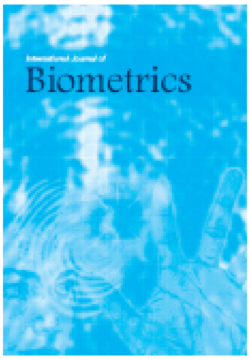


International Journal of Biometrics

ISSN online: 1755-831X - ISSN print: 1755-8301
https://www.inderscience.com/ijbm

Performance enhancement of symmetric hashed fingerprint template using dynamic threshold matching algorithm

S. Ajish, K.S. AnilKumar

DOI: 10.1504/IJBM.2022.10043185

Article History:

Received:	21 January 2021
Last revised:	16 July 2021
Accepted:	26 October 2021
Published online:	15 December 2022

Performance enhancement of symmetric hashed fingerprint template using dynamic threshold matching algorithm

S. Ajish*

Department of Futures Studies,
Research Centre,
University of Kerala,
Karyavattom, Thiruvananthapuram,
Kerala, India
Email: ajishs2014@gmail.com
*Corresponding author

K.S. AnilKumar

University of Kerala,
Thiruvananthapuram, Kerala, India
Email: ksanilksitm@gmail.com

Abstract: The fingerprint template protection is strenuous among other biometric template protections because the fingerprint template stores the minutiae points as security enhancement of the fingerprint template and results in the degradation of the matching performance. The modified symmetric hash method uses a secret key as a multiplication parameter for the hashing of the fingerprint biometric template. The irreversibility and unlikability analysis of the modified symmetric hashed fingerprint template exhibits better security. The multiplication of the fingerprint minutiae template by a secret key mitigates the accuracy of matching performance. This paper proposes a dynamic threshold matching algorithm in which the threshold values are derived from the secret key. Experimental results on FVC 2004 database show that the combination of modified symmetric hash function and the dynamic threshold matching algorithm prompt better security and excellent matching performance.

Keywords: fingerprint template; modified symmetric hashing; accuracy; irreversibility; unlinkability; minutiae matching; false match rate; false nonmatch rate; Euclidean distance; angular threshold.

Reference to this paper should be made as follows: Ajish, S. and AnilKumar, K.S. (2023) 'Performance enhancement of symmetric hashed fingerprint template using dynamic threshold matching algorithm', *Int. J. Biometrics*, Vol. 15, No. 1, pp.78–100.

Biographical notes: S. Ajish is a research scholar in the Department of Future Studies, University of Kerala. His research interests include information security and cryptography. He is author of a great deal of research studies published at national and international journals and conference proceedings.

K.S. AnilKumar received his PhD in Technology Management at the University of Kerala. Currently, he held the position of registrar at the University of Kerala. His research interests are related to e-governance, information security and cryptography. He has published research papers at national and international journals and conference proceedings.

1 Introduction

Due to the worldwide use of fingerprint biometrics for authentication and forensic identification, fingerprint biometric protection attains the particular interest of the research community. The main properties that should be satisfied by the protected templates (Inuma, 2014; Gomez-Barrero et al., 2016; Jain et al., 2008b) are

1 non-reversibility

2 unlinkability

3 accuracy

4 revocability.

The two classifications of template protection are

1 template or feature transformation

2 biometric cryptosystem (Ferrara et al., 2012; Nagar et al., 2010).

The feature transformation method transforms the biometric template into a protected template based on the password or keys given as an external parameter. At the time of authentication, the same feature transformation is performed on the query template, and it is compared against the protected template saved in the database.

The feature transformation is further classified into *salting* and *non-invertible* transforms (Ferrara et al., 2012). Salting or biohashing (Jin et al., 2004) applies a hash function based on a random seed (key) to generate the biocode. Salting transformations are invertible; if the challenger gets the key, the original biometric template can be easily regenerated from the protected biometric template by the attacker. Hence the security of salting relies on the key or password. The non-invertible transformation (Ratha et al., 2001, 2007; Lee et al., 2007) applies a trapdoor function which is hard to invert the transformed template to the original template.

Biometric cryptosystem (Uludag et al., 2004; Cavoukian et al., 2008) is used to secure the cryptographic keys or create cryptographic keys from biometric features. The biometric cryptosystem uses details of the biometric template called the helper data in the biometric database. The biometric cryptosystem is categorised into *key generation* and *key binding*. The key generation (Dodis et al., 2004, 2008) approach generates the cryptographic keys with the help of helper data and query patterns. The key binding (Juels and Wattenberg, 1999; Juels and Sudan, 2006) approach uses the helper data extracted from the unprotected pattern and an external cryptographic key, when matching the key is recuperated from helper data.

Among the protection of biometric templates, the fingerprint template protection is strenuous because the fingerprint template is store as minutiae points (Wieclaw, 2009;

Belhadj, 2017; Maltoni, 2005; Murmu and Otti, 2009). Each minutiae point is represented as $m = \{x, y, \theta, \gamma\}$ where x and y denote the x and y axis, θ denotes the orientation angle, γ denotes the minutiae type. A small change in x, y, and θ value due to template protection degrades matching accuracy. It is strenuous to model template protection that is both secure in-connection with cryptography and accurate in-connection with biometric matching.

Tulyakov et al. (2007) describes a symmetric hash function $h_1(c_1, c_2, \ldots, c_n) = \frac{c_1 + c_2 + \ldots + c_n}{n}$ are the n minutiae points. The experimental analysis of the symmetric hash function shows that there is degradation in the accuracy of matching. The symmetric hash function is modified (Ajish and Kumar, 2020) by multiplying the hashed output by a key value K in which $0 < K < 1$, to enhance the accuracy and non-reversibility. When the minutiae point is multiplied by a key value less than one, the parameters in the minutiae points ($m = \{x, y, \theta, \gamma\}$) becomes a smaller value. As a result, the number of minutiae points within the threshold value (Wieclaw, 2009; Belhadj, 2017; Maltoni, 2005; Murmu and Otti, 2009) increases, thereby an increase in true positive rate (TPR) (Bremananth and Chitra, 2006; Sabir, 2018) and false positive rate (FPR) (Bremananth and Chitra, 2006; Sabir, 2018). The increase in FPR (Bremananth and Chitra, 2006; Sabir, 2018) degrades the matching performance.

This paper proposes a dynamic threshold-based minutia matching algorithm to enhance the performance of the modified symmetric hashed fingerprint template (Ajish and Kumar, 2020). The multiplication of the hashed template with a key value increases the non-reversibility of the hashed template, but it degrades the accuracy of matching. In the dynamic threshold matching algorithm, the Euclidean distance threshold (Wieclaw, 2009; Belhadj, 2017; Maltoni, 2005; Murmu and Otti, 2009) and the angular difference threshold (Wieclaw, 2009; Belhadj, 2017; Maltoni, 2005; Murmu and Otti, 2009) is adjusted according to the key value used as a multiplication parameter. The combination of modified symmetric hashing and dynamic threshold matching algorithms enhances the irreversibility and unlinkability of the protected template and the accuracy of matching.

The remaining part of the paper is organised as follows. Section 2 gives an overview of the minutiae-based fingerprint template matching algorithm and the symmetric hash function for fingerprint template security. Section 3 describes the dynamic threshold matching algorithm. Section 4 analyses the performance of the dynamic threshold matching algorithm, and Section 5 analyses the security of the modified hashed fingerprint template. Section 6 concludes the paper.

