bifurcation and ridge ending. These types are formally defined in [2], where a bifurcation is thought of as a valley ending.



One cannot be sure even of the type being the same in different readings, since the distinction between bifurcation and ridge ending is not as clear in practice as in theory [2]. Pressure on the finger during a reading can squash adjacent ridges so that a ridge ending appears as a bifurcation.

3 Authentication using fingerprints

There are two phases in authentication using fingerprints.

1. A user U *enrols* through a trusted service that uses the measurement of a fingerprint to generate a template $\mathcal{T}$. The identity and template, $(U, \mathcal{T})$, are stored in a database.
2. To *authenticate* a user U produces a fingerprint reading $\mathcal{R}$. If $\mathcal{T}$ and $\mathcal{R}$ are close the user is accepted as being U.

Two important parameters in determining the quality of an authentication system are the *false rejection rate* (FFR) and the *false acceptance rate* (FAR). FFR records how often authentication fails when it should succeed, while FAR records how often authentication succeeds when it should fail. An overall measure is given by the point where FFR equals FAR. The error at this level is referred to as the *equal error rate* (EER).

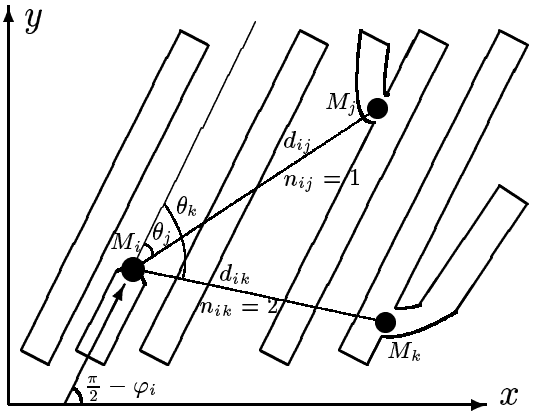
3.1 Enrolment: Template generation, processing and storage

Enrolment uses a fingerprint capture device [10] to generate a fingerprint image. Processing tools, such as in the VeriFinger toolkit [12], are used to tidy up the image. Edge detection is used, for example. After processing, features such as minutiae can be readily identified. Statistical techniques are often used to choose statically reliable features [3]. We use a single reading to form a template, since this simply preserves the ridge structure.

We represent the fingerprint template $\mathcal{T}$ by a minutiae set, referred to as the *feature set*, identified on the fingerprint. The set is obtained using `MINDTCT` [4], and aligned according to the `R92` algorithm [4]. The template $\mathcal{T}$ containing the minutiae is stored in an encrypted database.

For each minutiae M_i we have a vector containing position (x_i, y_i), local ridge orientation φ_i, type t_i, and a local feature vector F_i. Euclidean d_{ij} and ridge n_{ij} distances between M_i and M_j are given, the latter being the number of ridges between M_i and M_j in a straight line. It determines which minutiae are closer.

The angle θ_j between ridges at M_i and M_j is used. For neighbours M_j we record type and the angle φ_{ij} of the ridge at M_j relative to the ridge angle at M_i. The vector F_i includes $(d_{ij}\theta_j\varphi_{ij}n_{ij}t_j)$ for each of L neighbours M_j. The local minutiae geometry is shown here for $L = 2$.



3.2 Verification: Authentication and the matching process

Verification requires that a user present their finger for reading. Processing applied in the template generating process is also applied here, and again the fingerprint can be recorded as a set of minutiae, $\mathcal{R}$. Comparing $\mathcal{T}$ and $\mathcal{R}$, using a matching algorithm, produces a matching score. If the score is above a threshold, the user is accepted. We adopt a matching algorithm [6] with elements of local and global matching.

Local matching uses $F_i^{\mathcal{T}}$ and $F_j^{\mathcal{R}}$ for template and reading minutiae, respectively. Local matching is less reliable but is rotation and translation invariant, since it depends only on local feature vectors which have those invariances. We define [6] a weight vector W, containing a weight for each element of the F_i. This allows us to vary tolerance distribution. We use the same, empirically chosen, weight parameters as [6].

We specify a threshold parameter $t_p = 6(5L+1)$, for L nearest neighbours. We also define a local structure similarity function $s_l(i,j)$;

$$s_l(i,j) = \begin{cases} \frac{t_p - W.|F_i^{\mathcal{T}} - F_j^{\mathcal{R}}|}{t_p} & \text{if } W.|F_i^{\mathcal{T}} - F_j^{\mathcal{R}}| < t_p \\ 0 & \text{Otherwise} \end{cases}$$

so that $s_l(i,j) = 1$ for a perfect match and 0 for a mismatch.

