Generating Cancelable Fingerprint Template Using Distortion Transformation

Haiyong Chen
Key Laboratory for Information System Security, Ministry of Education of China
haiyongc@gmail.com

Hailiang Chen
Krannert School of Management, Purdue University, USA
chenhl@purdue.edu

Kwok-Yan Lam
Key Laboratory for Information System Security, Ministry of Education of China

Abstract

In a fingerprint verification system, templates are stored in the server database. To avoid the private issues in case that the database is compromised, researchers propose that only those distorted templates are stored, and they are matched also in the distorted form. If a distorted template is compromised, it can be "cancelled" by choosing just another distortion transform. In this paper, we present a method of transforming fingerprint minutiae distortedly and performing matching in the distorted form. In our method, a circular region is constructed around each minutia and region information is represented by all minutiae within its region. Transform is applied to all regions and only transformed regions are stored in the database as a distorted template. Experimental results show the performance of matching using transformed data.

Keywords: *Cancelled biometrics; Privacy; Fingerprint; Distortion transformation*

1. Introduction

Biometrics provides an irrefutable proof of personal identity and offers more security and convenience than traditional methods of personal recognition. However, users still have several concerns: Do different applications use the same biometric template? If the biometric template in an application is compromised, is it compromised forever? Will the biometric data be abused for an unintended purpose? Will the undeniable proof of biometrics-based access be used to track the individuals that may infringe upon an individual's right to privacy? Such privacy issues should be solved.

Cancelled biometrics is a privacy-protection technology that can help solve the above problems. Instead of storing the original biometric data in the system database during enrollment, the system stores only its noninvertible distorted version (eg., a hash). During recognition, the system would transform the data using the same noninvertible transform and perform matching in the transformed space. Different applications can use different noninvertible distorted versions, so a template would be usable only by the application that created it. If a distorted template is compromised, it can be "cancelled" by choosing just another distortion transform. The user can choose the transform parameters in terms of a password or PIN. If such a biometric template is compromised, the system can generate a new one using a different transform or different parameters.

In this paper, we propose a method of transforming fingerprint minutiae distortedly and performing fingerprint matching in the transformed space. The paper is outlined as follows: section 1 gives a brief introduction, section 2 introduces related work, section 3 describes our method in steps, section 4 shows experimental results, section 5 gives our conclusions.

2. Related Work

The core idea of cancelled biometrics is that a distortion transform is applied to a biometric template and only the transformed template is transmitted and stored. Related introduction can be found in [3],[4],[5],[6]. Nalini K. Ratha et al.[3] present several alternatives, such as Cartesian, polar, and functional transformation, and compare their relative merits empirically. Their experimental analysis shows that the surface folding transform achieves better results than the other two transforms, namely, Cartesian and polar. The polar transformation performs better than the

Cartesian transformation in terms of error rates while the polar transform is less strong. In our method, we use circular regions of minutiae to increase the strength, ensuring both the accuracy and security.

Fuzzy schemes share some similarities with the technology of cancelled biometrics. A cryptographic construction called fuzzy vault is proposed in [7] [8]. Clancy et al. [9] propose a fingerprint vault using multiple minutiae location sets. They create the maximum number of chaff points to construct R that locks k. U. Uludag et al. [10] also propose a fuzzy fingerprint vault that uses minutiae lines to lock a secret. Fuzzy extractor scheme[11][12] proposes how to generate strong keys from noisy data. In fact, templates of the above methods are also transformed forms .

Biometric Encryption is another related technology. It is a process that securely binds a PIN or a cryptographic key to a biometric, so that neither the key nor the biometric can be retrieved from the stored template. Fen Hao et al.[13] propose a practical and secure way to integrate the iris biometric into cryptographic applications. F. Monrose et al.[14] proposes a new approach to strengthening the security of user chosen passwords. Their techniques make use of habitual patterns in a user's typing rhythm for generating strong cryptographic keys that could be used, for example, for file encryption, VPN access, etc. Their extended work of generating cryptographic keys from voice is introduced in [15][16][17].

3. The Proposed Method

In this section, we will introduce the method of distortedly transforming fingerprint minutiae and performing matching in the transformed form. The transform is application dependent, meaning that the templates cannot be reused by another applications. The user can provide the transform parameters in terms of a password or PIN. Figure 3.1 shows that a user generates different distortion forms using different PINs.