2 Research background

2.1 Minutiae-based fingerprint template matching algorithm

Part 2 of ISO/IEC 19794 standards (ISO/IEC 19794-2:2005, 2005) (Belhadj, 2017) represents the minutiae points and fingerprint matching based on minutiae points. The ISO/IEC 19794 standard represents the minutiae point as $m = \{x, y, \theta, \gamma\}$, where x and y denote the x and y axis, θ indicate the minutiae angle, and γ represent the minutiae type. The minutiae-based matching algorithm calculates the similarity score between the template minutiae $T = \{m_1^t, m_2^t, m_3^t, \ldots, m_i^t\}$ and the query minutiae $Q = \{m_1^q, m_2^q, m_3^q, \ldots, m_j^q\}$

(Wieclaw, 2009). The Euclidean distance (Belhadj, 2017; Maltoni, 2005) between two minutiae points m_i^t and m_j^q is calculated using equation (1).

$$ed\left(m_i^t, m_j^q\right) = \sqrt{\left(x_i^t - x_j^q\right)^2 + \left(y_i^t - y_j^q\right)^2} \tag{1}$$

The angular difference (Belhadj, 2017; Maltoni, 2005) between two minutiae points m_i^t and m_j^q is calculated using equation (2).

$$dd\left(m_i^t, m_j^q\right) = \min\left(\left|\theta_i^t - \theta_j^q\right|, 360 - \left|\theta_i^t - \theta_j^q\right|\right) \tag{2}$$

If the Euclidean distance $ed(m_i^t, m_j^q)$ between two minutiae points is less than a threshold distance th_{ed} and the angular difference $dd(m_i^t, m_j^q)$ between two minutiae points is less than the threshold angular difference th_θ then the two minutiae points are considered as matching minutiae.

The fingerprint should be aligned to increase the number of matching minutiae, for that a *map*(.) function (Belhadj, 2017; Maltoni, 2005) has used, that map the minutiae m_j^q into $\tilde{m}_j^q$ using the geometrical translation [Δ*x*, Δ*y*] and rotation [θ_r]. $map_{\Delta x, \Delta y, \theta_r}$ $(m_j^q = \{x_j^q, y_j^q, \theta_j^q\}) = \tilde{m}_j^q = \{\tilde{x}_j^q, \tilde{y}_j^q, \tilde{\theta}_j^q\}$ where

$$\begin{bmatrix} \tilde{x}_j^q \\ \tilde{y}_j^q \end{bmatrix} = \begin{bmatrix} \cos(\theta_r) & -\sin(\theta_r) \\ \sin(\theta_r) & \cos(\theta_r) \end{bmatrix} \begin{bmatrix} x_j^q \\ y_j^q \end{bmatrix} + \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix}$$

The match function *mm*(.) (Wieclaw, 2009; Belhadj, 2017; Maltoni, 2005) returns 1 when the minutiae $\tilde{m}_j^q$ matches with the minutiae m_i^t.

$$mm\left(\tilde{m}_j^q, m_i^t\right) = \begin{cases} 1 & \text{if } ed\left(m_i^t, m_j^q\right) \le th_{ed} \text{ and } dd\left(m_i^t, m_j^q\right) \le th_\theta \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

The match score between the input template *T* and the query template *Q* is calculated using the equation

$$\underset{\Delta x, \Delta y, \theta_r, P}{maximise} \sum_{i=1}^{m} mm\left(map_{\Delta x, \Delta y, \theta_r}\left(m_{jP(i)}^q, m_i^t\right)\right) \tag{4}$$

where *P*(*i*) (Maltoni, 2005) is a pairing function used to decide whether there is matching minutiae in *Q* (query template) and *T* (reference template).

2.2 Symmetric hash function for securing fingerprint template

Tulyakov et al. (2007) proposes a hash function and matching algorithm by considering the accidental shift (Kumar et al., 2010) ($f(z) = rz + t$), where *z* represents the minutiae point m_i, *r* represent the scalar rotation parameter and *t* represents the translation parameter of the accidental shift) during the registration and authentication scans. The author proposes a symmetric hash function, which is independent of the order of input.

Consider the n minutiae points $\{c_1, c_2, \ldots, c_n\}$ then the m symmetric hash functions described by Tulyakov et al. (2007) are as follows.

$$\begin{aligned} h_1(c_1, c_2, \ldots, c_n) &= \frac{c_1 + c_2 + \ldots + c_n}{n} \\ h_2(c_1, c_2, \ldots, c_n) &= \frac{c_1^2 + c_2^2 + \ldots + c_n^n}{n^2} \\ &\ldots \\ h_m(c_1, c_2, \ldots, c_n) &= \frac{c_1^m + c_2^m + \ldots + c_n^m}{n^m} \end{aligned} \tag{5}$$

The m and n in the above equations represents the number of hash functions and the number of minutiae points as input receptively, if $m < n$ then it is impractical to generate the original minutia from the hashed template.

Algorithm 1 Modified hash algorithm for fingerprint minutiae

```
Result: Hashed output
Input: Fingerprint minutiae
Output: Hashed fingerprint minutiae
Read the fingerprint minutiae M in matrix form. Find n = number of rows in the matrix.
while All minutiae m_i of M is processed do
    read the x and y coordinate of m_i from M
    read all other minutiae m_j from M
    calculate $sd(m_i, m_j) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$
    find the minimum value of sd and represent the minutiae as m_j;
    Calculate $H(m_i, m_j) = \left(\frac{m_i + m_j}{2}\right) * secret - key$
end
```

3 Fingerprint minutiae matching algorithm with dynamic threshold values

The symmetric hash function is modified (Ajish and Kumar, 2020) to enhance the non-reversibility property of the hashed fingerprint template. The security of the hash functions depends on the hashing type and the total minutiae point given as input. The symmetric hash function does not use any secret value; it is modified by using a secret key as a multiplication parameter. When the hashed output is multiplied by a secret key, it enhances the security and non-reversibility of the hashed template. The modified symmetric hash algorithm is described in Algorithm 1.

The minutiae point m_i^t (Lee et al., 2007; Uludag et al., 2004) in T is considered to match m_j^q in Q if the Euclidean distance and angular difference between them should be less than the threshold distance th_{ed} and threshold angular difference th_θ respectively. That is

$$ed\left(m_i^t, m_j^q\right) = \sqrt{\left(x_i^t - x_j^q\right)^2 + \left(y_i^t - y_j^q\right)^2} \leq th_{ed} \tag{6}$$

and

$$dd\left(m_i^t, m_j^q\right) = \min\left(\left|\theta_i^t - \theta_j^q\right|, 360 - \left|\theta_i^t - \theta_j^q\right|\right) \leq th_\theta \tag{7}$$

where *ed*(;) represents the Euclidean distance between the minutiae points and *dd*(;) describes the angular difference between the two minutiae points (Lee et al., 2007; Uludag et al., 2004; Cavoukian et al., 2008), where the Euclidean distance threshold the_d and the angular difference threshold th_θ are constant values.

The fingerprint minutiae templates are hashed using the symmetric hash function described in Algorithm 1. The multiplication of the minutiae points by a key value less than one decreases the (*x*, *y*) coordinate values and *θ* values. The decrease in (*x*, *y*) coordinate value and the *θ* values consequence more minutiae points fall inside the Euclidean threshold the_d and angular threshold th_θ as represented in Table 2.