One calculates $s_l(i,j)$ for all pairs of minutiae M_i, M_j from $\mathcal{T}$ and $\mathcal{R}$, respectively. One adopts the structure supported by matching the local neighbourhoods given corresponding i to j for $\max_{i,j} s_l(i,j)$. Within this structure one considers matching the global parameters of the minutiae; distance, angle and ridge angel relative to the reference minutiae.

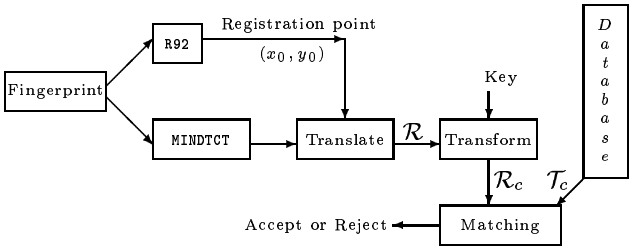
The reference minutiae in $\mathcal{T}$ and $\mathcal{R}$ are M_i and M_j, respectively, since they are taken to be in correspondence. Each minutiae in $\mathcal{R}$ is matched against each in $\mathcal{T}$, and a non–zero score $s_g(i,j)$ is given, $\frac{1}{2} + \frac{1}{2} s_l(i,j)$, if the points i and j are within some bounding box of each other in the global parameter space. One avoids double matching by setting $s_g(i,j)$ to zero if there any exists a k such that $s_g(i,k) \geq s_g(i,j)$ or $s_g(k,j) \geq s_g(i,j)$. The overall matching score is expressed as

$$M_s = 100 \frac{\sum_{i,j} s_g(i,j)}{\max\{M,N\}} \ .$$

A reading $\mathcal{R}$ is accepted as validating the user if the matching score between $\mathcal{R}$ and $\mathcal{T}$ is higher than some threshold. We note, in particular, that spurious minutiae tend to count negatively to the matching score in that they increase M and N without adding any $s_g(i,j)$.

4 Enrolment and authentication for key–based templates

We add an extra step to the system in Section 3. Prior to storage a non–invertible mapping, our example of which is described in Section 5, is applied to the template. For us this mapping is a global transformation which acts on the global minutiae parameters and reflects some minutiae across a line. The local feature vectors are calculated after this transformation, using the ridge structure at their new location. Before a comparison with the stored template, the same mapping is applied against the reading. We show the modified verification process here.



We give two scenarios for key storage. Let $\mathcal{T}$ be a biometric template and $\mathcal{T}_c = C_k(\mathcal{T})$ the keyed cancelable biometric.

Scenario One: User key only storage.

The user *enrols* through a trusted service that uses a key k to generate a cancelable biometric $\mathcal{T}_c$ from k and the template $\mathcal{T}$ generated from the user's biodata. The user receives a smart card containing k, and the database stores $(\texttt{User}, \mathcal{T}_c)$.

For *verification*, a user presents $(\texttt{User}, \mathcal{R}, k)$ to the server. The reading $\mathcal{R}$ and key k are used, by the server, to calculate a cancelable biometric $\mathcal{R}_c$. The server compares $\mathcal{R}_c$ and $\mathcal{T}_c$ and, if close, the user is accepted.

Scenario Two: Database key and template storage.

The user *enrols* through a trusted service that uses a key k to generate a cancelable biometric $\mathcal{T}_c$ from k and the template $\mathcal{T}$ generated from the user's biodata. The database stores $(\texttt{User}, \mathcal{T}_c, k)$.

For *verification*, a user presents $(\texttt{User}, \mathcal{R})$ to the server. The reading $\mathcal{R}$ and key k are used, by the server, to calculate a cancelable biometric $\mathcal{R}_c$. The server compares $\mathcal{R}_c$ and $\mathcal{T}_c$ and, if close, the user is accepted.

4.1 Matching after the transformation

Let $\mathcal{X}$ be the set of all possible images, and let the matching algorithm M act on any two images x_1, x_2 to give a matching score $M(x_1, x_2)$. Let $\mathcal{C}$ be a set of keyed transformations, $\mathcal{C} : \mathcal{X} \to \mathcal{X}_\mathcal{C}$. Let C_k denote the particular transformation associated with $k \in \mathcal{K}$, where $\mathcal{K}$ is the key set, and the matching algorithm M' act on any two transformed images $C_k(x_1), C_k(x_2)$ to give a matching score $M'(C_k(x_1), C_k(x_2))$.