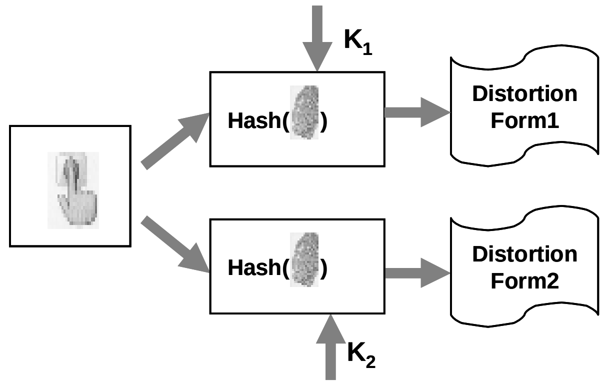
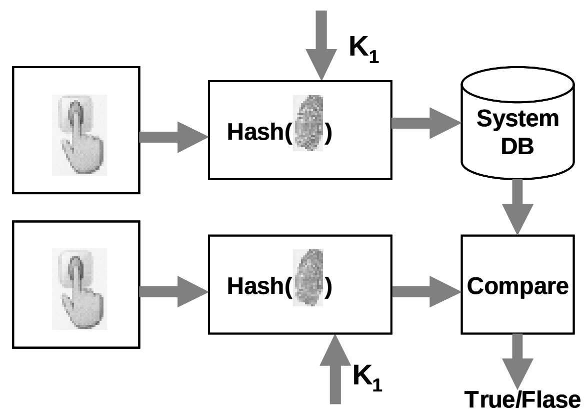


Figure 3.1 application dependent transform

During enrollment, the system transforms original fingerprint data using the parameters provided by the user, then stores the distortion form in the server database. During recognition, the system transforms a sample of the user’s fingerprint using parameters provided by the user, then performs matching with the templates in the transformed form. Figure 3.2 shows the stages of enrollment and recognition.



Enrollment and recognition using the distortion form

In our method, we use circular regions which are built around each minutia. The set of hashed regions is stored as the distortion form. The method can be presented in the following steps.

3.1 Construct circular regions

we construct a circular region around each minutia. Each region is identified by the centered minutia, and has the same radius. The number of circular regions we construct equals that of minutia in a fingerprint. Region information can be represented by the set of minutiae in its region. Different circular regions have different quantity of minutiae due to the random distribution of minutiae. And minutiae near the border of fingerprint may have a partial circular region.

3.2 Hash circular regions

In this step, we introduce how to hash a circular region.

Let A_i be a minutia represented as (x, y, α) in Cartesian coordinate system, where (x, y) is the position and α is the orientation. Let $R(A_0)$ be a region that A_0 is the centered minutia. Suppose $R(A_0)$ has k minutiae, then $R(A_0)$ can be represented as $\{A_0, A_1, \ldots, A_{k-1}\}$.

Convert all minutiae of $R(A_0)$ from Cartesian to polar system. The minutiae positions are measured in polar coordinates with reference to the core position. The angles are measured with reference to the orientation of the core. Then A_i is represented as (ρ, e, α)

Rotate all minutiae of $R(A_0)$ at one time to ensure the orientation of A_0 equals 90. Then minutia A_i is converted to (ρ, e_r, α_r).

Hash all minutiae of $R(A_0)$. Assume a_1, a_2, b_1, b_2, c_1, c_2 are transform parameters provided by the user, and v_1, v_2 are hashed values of a minutia.

$v_1=a_1*\rho+b_1*e_r+c_1*\alpha_r$

$v_2=a_2*\rho+b_2*e_r+c_2*\alpha_r$

Given v_1, v_2, a_1, a_2, b_1, b_2, c_1, c_2, it is impossible to restore $(\rho, e_r, \alpha r)$ for the number of equations is less than that of variables.

On the other hand, we classify minutiae of $R(A_0)$ into several levels according to the range of ρ. The larger ρ is, the higher the level of minutia is. Figure 3.3 shows levels of minutiae in a region. In Figure 3.3, there are 5 levels: B_1, B_2, B_3, B_4, B_5. Two minutiae belong to B_1, four minutiae belong to B_2, two minutiae belong to B_3, six minutiae belong to B_4, four minutiae belong to B_5.