Table 1 Euclidean distance *ed* and angular difference *dd* between fingerprint template 101-1 and 102-1

Sl no.	X_1	Y_1	Θ_1	X_2	Y_2	Θ_2	*ed*	*dd*
1	222	54	1	235	21	3	35.47	114.65
2	262	55	1	104	38	3	158.91	114.65
3	130	62	3	135	40	3	22.56	0.00
4	263	64	1	216	54	3	48.05	114.65
5	161	69	1	42	117	3	128.32	114.65
6	247	89	1	159	118	3	92.66	114.65
7	23	135	3	223	141	3	200.09	0.00
8	67	139	3	273	147	3	206.16	0.00
9	185	141	3	166	162	3	28.32	0.00
10	253	173	3	220	182	1	34.21	114.65
11	185	178	1	142	185	1	43.57	0.00
12	164	206	1	165	204	3	2.24	114.65
13	246	211	1	183	244	3	71.12	114.65
14	188	225	3	124	251	3	69.08	0.00
15	208	269	3	37	264	1	171.07	114.65
16	147	275	3	257	268	3	110.22	0.00
17	194	276	3	234	279	3	40.11	0.00
18	201	277	1	174	244	5	42.64	229.30
19	42	278	1	180	240	7	143.14	343.95
20	197	267	5	168	240	7	39.62	114.65
21	192	264	7	12	288	7	181.59	0.00
22	252	12	7					
23	180	264	7					

The modified symmetric hash consider each minutiae points (m_i) of the fingerprint template and find out the nearest minutiae point m_j. The nearest minutiae point is find-out to reduce the displacement between minutiae (m_i) and the hashed output minutiae. The minutiae is hashed by taking the average and multiply it with a secret key, i.e., $H(m_i, m_j) = \left(\frac{m_i + m_j}{2}\right) * key$, key, where 0 < *key* < 1. The minutiae values of the fingerprint 101-1 and 101-2 are represented in Table 1. The Euclidean distance *ed* and angular difference *dd* between fingerprint template 101-1 and 102-1 are represented in Table 1. The minutiae values of the hashed fingerprint template 101-1 and 101-2 with key 0.5 are represented in Table 2. The Euclidean distance *ed* and angular difference *dd* between the hashed fingerprint template (key value 0.5) 101-1 and 102-1 are represented in Table 2. The data from Tables 1 and 2 points out that when the minutiae points are hashed with a key value less than 0.75, Euclidean distance *ed* and angular difference *dd* values become small and more minutiae fall inside the threshold value.

Table 2 Euclidean distance *ed* and angular difference *dd* between hashed fingerprint template 101-1 and 102-1

Sl no.	X_1	Y_1	θ_1	X_2	Y_2	θ_2	*ed*	*dd*
1	12.1	2.725	0.05	11.275	1.875	0.15	1.18	5.73
2	13.125	2.975	0.05	5.975	1.95	0.15	7.22	5.73
3	7.275	3.275	0.1	5.975	1.95	0.15	1.86	2.87
4	13.125	2.975	0.05	11.275	1.875	0.15	2.15	5.73
5	7.275	3.275	0.1	3.65	3.875	0.15	3.67	2.87
6	12.75	3.825	0.05	8.125	7	0.15	5.61	5.73
7	2.25	6.85	0.15	11.075	8.075	0.1	8.91	2.87
8	2.25	6.85	0.15	12.4	7.2	0.15	10.16	0.00
9	9.25	7.975	0.1	7.7	8.675	0.1	1.70	0.00
10	12.475	9.6	0.1	11.075	8.075	0.1	2.07	0.00
11	8.725	9.6	0.05	7.675	9.725	0.1	1.06	2.87
12	8.8	10.775	0.1	7.675	9.725	0.1	1.54	0.00
13	12.475	9.6	0.1	9.075	12.1	0.25	4.22	8.60
14	8.8	10.775	0.1	7.3	12.275	0.25	2.12	8.60
15	10.225	13.65	0.1	1.225	13.8	0.2	9.00	5.73
16	8.175	13.475	0.25	12.275	13.675	0.15	4.10	5.73
17	9.875	13.825	0.1	12.275	13.675	0.15	2.40	2.87
18	9.875	13.825	0.1	8.85	12.1	0.3	2.01	11.46
19	4.725	13.825	0.1	9.075	12.1	0.25	4.68	8.60
20	9.725	13.275	0.3	8.55	12.1	0.3	1.66	0.00
21	9.725	13.275	0.3	1.225	13.8	0.2	8.52	5.73
22	12.85	1.675	0.2					
23	9.3	13.2	0.35					

The data reported in Table 7 states that hashing of fingerprint minutiae templates with key value 0.25 increases the EER to 33.12%, FMR100 to 88.58%, and ZeroFMR to

100%. The hashing of fingerprint template with key value 0.5 raises the EER to 18.91%, FMR100 to 88.58%, and ZeroFMR to 100%. The data analysis of Table 8 points out that hashing of fingerprint templates with a key value less than 0.75 degrades the matching performance.

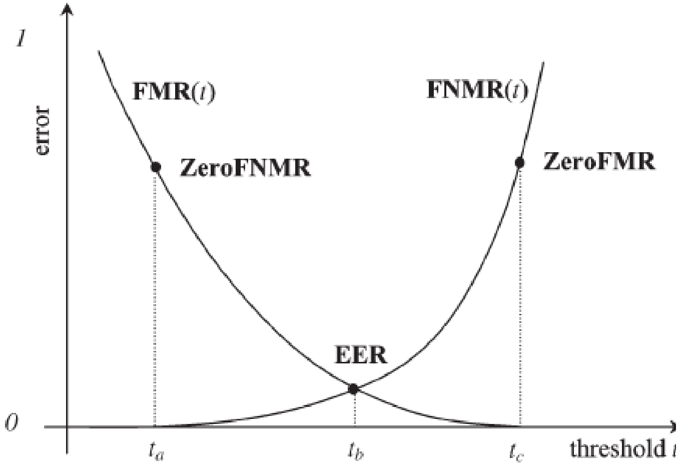
So the static Euclidean distance threshold the_d and the angular difference threshold th_θ are replaced by a dynamic threshold derived from the key value. When the key value is equal to 0.75, the matching performance is almost similar to unprotected fingerprint template performance.

3.1 Generation of dynamic euclidean distance threshold th_{ed} and the angular difference threshold th_θ

The trial and error experimental analysis shows that the best Euclidean distance threshold for a key value of 0.25 is 0.3 * the_d, and the angular difference threshold is 0.3 * th_θ. The key value of 0.75 indicates the reliable results with the Euclidean distance threshold value of 1.0 * the_d and angular difference threshold value of 1.0 * th_θ. The algorithm for generating the dynamic Euclidean distance threshold the_d and the angular difference threshold th_θ described in Algorithm 2. The different key values and the corresponding Euclidean distance threshold and angular difference threshold are reported in Table 3. The default Euclidean distance threshold is 15, and the angular difference threshold is 14. As indicated in Table 3, when the key value decreases from 0.75, the Euclidean threshold and angular threshold value decrease. The correctness of the algorithm is checked for a range of key values between 0 and 1.

When the key value is greater than 0.25, the variable diff is calculated using the equation '*diff* = (*key value* – 0.25) * 10'. The Euclidean Threshold is calculated using the equation '*Euclidean threshold* = (0.3 + (*diff* * 0.14)) * the_d', where the_d is the default Euclidean threshold value (15). The angular threshold is calculated using the equation '*angular threshold* = (0.3 + (*diff* * 0.14)) * th_θ', where th_θ is the default angular threshold value (14).

Figure 1 FMR/FNMR curve with EER, ZeroFMR and ZeroFNMR points highlighted



When the key value is less than 0.25, the variable diff is calculated using the equation '*diff* = (0.25 – *key value*) * 10'. The Euclidean threshold is calculated using the equation '*Euclidean threshold* = (0.3 – (*diff* * 0.12)) * th_{ed}' and the angular threshold is calculated using the equation '*angular threshold* = (0.3 – (*diff* * 0.12)) * th_θ'.