We consider compatibility of M and M' with the transformation.

Definition 1 *We say a transform $\mathcal{C} : \mathcal{X} \to \mathcal{X}_{\mathcal{C}}$ is ϵ–match preserving, for the pair (M, M') of matching algorithms, if, for any pair of images $x_1, x_2 \in \mathcal{X}$ and for every $k \in \mathcal{K}$, $|M'(C_k(x_1), C_k(x_2)) - M(x_1, x_2)| \leq \epsilon$.*

In practice only a subset of images and keys would be considered.

The 'effective' number of keys is an indication of the level of security of system. Two different keys may produce similar matching results and so one key can replace the other.

Consider two keys k_1 and k_2. For an image x let the transformed values under keys k_1 and k_2 be $C_{k_1}(x)$ and $C_{k_2}(x)$.

Definition 2 *We say k_1 and k_2 are κ–distinct if, for any not necessarily distinct image pair x, x',*

$$|M'(C_{k_2}(x), C_{k_1}(x') - M'(C_{k_1}(x), C_{k_1}(x'))| \geq \kappa \qquad .$$

If κ is small, false acceptance can result. That is, a new reading can be accepted under an incorrect key. We identify a subset of keys that can be reliably distinguished. The size of this set must be sufficiently large to provide the required security.

5 A key based transformation for fingerprints

Here we define a key dependent transformation of the minutiae. The key $0 \leq \phi \leq \pi$ is an angle specifying a line $ax + b$ through the core point (o_x, o_y). Our transformation is illustrated in Figure 1 and specified as;

`Transformation`	
Input: $\phi, (o_x, o_y), (x_i, y_i)$	
$p_i = o_x - x_i, q_i = o_y - y_i.$	
If $p_i \tan(\phi) > q_i$	
$\omega = 2(\theta - \tan^{-1}(q_i/p_i))$	
$p'_i \longleftarrow \cos(\omega)p_i - \sin(\omega)q_i$	$q'_i \longleftarrow \sin(\omega)p_i + \cos(\omega)q_i$
$x_i \longleftarrow p'_i + o_x$	$y_i \longleftarrow q'_i + o_y.$
Output (x_i, y_i)	

This algorithm means minutiae under the line specified by the key are reflected to above the line, while those above are unchanged.

As noted in Section 4 the local feature vectors are calculated using the new minutiae positions and the ridge structure from the original reading. Without reference to the ridge structure the distribution of minutiae

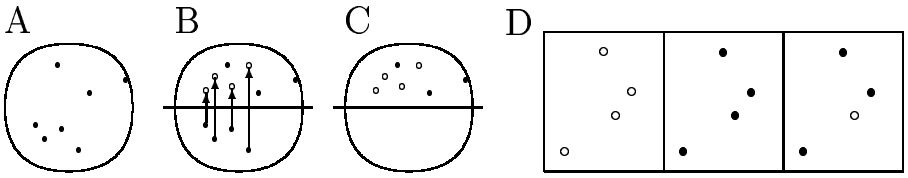


Fig. 1. Diagrams A–C respectively show the minutiae as in the fingerprint reading, the transformation with the hollow circles being the new positions, and the positions after the transformation. We refer to the region all the minutia lie in at the end as the "upper region". The three possible neighbourhood structures in the upper region are shown in Diagram D; all–reflected, none reflected and a mixture.

is effectively the same as fingerprints. There are, however, some relative minutiae positions that are at best unlikely. In particular, one of two very close minutiae in the transformed space is likely to be have been transformed, so the resulting cancelable biometric leaks information about the original fingerprint.

5.1 The effect of the transformation on matching

In general applying transform to templates requires a new matching algorithm be used for the transformed data. Here we argue that for our proposed transform, it is plausible to use the same matching algorithm.

Local structures were discussed in Section 3.2 and are dependent upon the local feature vectors of each minutiae. The ridge structure used to generate the local feature vectors for the cancelable template $\mathcal{T}_c$ belongs to the "upper region" of the image. As such, the local structure vectors are unlikely to be similar to those generated prior to the transformation.