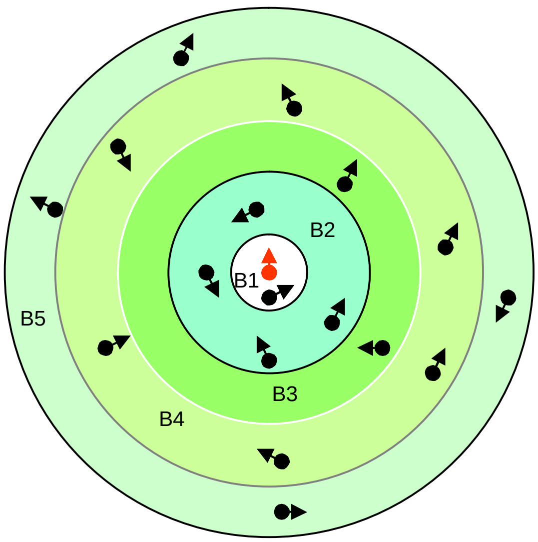


Figure 3.3 levels of minutiae in a circular region

With the additional attribute of level, a minutia of a hashed region can be represented as $\{v_1, v_2, \text{level}\}$. The algorithm of hashing a circular region is described as follows:

Algorithm:

Hash a circular region

Input:

$R(A_0)=\{A_0, A_1, \ldots, A_{k-1}\}$

$a_1, a_2, b_1, b_2, c_1, c_2$

Output: $HR(A_0)=\{h(A_0), h(A_1), \ldots, h(A_{k-1})\}$

Declaration:

$R(A_0)$ –the circular region around A_0

$HR(A_0)$ –the result of hashing $R(A_0)$

$a_1, a_2, b_1, b_2, c_1, c_2$–transform parameters

A_i– a minutia in the region

$h(A_i)$-the result of hash A_i

$\rho(A_i)$, $e_r(A_i)$, $\alpha_r(A_i)$ – ρ, e_r, α_r of minutia A_i

$v_1(A_i)$, $v_2(A_i)$, level (A_i) – v_1, v_2, level of minutia A_i

Method:

Convert minutiae of $R(A_0)$ from Cartesian to polar system;

Rotate all minutiae of $R(A_0)$ at one time to ensure the orientation of A_0 equals 90;

for(i=0; i<k; i++)

{

$v_1(A_i)=a_1*\rho(A_i)+b_1*e_r(A_i)+c_1*\alpha_r(A_i)$;

$v_2(A_i)=a_2*\rho(A_i)+b_2*e_r(A_i)+c_2*\alpha_r(A_i)$;

$h(A_i)=\{v_1(A_i), v_2(A_i), \text{level}(A_i)\}$;

}

return $\{h(A_0), h(A_1), \ldots, h(A_{k-1})\}$;

Using the above algorithm, we can hash all circular regions, and store the hashed regions as the transformed template.

3.3 Matching using hashed regions

Some matching algorithms using original minutiae are introduced in [18], [19], [20], [21]. Here we introduce our matching method using hashed data. Suppose a template has N minutiae-$\{T_1, T_2, \ldots, T_N\}$, a sample has M minutiae -$\{S_1, S_2, \ldots, S_M\}$, then the template has N hashed regions- $\{HR(T_1), HR(T_2), \ldots, HR(T_N)\}$, and the sample has M hashed regions -$\{HR(S_1), HR(S_2), \ldots, HR(S_M)\}$. The matching algorithm is described as follows:

Algorithm:

Matching using hashed regions

Input:

$\{HR(T_1), HR(T_2), \ldots, HR(T_N)\}$
$\{HR(S_1), HR(S_2), \ldots, HR(S_M)\}$

Output: True or False

Declaration:

HashedRegionMatch(HR(T_i),HR(S_j)) –a function of matching HR(T_i) against HR(S_j), returns the number of matched hashed minutiae

Method:

```
for(i=1; i<=N; i++)
  for(j=1; j<=M; j++)
  {
    if(HashedRegionMatch(HR(Ti),HR(Sj))>=threshold(region_match_num))
      return True;
  }
return False
```

In HashedRegionMatch(HR(U),HR(W)), assume $h(U_p)$ is a hashed minutia in HR(U), $h(W_q)$ is a hashed minutia in HR(W), $h(U_p)$ and $h(W_q)$ are matched only if the following inequations are satisfied at the same time:

$$\begin{cases} |v_1(U_p) - v_1(W_q)| < threshold(v_1) \\ |v_2(U_p) - v_2(W_q)| < threshold(v_2) \\ |level(U_p) - level(W_q)| = 0 \end{cases}$$

4. Experimental Results

Images of FVC2002 DB1B and DB2B were used in our experiments. Each database consists of 80 images (10 distinct fingers, 8 instances each). In our experiments, we first extract minutiae from fingerprint images, then transform fingerprint minutiae distortedly using our method with different values of parameters, which includes a_1, a_2, b_1, b_2, c_1, c_2 and the number of levels. The parameters of a_1, a_2, b_1, b_2, c_1, c_2 are working like a PIN, required to be presented by the user during enrollment and recognition. Using such parameters, a user can generate different distorted templates and cancel any template just by changing parameters.