Algorithm 2 Algorithm to generate dynamic threshold value

```
Result: Dynamic Euclidean and angular threshold value
if key value ≥ 0.25 then
    $diff = (key\ value - 0.25) * 10$;
    $Euclidean\ threshold = (0.3 + (diff * 0.14)) * th_{ed}$;
    $Angular\ threshold = (0.3 + (diff * 0.14)) * th_\theta$;
else
    $diff = (0.25 - key\ value) * 10$;
    $Euclidean\ threshold = (0.3 - (diff * 0.12)) * th_{ed}$;
    $Angular\ threshold = (0.3 - (diff * 0.12)) * th_\theta$;
end
```

4 Result analysis

The FVC 2004 database is used to analyse the performance of unprotected fingerprint template, hashed fingerprint template, and hashed fingerprint template with dynamic threshold matching algorithm. The four different databases in the FVC 2004 are DB1, DB2, DB3, and DB4. Each DB1, DB2, DB3, and DB4 (set B) database consists of fingerprint images of ten different fingers and eight impressions of each finger.

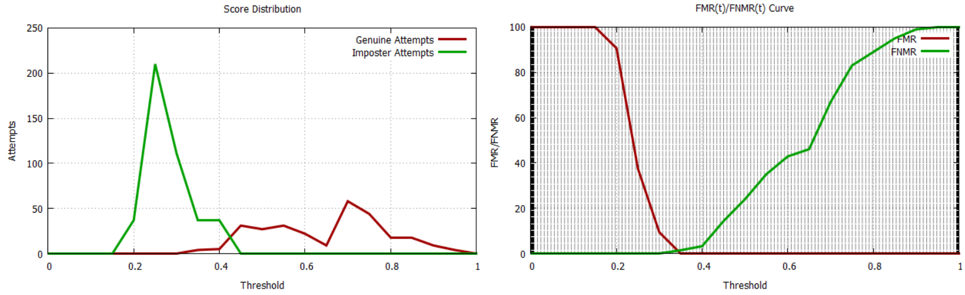
Table 3 Different key values and dynamic thresholds

Key	*0.1*	*0.25*	*0.5*	*0.75*
Euclidean distance	1.8	4.5	9.75	15
Angular difference	1.68	4.2	9.1	14

Table 4 Result on DB1-B of unprotected fingerprint template, hashed fingerprint template and dynamic hashed fingerprint template

Algorithm	*EER (%)*	*FMR100 (%)*	*FMR1000 (%)*	*ZeroFMR (%)*	*ZeroFNMR (%)*
Unprotected	2.24	2.43	2.56	2.64	12.41
Hashed key = 0.25	33.56	88.91	99.04	100	89.71
Hashed key = 0.5	18.89	68.22	76.78	79.87	84.78
Hashed key = 0.75	3.84	3.96	4.02	4.12	28.23
Dynamic hashed key = 0.25	3.58	3.87	3.93	4.09	29.52
Dynamic hashed key = 0.5	1.76	1.91	1.98	2.12	27.14
Dynamic hashed key = 0.75	3.84	3.96	4.02	4.12	28.23

Figure 2 Score distribution and FMR(t)/FNMR(t) curve of unprotected fingerprint template, (a) score distribution of unprotected fingerprint template (b) FMR(t)/FNMR(t) curve of unprotected fingerprint template (see online version for colours)



(a) (b)

Each fingerprint image in the database is represented as F_{ij}, where $I = 1, 2, \ldots, 10$ and $j = 1, 2, \ldots, 8$, and the corresponding fingerprint minutiae template is represented as T_{ij}.

Table 5 Result on DB2-B of unprotected fingerprint template, hashed fingerprint template and dynamic hashed fingerprint template

Algorithm	*EER (%)*	*FMR100 (%)*	*FMR1000 (%)*	*ZeroFMR (%)*	*ZeroFNMR (%)*
Unprotected	3.61	3.21	3.59	3.64	12.71
Hashed key = 0.25	35.22	89.31	99.12	100	88.47
Hashed key = 0.5	20.14	63.91	72.23	75.71	82.89
Hashed key = 0.75	4.01	4.44	4.76	4.88	29.23
Dynamic hashed key = 0.25	3.91	4.01	4.15	4.96	29.56
Dynamic hashed key = 0.5	2.11	2.23	2.31	2.47	27.11
Dynamic hashed key = 0.75	4.01	4.44	4.76	4.88	29.23

Table 6 Result on DB3-B of unprotected fingerprint template, hashed fingerprint template and dynamic hashed fingerprint template

Algorithm	*EER (%)*	*FMR100 (%)*	*FMR1000 (%)*	*ZeroFMR (%)*	*ZeroFNMR (%)*
Unprotected	1.54	1.97	3.02	3.21	10.89
Hashed key = 0.25	32.03	88.75	99.01	100	87.59
Hashed key = 0.5	18.65	65.46	70.74	73.21	81.27
Hashed key = 0.75	3.51	32.38	42.01	42.86	26.56
Dynamic hashed key = 0.25	3.69	4.92	5.31	5.36	10.89
Dynamic hashed key = 0.5	1.75	1.76	1.77	1.78	35.95
Dynamic hashed key = 0.75	3.51	32.38	42.01	42.86	26.56

For each database and each algorithm:

- The fingerprint templates T_{ij} are generated from the fingerprint images F_{ij} and it is stored on disk. The fingerprint template generation algorithm may **Fail(F)**, **Timeout(T)**, or **Crash(C)** during the template generation stage. The **Fail(F)**,

Timeout(**T**) five seconds, and **Crash**(**C**) rejections are summed to obtain the **REJ**$_{ENROLL}$.

- Each fingerprint images F_{ik} ($j < k \leq 8$) is matched with the fingerprint template T_{ij} to output and store the genuine match score **gms**$_{ijk}$. If **REJ**$_{ENROLL}$ = 0, the number of genuine recognition attempts (*NGRA*) is ((8 * 7) / 2) * 10 = 280.
- Each fingerprint images F_{1j}, j = 1, …, 10, is matched with the fingerprint templates of different fingerprint T_{ij} to output and store the imposter match score **ims**$_{ijk}$. If **REJ**$_{ENROLL}$ = 0, the number of imposter recognition attempts (*NIRA*) is ((10 * 79) / 2) = 395.