The local structure matching algorithm has two purposes. It identifies a pair of minutiae, one from $\mathcal{T}$, the other from $\mathcal{R}$, to put into correspondence. Global matching takes place relative to this pair. The algorithm also calculates local matching scores $s_l(i,j)$ used in global matching.

Consider two similar readings, $\mathcal{R}$ and $\mathcal{R}'$. Following Definition 1 we want to consider if the variances tolerated between $\mathcal{R}$ and $\mathcal{R}'$ are still tolerable between the transformed $\mathcal{R}_C$ and $\mathcal{R}'_C$, that is, is the tolerance preserved. We consider the effect on the different parameters, recalling that the local structure depends on Euclidean distance and direction angle between minutiae, relative ridge angle, ridge distance and type.

Tolerable variances in position, and thus Euclidean distance, are still tolerable after the transform. In general it is likely the neighbourhoods will change, however it is still possible there is a primary preserved closest matching neighbourhood. Local matching only needs one neighbourhood. However the neighbourhoods are based on ridge distance, which changes

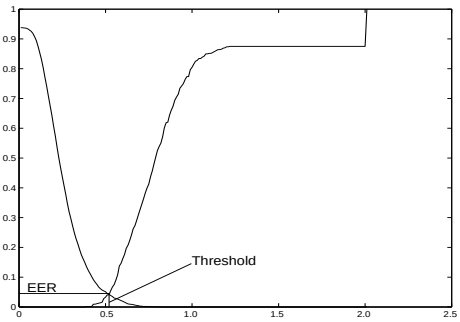
substantially in the transform. The ridge structures are all in the upper region, so tolerable variations on the original ridge structure cannot easily guarantee tolerable variations on the final ridge structure. In general there will be little correlation between the ridge structure in the original location and that in the final location. Thus ridge distance and angles could (and do) cause significant problems in tolerance preservation under the transform. While the ridge structure is problematic, the tolerance in Euclidean structures suggests the transformation is worth investigating.

6 Results and analysis

We used a batch of 80 fingerprints from [9]. Those images consist of 8 images each of 10 fingerprints. NIST Fingerprint Image Software [4] was used to identify the `R92` core point and extract the minutiae.

We implemented the transformation algorithm to transform minutiae positions. We also implemented the matching algorithm [6], which returns a similarity level, between 0 and 2, for two feature sets.

We considered matching the fingerprints before the transforms. The false rejection rates and false acceptance rates are shown here for $L = 5$. The axes are percentage of cases (the y–axis) and matching score. For pre–transformation we have an approximate Equal Error Rate (EER) of 4%, occuring at a threshold of about 0.52. To the right we tabulate $L = 2 \ldots 6$ results.



L	Equal error rate	Threshold
2	2%	0.53
3	2.5%	0.53
4	3.5%	0.51
5	4%	0.52
6	$31 - 50\%$	$0.03 - 0.035$

Having analysed the pre–transformation matching, we applied the transforms for twenty keys, spaced by $\pi/10$, to each original template, and consider matching on the results. There are three types of results.

1: We calculate the FRR, FAR and EER rates post–transform. We illustrate a typical $L = 5$ example in Figure 2A. There is a significant increase in FRR and FAR. In particular, errors in aligning transformed templates reduces matching accuracy. EER post–transform was found to be $\approx 16.8\%$, at a threshold of ≈ 0.54. While the thresholds are similar, which is good, the EER is significantly higher. This result supports the use of the transform since other systems [3] have rates as high as 30%.

2: Following Definition 2 we compare matching between images transformed under different keys. For a useful algorithm the probability of matching an image transformed under different keys should be low. In Figure 2B we illustrate how the matching score between transforms under different keys, of the same typical image, are still high. For example, at a threshold of about 0.52, about 70% of the images were still accepted, even though only 5% should be. This very high error rate is a significant problem, at best meaning a small key space.

3: We examine how the transformation and matching algorithms perform under the measure in Definition 1. Though small variance in FFR and FAR curves is good, the reality is shown in the shifts table to the right. The variance in matching score is rather large. Since the definition of ϵ–invariance requires ϵ to cover all transformations, ϵ is too large to be of use. The proportion of such large shifts is also high .