Let num(levels) be the number of levels. We choose 3 values for *num(levels)*: 3, 5, and 8. For each value of *num(levels)*, we randomly generate 10 groups of a_1, a_2, b_1, b_2, c_1, c_2 . Thus, for each fingerprint database, we perform a total of 8400 genuine matches(For any group of parameters-*num(levels)*, a_1, a_2, b_1, b_2, c_1, c_2, we perform 280 genuine matches) and 10800 impostor matches (For any group of parameters-*num(levels)*, a_1, a_2, b_1, b_2, c_1, c_2 , we perform 360 imposter matches). Our experiments were run on a PC with Pentium IV 1.8GHz CPU and 512M RAM. Table 1, 2 show the accuracy, speed, template size of our method against FVC2002 DB1B, DB2B respectively. Figure 4.1, 4.2 show ROC curves against FVC2002 DB1B, DB2B respectively.

Table 1 Performance of matching in the transformed form (DB1B)

Region radius	100		
Total of matches	19200		
num(levels)	8	5	3
EER	6.0%	7.1%	8.7%
Avg match time	0.03s		
Avg template size	15.5kb		

Table 2 Performance of matching in the transformed form (DB2B)

Region radius	100		
Total of matches	19200		
The number of levels	8	5	3
EER	3.6%	4.9%	5.5%
Avg match time	0.04s		
Avg template size	23kb		

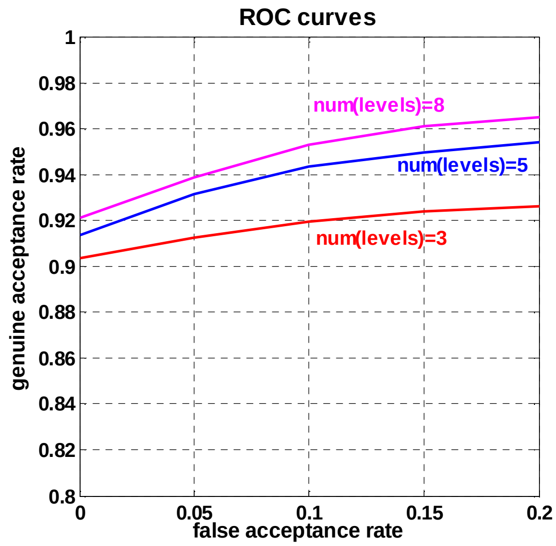


Figure 4.1 A comparison of ROC curves (DB1B)

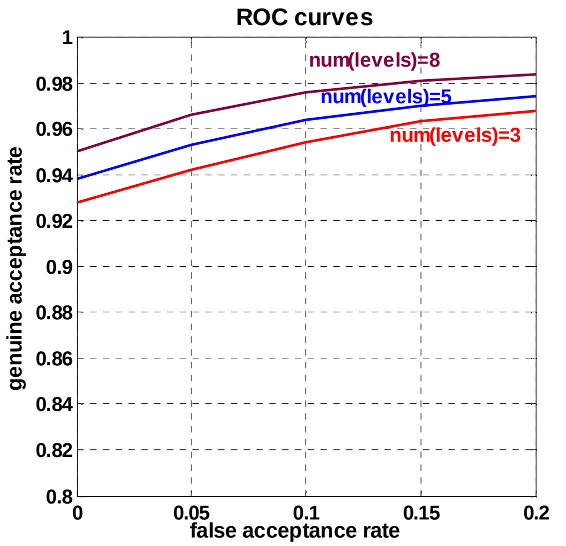


Figure 4.2 A comparison of ROC curves (DB2B)

5. Conclusions

To solve the problems of privacy and security in case the fingerprint database is compromised, this paper proposes a method of transforming fingerprint minutiae and performing matching in the transformed form. Hashed versions of circular regions are stored as "cancelled template" and different applications can use different distorted forms. The noninvertible transform is based on a simple rule: when the number of linear equations is less than that of variables, it is impossible to compute the values of variables. Our experiments show the accuracy and processing speed using distorted templates. However, this method also has a security-accuracy problem. When the number of "levels " increases, our method has a good accuracy rate, but the transform is less strong against brute-force attacks. Therefore, in the future work, we will focus updating the method of transforming and matching, making the matching less dependent on the " levels" of minutiae.