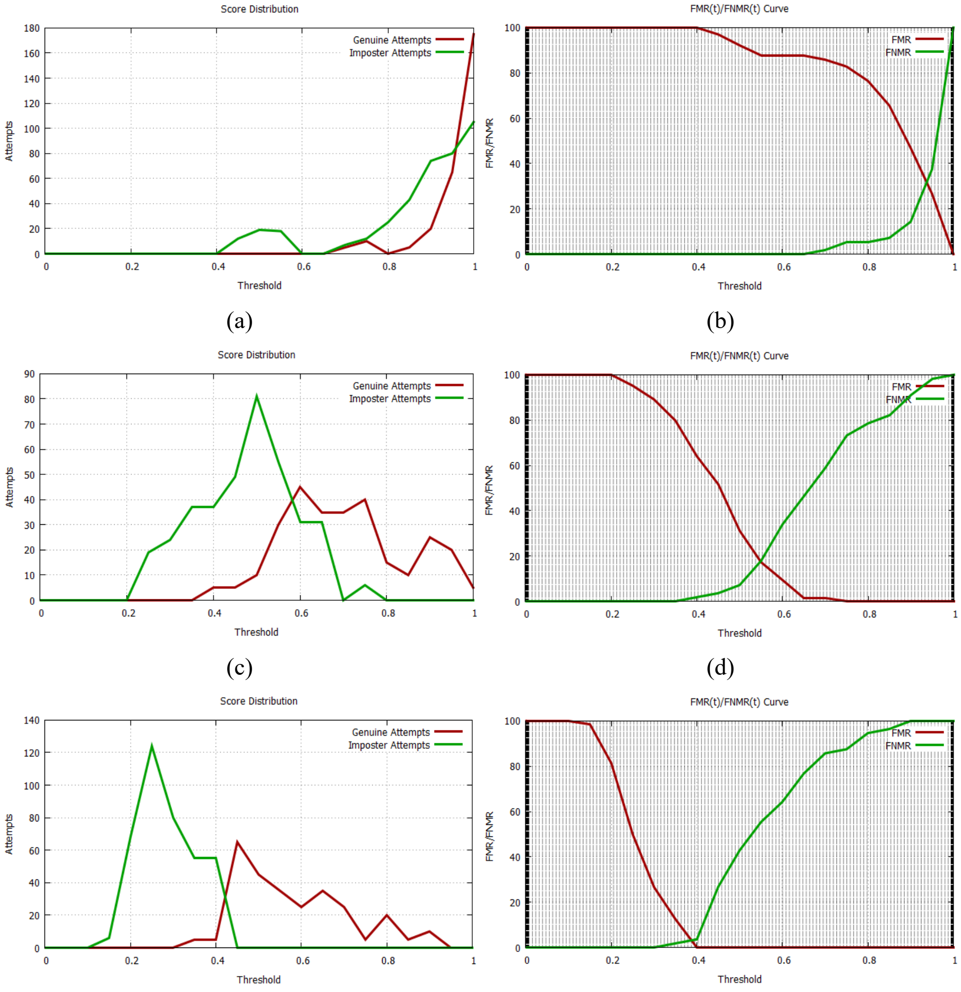
Table 7 Result on DB4-B of unprotected fingerprint template, hashed fingerprint template and dynamic hashed fingerprint template

Algorithm	*EER (%)*	*FMR100 (%)*	*FMR1000 (%)*	*ZeroFMR (%)*	*ZeroFNMR (%)*
Unprotected	1.26	1.33	1.41	1.43	9.37
Hashed key = 0.25	31.67	87.35	98.74	100	87.57
Hashed key = 0.5	17.97	61.59	71.86	73.71	79.75
Hashed key = 0.75	3.16	3.44	3.55	3.57	26.58
Dynamic hashed key = 0.25	3.15	3.41	3.52	3.53	26.51
Dynamic hashed key = 0.5	1.68	1.72	1.76	1.79	21.77
Dynamic hashed key = 0.75	3.16	3.44	3.55	3.57	26.58

Table 8 State of art comparison of different methods – average result of the four data base (DB1-B, DB2-B, DB3-B and DB4-B)

Algorithm	*EER (%)*	*FMR100 (%)*	*FMR1000 (%)*	*ZeroFMR (%)*	*ZeroFNMR (%)*
Unprotected	2.16	2.24	2.65	2.73	11.35
Hashed key = 0.25	33.12	88.58	98.98	100	88.34
Hashed key = 0.5	18.91	64.8	72.9	75.63	82.17
Hashed key = 0.75	3.63	11.06	13.59	13.86	27.65
Dynamic hashed key = 0.25	3.58	4.06	4.24	4.5	24.14
Dynamic hashed key = 0.5	1.83	1.91	1.96	2.04	27.99
Dynamic hashed key = 0.75	3.63	11.06	13.59	13.86	27.65
P101 (Cappelli et al., 2005)	2.07	2.54	4.70	6.21	-
P047 (Cappelli et al., 2005)	2.10	2.96	4.61	6.59	-
P071 (Cappelli et al., 2005)	2.30	2.73	5.10	10.01	-
Jain et al. (2008a)	2.90	7.03	18.24	34.98	38.47
Cappelli et al. (2010) (local)	4.91	9.43	18.28	36.15	97.68
Medina-Perez et al. (2014)	3.46	8.10	20.79	31.41	39.66
Fu et al. (2013)	3.42	9.84	26.06	52.39	42.74
Khanyile et al. (2014)	3.31	7.75	19.50	29.75	36.77

Figure 3 Score distribution and FMR(t)/FNMR(t) curve of hashed fingerprint template (DB4-B), (a) score distribution of hashed fingerprint template (key = 0.25) (b) FMR(t)/FNMR(t) curve of hashed fingerprint template (key = 0.25) (c) score distribution of hashed fingerprint template (key = 0.5) (d) FMR(t)/FNMR(t) curve of hashed fingerprint template (key = 0.5) (e) score distribution of hashed fingerprint template (key = 0.75) (f) FMR(t)/FNMR(t) curve of hashed fingerprint template (key = 0.75) (see online version for colours)



(a) (b)

(c) (d)

(e) (f)

- The genuine and imposter match scores distributions are calculated, and it is graphically represented to identify the separation between the two classes.
- The false match rate (**FMR**(**t**)) and false non-match rate (**FNMR**(**t**)) for a threshold range 0 to 1 is calculated using the below equations.

$$\mathbf{FMR}(\mathbf{t}) = \frac{card\left\{\mathbf{ims}_{ijk} \middle| \mathbf{ims}_{ijk} \geq t\right\}}{\mathbf{NIRA}} \tag{8}$$

$$\mathbf{FNMR(t)} = \frac{card\left\{\mathbf{gms}_{ijk} \middle| \mathbf{gms}_{ijk} < t\right\} + \mathbf{REJ}_{NGRA}}{\mathbf{NGRA}} \tag{9}$$

The **FMR(t)** and **FNMR(t)** curve are plotted for the threshold range 0 to 1.

- The equal error rate (**EER**) is measured from the **FMR(t)**/**FNMR(t)** curve, the **EER** is the point in the **FMR(t)**/**FNMR(t)** curve where **FMR(t)** = **FNMR(t)** as shown in Figure 1.
- The ROC receiving operating curve is plotted with **FMR(t)** as a function of **FNMR(t)**, the curve is plotted in log-log scale.
- **ZeroFMR** is the lowest FNMR at which **FMR** = 0, and **ZeroFNMR** is the lowest FMR at which **FNMR** = 0.

$$\mathbf{ZeroFMR(t)} = \min_t \left\{\mathbf{FNMR(t)} \middle| \mathbf{FMR(t)} = 0\right\} \tag{10}$$

$$\mathbf{ZeroFNMR(t)} = \min_t \left\{\mathbf{FMR(t)} \middle| \mathbf{FNMR(t)} = 0\right\} \tag{11}$$

- **FMR100** is the lowest FNMR for **FMR** ≤ 1% and **FMR1000** is the lowest FNMR for **FMR** ≤ 0.1%

4.1 Performance evaluation

The EER, FMR100, FMR1000, ZeroFMR, ZeroFNMR of the FVC 2004 (DB1-B, DB2-B, DB3-B and DB4-B) are listed in Tables 4, 5, 6, and 7, respectively. The score distribution and FMR(t)/FNMR(t) curve (DB4-B) of the unprotected fingerprint template, hashed fingerprint template and hashed fingerprint template with dynamic threshold matching are shown in Figures 2, 3, and 4, respectively. The state of art comparison of the average value of EER, FMR100, FMR1000, ZeroFMR, ZeroFNMR of different methods are listed in Table 8. The following observations are made:

1 The lowest average value of EER obtained for the hashed fingerprint template with dynamic threshold matching with key value 0.5, and the EER is 1.83%. The hashing of the fingerprint template with a key value of less than 0.75 increases the EER to 33.2%. The use of a dynamic threshold matching algorithm reduces the EER to 3.58%.