L	Average	Maximum
2	0.147	0.935
3	0.148	0.746
4	0.152	0.849
5	0.161	0.877
6	0.076	1.955

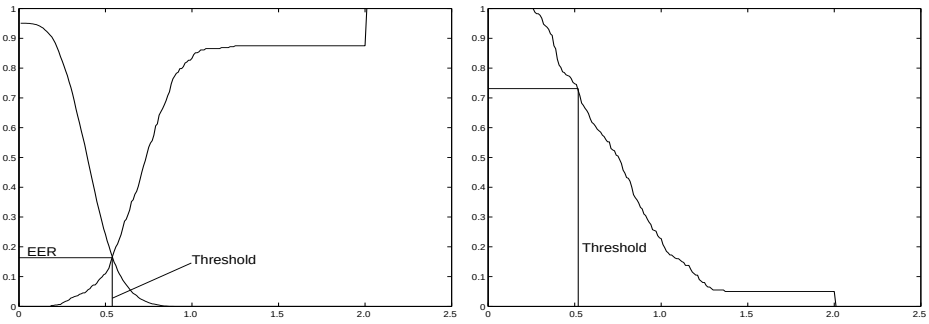


Fig. 2. On the left we show a typical $L = 5$ post–transform result. The graphs show the proportion of cases where false results are accepted (the higher curve at the beginning) and where true results are rejected. The axes are percentage of cases (the y–axis) and matching score. The EER occurs at a matching score of about 0.54. The proportion of errors at the threshold is about 17%. On the right we compare we compare the matching scores of images resulting from transformations under different keys, for a typical fingerprint image. The graph shows the proportion of cases having a matching score of at least that on the bottom axis. We see, for example, about 70% of the images are matched with a threshold of 0.52 suggested by the pre–transformation results.

Results for $L = 2$ to $L = 5$ appear similar. For $L = 6$ the correlation required means either the image is very close and can match, or is not very close and won't match, even if by eye the images are similar. This suggests using neighbourhoods of at most five in this context.

7 Conclusion

The use of biometrics in user authentication systems is very promising. However, without adequate security considerations, the compromise of such biometrics may result in them being useless for the user forever. We consider the cancelable biometrics of Ratha *et al* [11] and describe some measures of how good a cancelable fingerprint system, primarily consisting of a key–based transformation and a matching algorithm, is.

We consider a specific transformation and matching algorithm, and use the measures to suggest the pair is not useful in it's current form. We consider that good key–based template generators and matching algorithms probably need to be developed together, rather than explicitly using existing matching algorithms.

References

1. J. Araque, M. Baena, and P. Vizcaya 'Synthesis of fingerprint images', *Proceedings of the 16th International Conference on Pattern Recognition*, **II** (2002) 422–425.
2. R. M. Bolle, A. W. Senior, N. K. Ratha and S. Pankanti 'Fingerprint minutiae: A constructive definition.' *Biometric Authentication* (2002) 58–62.
3. T. Clancy, D. Lin and N. Kiyavash, 'Secure smartcard–based fingerprint Authentication', *ACM Workshop on Biometric Methods and Applications*, (2003) 45–52.
4. M. Garris, C. Watson, R. McCabe and C. Wilson 'Users guide to NIST fingerprint image software.' NIST (2001).
5. C. Hill 'Risk of masquerade arising from the storage of biometrics'. Honours thesis, ANU, (2001). `http://chris.fornax.net/download/thesis/thesis.pdf`
6. X. Jiang and W. Yau 'Fingerprint Minutiae Matching Based on the Local And Global Structures' *Proceedings of the 15th International Conference on Pattern Recognition* **II** (2000) 6038–6041.
7. A. Juels and M. Sudan 'A fuzzy vault scheme', *Proceedings of the 2002 IEEE International Symposium on Information Theory* (2002) 408–.
8. A. Juels and M. Wattenberg 'A fuzzy commitment scheme.' *ACM Conference on Computer and Communications Security* (1999) 28–36.
9. D. Maltoni, D. Maio, A. Jain, and S. Prabhakar **Handbook of Fingerprint Recognition**, (Springer–Verlag, New York, 2003).
10. L. O'Gorman 'Practical systems for personal fingerprint authentication.' *IEEE Computer* **33**(2) (2000) 58–60.
11. N. K. Ratha, J. H. Connell and R.M. Bolle, 'Enhancing security and privacy in biometrics–based authentication systems' *IBM Systems Journal* **40**(3) (2001) 614–634.
12. VeriFinger. Neurotechnologija Ltd. `http://www.neurotechnologija.com`.
13. C. Vielhauer, R. Steinmetz and A. Mayerhofer, 'Biometric Hash based on Statistical Features of Online Signatures' *Proceedings of the 16th International Conference on Pattern Recognition* **I** (2002) 123–126.