References

[1] U Uludag, S. Pankanti, S. Prabhakar and A. K. Jain, "Biometric cryptosystems: issues and challenges," Proceedings of the IEEE, Vol. 92, No. 6, pp. 948–960, 2004.
[2] A.K. Jain, A. Ross, S. Pankanti, "Biometrics: a tool for information security," Information Forensics and Security, IEEE Transactions on , vol.1, no.2pp. 125- 143, June 2006
[3]Nalini K. Ratha, Sharat Chikkerur, Jonathan H. Connell, Ruud M. Bolle, "Generating Cancelable Fingerprint Templates," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 4, pp. 561-572, Apr., 2007
[4] N. Ratha, J. Connell, R. Bolle, and S. Chikkerur, "Cancelable Biometrics: A Case Study in Fingerprints," *Proc. Int'l Conf. Pattern Recognition,* 2006.
[5] R. Ang, R. Safavi-Naini, and L. McAven, "Cancelable Key-Based Fingerprint Templates," *Proc. 10th Australian Conf. Information Security and Privacy,* pp. 242-252, July 2005.
[6] M. Savvides, B.V.K. Vijaya Kumar, and P.K. Khosla, "Cancelable Biometric Filters for Face Recognition," *Proc. Int'l Conf. Pattern Recognition,* pp. 922-925, 2004.
[7] A. Juels and M. Wattenberg, "A fuzzy commitment scheme," Proceeding of the 6th ACM Conference on Computer and Communication Security, CCCS, 1999
[8] A. Juels and M. Sudan, "A fuzzy vault scheme," Proceedings of IEEE International Symposium on Information Theory, 2002.
[9] T. C. Clancy, N. Kiyavash, and D. J. Lin, "Secure Smartcard-Based Fingerprint Authentication", ACM SIGMM 2003 Multimedia, Biometrics Methods and Applications Workshop, pp. 45-52, 2003.
[10] U. Uludag, S. Pankanti, and A. K. Jain, "Fuzzy vault for fingerprints," in Proc. AVBPA, Lecture Notes in Computer Science 3546, pp. 310--319, Springer, 2005.
[11]Y. Dodis, L. Reyzin, and A. Smith, "Fuzzy Extractors: How to Generate Strong Keys from Biometrics and Other Noisy Data," Proc. Int'l Conf. Theory and Applications of Cryptographic Techniques, pp. 523-540, May 2004.
[12]X. Boyen, "Reusable cryptographic fuzzy extractors," CCS 2004, pp. 82–91, ACM Press.
[13] Feng Hao; R. Anderson, J. Daugman, "Combining Crypto with Biometrics Effectively," Computers, IEEE Transactions on , vol.55, no.9pp. 1081- 1088, Sept. 2006
[14] F. Monrose, M.K. Reiter, and R. Wetzel, "Password hardening based on keystroke dynamics," Proceedings of sixth ACM Conference on Computer and Communications Security, CCCS 1999. pp. 73 – 82.
[15] F. Monrose, M.K. Reiter, Q. (Peter) Li , and S. Wetzel. Cryptographic Key Generation from Voice (Extended Abstract).In Proceedings of the 2001 IEEE Symposium on Security and Privacy, May 2001. 12 pages.
[16] F. Monrose, M.K. Reiter, Q. (Peter) Li , and S.Wetzel. Using Voice to Generate Cryptographic Keys. In 2001: A Speaker Odyssey. The Speech Recognition Workshop, Crete, Greece, June, 2001. Six pages.
[17] F. Monrose, M. K. Reiter, Q. Li, D. P. Lopresti, and C. Shih, "Toward speech-generated cryptographic keys on resource constrained devices," in Proc. 11th USENIX Security Symp.,2002, pp. 283 – 296.
[18] Tsai-Yang Jea, Venu Govindaraju: A minutia-based partial fingerprint recognition system. Pattern Recognition 38(10): 1672-1684 (2005)
[19] Dongjae Lee; Kyoungtaek Choi; Jaihie KIm, "A robust fingerprint matching algorithm using local alignment,"

Pattern Recognition, 2002. Proceedings. 16th International Conference on , vol.3, no.pp. 803- 806 vol.3, 2002
[20] Xudong Jiang; Wei-Yun Yau, "Fingerprint minutiae matching based on the local and global structures," Pattern Recognition, 2000. Proceedings. 15th International Conference on , vol.2, no.pp.1038-1041 vol.2, 2000
[21] Jain, A.; Lin Hong; Bolle, R., "On-line fingerprint verification," Pattern Analysis and Machine Intelligence, IEEE Transactions on , vol.19, no.4pp.302-314, Apr 1997