2 For the hashed fingerprint template with dynamic threshold matching with key value 0.5, the FMR100 = 1.91% and FMR1000 = 1.96%, it is the lowest value. The fingerprint template hashing with a key value less than 0.75 increases the FMR100 and FMR1000.

3 The average ZeroFMR obtained for the hashed fingerprint template with dynamic threshold matching with a key value of 0.5 is 2.04%, which is the lowest value. The lowest average value of ZeroFNMR is obtained for the unprotected fingerprint template, and the ZeroFNMR is 11.35%.

4 The ROC curve of unprotected, hashed, and hashed with a dynamic threshold matching algorithm is shown in Figure 5. The ROC curve of the hashed fingerprint

template with a dynamic threshold matching algorithm plot nearer to the x-axis denotes better performance.

The performance analysis results conclude that the best fingerprint matching algorithm is the hashed fingerprint template with a key value of 0.5 and a dynamic threshold matching algorithm. The dynamic threshold matching algorithm overcomes the matching performance degradation due to the fingerprint template hashing.

5 Security analysis

The ISO/IEC IS 24745 on the biometric information protection (ISO/IEC, 2013) represents the irreversibility and unlinkability property of biometric templates. The irreversibility property means that it should be computationally hard to reverse the protected template to the original template. Unlinkability property measures the difficulty in determining whether two templates originate from the same biometric instances or not.

5.1 Unlinkability

Gomez-Barrero et al. (2016) proposes two measures: $D_{\leftrightarrow}(s)$ and $D_{\leftrightarrow}^{sys}$, where $D_{\leftrightarrow}^{sys} \in [0, 1]$, used to measure the unlinkability of the system and $D_{\leftrightarrow}(s) \in [0, 1]$ used to measure the unlinkability of a specific score. When the value of $D_{\leftrightarrow}^{sys} = 1$ which means the system is fully linkable and when the value of $D_{\leftrightarrow}^{sys} = 0$ which means the system is fully unlinkable. Any value of $D_{\leftrightarrow}^{sys}$ between 0 and 1 indicates that the system is semi linkable. The increase in $D_{\leftrightarrow}(s)$ from 0 to 1 indicates the decrease in the degree of unlinkability (Gomez-Barrero et al., 2017).

The success of linkability depends on determining whether two templates originate from the mated samples (H_m) (Gomez-Barrero et al., 2016, 2017) or non-mated samples (H_{nm}): $p(s|H_m) > p(s|H_{nm})$. The likelihood ration (Gomez-Barrero et al., 2017) LR(s) defined as

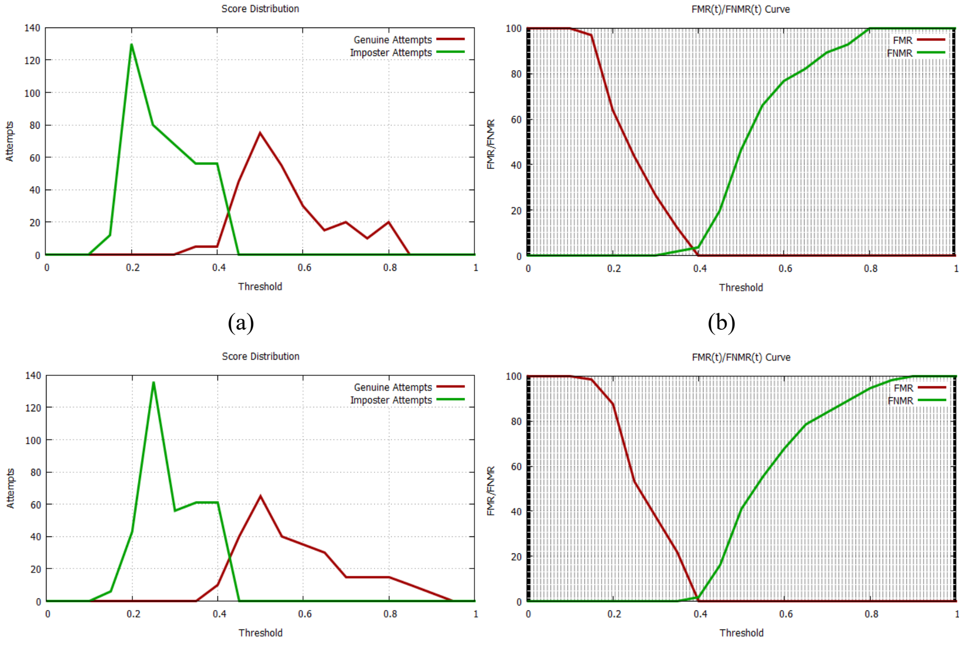
$$LR(s) = p\left(s\middle|H_m\right)/p\left(s\middle|H_{nm}\right) \tag{12}$$

The two particular cases that can be defined based on LR(s) are

- If $LR(s) \leq 1$, it can be interpreted that the two templates be a member of non-mated instances, so the templates are unlinkable to that particular score s. Therefore the value of $D_{\leftrightarrow}(s) = 0$.
- If $LR(s) > 1$, it can be interpreted that the two templates be a member of mated instances, so the templates are a little bit linkable to that particular score s. Therefore the value of $D_{\leftrightarrow}(s)$ will be closer to 1.

$D_{\leftrightarrow}(s)$ is defined as an expression of s and $LR(s)$ value. The value of $LR(s)$ is in the span $[0, \infty)$, a two-step normalisation method is proposed by Gomez-Barrero et al. (2016) to transform the value of $D_{\leftrightarrow}(s)$ in the desired range [0, 1]. The first step normalises $LR(s)$-1 to the reach [0.5, 1] with a sigmoid function. The second step subtract 0.5 and multiplied it by 2, and map to the range [0, 1].

Figure 4 Score distribution and FMR(t)/FNMR(t) curve of hashed fingerprint template with dynamic threshold matching algorithm, (a) score distribution of dynamic hashed fingerprint template (key = 0.25) (b) FMR(t)/FNMR(t) curve of hashed fingerprint template (key = 0.25) (c) score distribution of dynamic hashed fingerprint template (key = 0.5) (d) FMR(t)/FNMR(t) curve of dynamic hashed fingerprint template (key = 0.55) (see online version for colours)



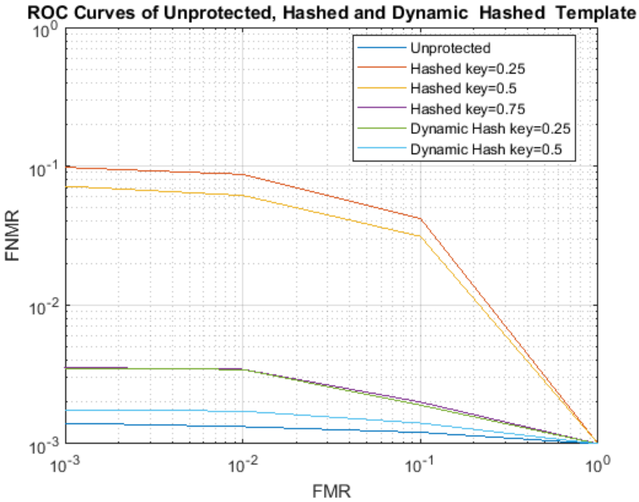
(a)

(b)

(c)

(d)

Figure 5 ROC curve of unprotected, hashed and dynamic hashed fingerprint template (see online version for colours)



Therefore $D_{\leftrightarrow}(s)$ is defined as

$$D_{\leftrightarrow}(s) = \begin{cases} 0 & \text{if } LR(s) \le 1 \\ 2\left(\left(1+e^{-(1+LR(s))}\right)^{-1} - 0.5\right) & \text{if } LR(s) > 2 \end{cases} \tag{13}$$

By the definition of sigmoid function,

$$\left(1+e^{-(1+LR(s))}\right)^{-1} \to 0.5 \text{ when } LR(s) \to 1 \tag{14}$$

$$\left(1+e^{-(1+LR(s))}\right)^{-1} \to 1 \text{ when } LR(s) \to \infty \tag{15}$$

The second step of normalisation shifts the value of $D_{\leftrightarrow}(s)$ from [0.5, 1] to [0, 1].

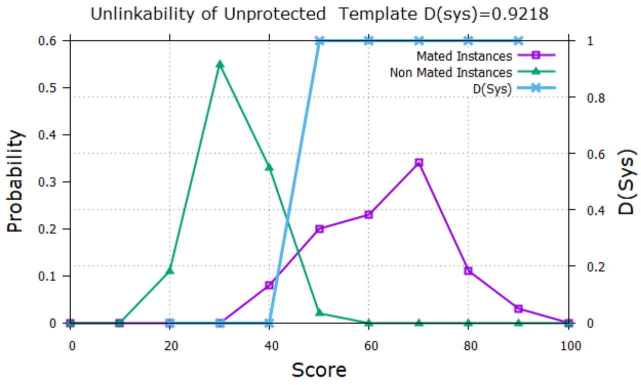
The useful estimation in the case of unlinkability is the unlinkability of the whole system. The unlinkability of the system $D_{\leftrightarrow}^{sys}$ defined as the partial area under the curve $D_{\leftrightarrow}(s)$, normalised by using the value $p(s|H_m)$ into the range [0, 1]. The unlinkability of the whole system in the score range $[s_{\min}, s_{\max}]$ is computed using the equation

$$D_{\leftrightarrow}^{sys} = \int_{s_{\min}}^{s_{\max}} D_{\leftrightarrow}(s) \cdot p\left(s \middle| H_m\right) ds. \tag{16}$$

5.1.1 Unlinkability of unprotected fingerprint template

The unlinkability analysis graph of the unprotected fingerprint template is shown in Figure 6. The unlinkability of the unprotected fingerprint system $D_{\leftrightarrow}^{sys}$ = 0.9218, which is closer to the upper bound value, means that the unprotected fingerprint template is almost fully linkable.

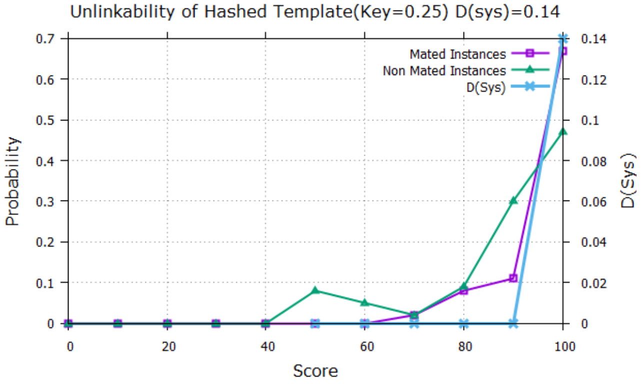
Figure 6 Unlinkability analysis of unprotected fingerprint template (see online version for colours)



5.1.2 Unlinkability of modified hashed fingerprint template with key value 0.25

The unlinkability analysis graph of the modified hashed fingerprint template with key value 0.25 is shown in Figure 7. The unlinkability of the modified hashed fingerprint template with key value 0.25 $D_{\leftrightarrow}^{sys}$ = 0.14, which is closer to the lower bound value, which means that the modified hashed fingerprint template with key value 0.25 is partially linkable.

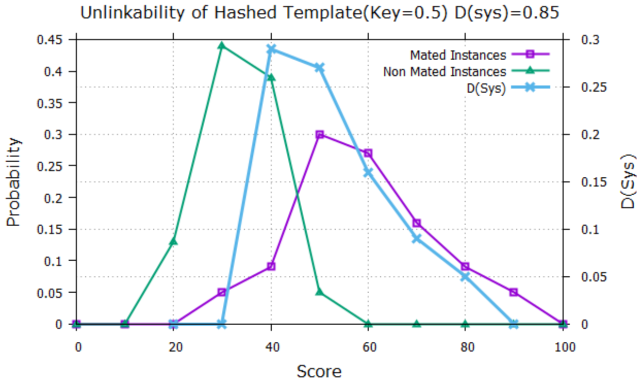
Figure 7 Unlinkability analysis of hashed fingerprint template with key value 0.25 (see online version for colours)



5.1.3 Unlinkability of modified hashed fingerprint template with key value 0.5

The unlinkability analysis graph of the hashed fingerprint template with key value 0.5 is shown in Figure 8. The unlinkability of the hashed fingerprint template with key value 0.5 $D_{\leftrightarrow}^{sys}$ = 0.85, which is closer to the upper bound value, means that the hashed fingerprint's linkability with key value 0.5 is almost fully linkable.

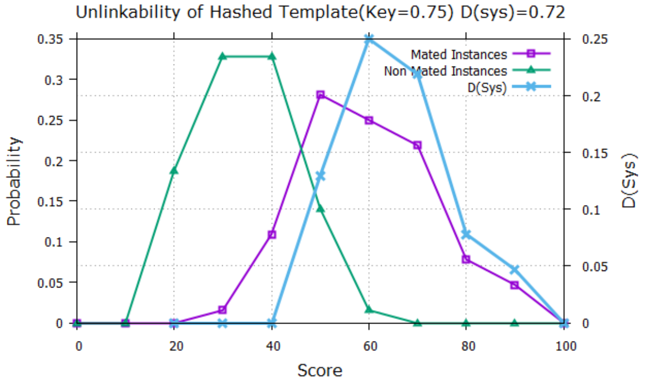
Figure 8 Unlinkability analysis of hashed fingerprint template with key value 0.5 (see online version for colours)



5.1.4 Unlinkability of hashed fingerprint template with key value 0.75

The unlinkability analysis graph of the hashed fingerprint template with key value 0.75 is shown in Figure 9. The unlinkability of the hashed fingerprint template with key value 0.75 $D_{\leftrightarrow}^{sys}$ = 0.72, which means that the linkability of the hashed fingerprint with key value 0.75 is partially linkable.

Figure 9 Unlinkability analysis of hashed fingerprint template with key value 0.75 (see online version for colours)



5.1.5 *Comparison of unlinkability of different methods*

The unlinkability analysis value of different methods $D_{\leftrightarrow}^{sys}$ are reported in Table 9. The unlinkability analysis of the hashed fingerprint template and hashed fingerprint template with dynamic threshold matching algorithm are same. The study of data in Table 9 states that when the fingerprint template is hashed with a key value of 0.25, the unlinkability $D_{\leftrightarrow}^{sys}$ of the system equal to 0.14, which is a lower bound value which means the system is almost unlinkable. The unlinkability $D_{\leftrightarrow}^{sys}$ of the unprotected fingerprint template is 0.92, which is an upper bound value which means the system is almost fully linkable.

Table 9 Comparison of $D_{\leftrightarrow}^{sys}$ of different methods

Method	$D_{\leftrightarrow}^{sys}$
Unprotected fingerprint template	0.9218
Hashed fingerprint template with key 0.25	0.14
Hashed fingerprint template with key 0.5	0.85
Hashed fingerprint template with key 0.75	0.72
Hashed fingerprint template with key 0.25 and using dynamic matching	0.14
Hashed fingerprint template with key 0.5 and using dynamic matching	0.85
Hashed fingerprint template with key 0.75 and using dynamic matching	0.72

5.2 *Irreversibility*

Irreversibility or non-invertibility is the difficulty in inverting the transformed or protected biometric template to the original template. The irreversibility measure estimates the probability of an attacker being able to determine the original template from the protected template. Nagar and Jain (2009) proposes a method to measure the non-reversibility of the protected fingerprint template. The proposed method builds a relationship between the attempts required by an attacker to regenerate the part of the original biometric template. The coverage effort (CE) curve indicates how much effort is required to regenerate the original minutiae from the transformed minutiae. The three steps in the measurement of CE curve are

1 pre-image computation

2 minutiae likelihood computation

3 non-invertibility measure computation.

5.2.1 Pre-image computation

To compute the pre-image the advisory first selects a minutiae point and expand the pre-image by selecting the neighbourhood minutiae points of the form (i, j), $(i + 1, j)$, $(i, j + 1)$, $(i + 1, j + 1)$. The pre-image set can also be expanded using the form (i, j), $(i - 1, j)$, $(i, j - 1)$, $(i - 1, j - 1)$. If two or more minutia points in the pre-image set are closer, only one minutiae among them is considered for inclusion in the pre-image set. The success of the reversibility attack depends on the accuracy of guessing the pre-image set. Complete link clustering (Jain and Dubes, 1988) with splitting criteria is used to improve the accuracy of the guessed pre-image. An eight-point 3D neighbourhood (Jochem et al., 2018) method is used to include θ in the pre-image set instead of the 2D method.

5.2.2 Pre-image likelihood computation

Consider a transformed minutiae v and the pre-images u^1, u^2, u^3, …, u^m of the transformed minutiae v. Any one value of $l_v \in (1, 2, 3, \ldots, m)$ indicates the pre-image of v which is the true one. The probability $P(l_v = r|v = a = (x_v, y_v, \theta_v)$ is calculated using Bayes theorem

$$P\left(l_v = r \middle| v = a\right) = \frac{p\left(v = a \middle| l_v = r\right) * p\left(l_v = r\right)}{\sum_{i=1}^{m} p\left(v = a \middle| l_v = r\right) * p\left(l_v = r\right)} \tag{17}$$

The probability of $P(l_v = i) = 1/m$; $\forall i = 1, 2, 3, \ldots, m$.

5.2.3 Non-invertibility measure computation

The non-invertibility is the total ciphering needed by an attacker to regenerate the actual minutiae set from the protected minutiae template. Suppose there are mi pre-images for the i^{th} minutiae, then the n-tuples the attacker needs to prioritise is very large. To reduce the complexity of the irreversibility analysis, the attacker reduces the size of the pre-image set by selecting only the most probable pre-image of each minutia. For each minutiae v_i; $i = 1, 2, 3, \ldots, n$ the attacker needs to check only the 2^{H_i} most probable pre-images. Where H_i is the entropy which is calculated using equation (18)

$$H_i = \sum_{i=1}^{m_i} P\left(l_{v_i} = r \middle| v_i\right) \log_2 P\left(l_{v_i} = r \middle| v_i\right) \tag{18}$$

where m_i is the total minutiae in the pre-image set. The total number of guesses for each minutia is $\pi_i 2^{H_i}$ and the total effort for reversing each minutia is $\frac{1}{n}\sum_i H_i$ bits per minutiae. The coverage is the fraction of minutiae identified among the total minutiae in the searched space.

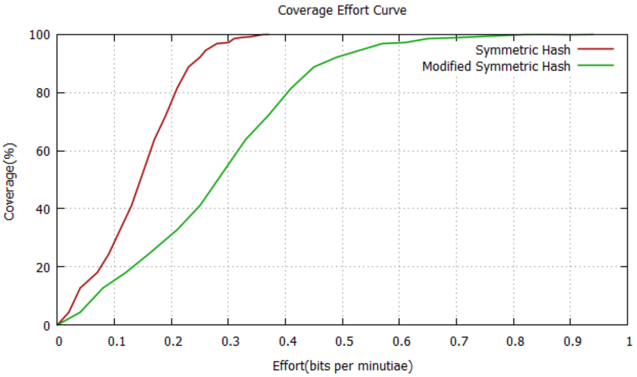
5.2.4 CE curve of the symmetric hashed minutiae

The first minutiae value of the unprotected fingerprint template and the hashed fingerprint template are (222, 54, 1) and (242, 54, 1), respectively. The two minutia value indicates that the hashing of the fingerprint template using the symmetric hash proposed by Tulyakov et al. (2007) does not displace the x and y location of the minutiae in the hashed template. So the attacker can easily select the minutiae's pre-image, and the total minutiae points in the pre-image (m) should be a small value. The minimum number of minutia points chosen for the formation of the pre-image set is 23. The CE curve of the symmetric hashed fingerprint template is shown in Figure 10. The effort required for 100% coverage of the hashed fingerprint template is 0.37.

5.2.5 CE curve of the modified symmetric hashed minutiae with key value 0.25

The first minutiae value of the unprotected fingerprint template and the modified hashed fingerprint template are (222, 54, 1) and (60.5, 13.625, 0.25), respectively. The hashing of the minutiae point by using the modified symmetric hash method results in an effective displacement of minutiae points in the x y plane and also in terms of the θ value. So the number of minutiae points in the pre-image set should be massive. The minimum number of minutiae points selected for pre-image is 300, and the CE graph of the modified symmetric hashed fingerprint template is shown in Figure 10. The effort required for 100% coverage of the modified hashed fingerprint template with key value 0.25 is 0.94.

Figure 10 CE curve of symmetric hash and modified symmetric hash with dynamic threshold (see online version for colours)



6 Conclusions

The fingerprint template's security is strenuous among the biometric template protection because the fingerprint template is stored as minutiae points. The modified symmetric hash method uses a key value as a multiplication parameter for the hashing of the fingerprint biometric template. The irreversibility and unlikability analysis of the modified symmetric hashed fingerprint template exhibits better security. The multiplication of the fingerprint minutiae template by a key value mitigates the accuracy of matching performance. The degradation in the accuracy of matching is overcome by the use of a dynamic threshold matching algorithm. The hashing of fingerprint template with key value 0.5 and 0.25 increases the EER to 18.91 and 33.12, respectively. Dynamic threshold values reduce the EER to 1.83 for secret key 0.5 and 3.58 for secret key 0.25. The performance analysis of the modified hashed fingerprint template with a dynamic threshold matching algorithm in terms of EER, FMR100, FMR1000, ZeroFMR, ZeroFNMR, and ROC curve shows better results.

References

Ajish, S. and Kumar, K.A. (2020) 'Security and performance enhancement of fingerprint biometric template using symmetric hashing', *Computers & Security*, Vol. 90, No. 1, p.101714.

Belhadj, F. (2017) *Biometric System for Identification and Authentication*, PhD thesis, Ecole Nationale Supérieure en Informatique, Alger.

Bremananth, R. and Chitra, A. (2006) 'New methodology for a person identification system', *Sadhana*, Vol. 31, No. 3, pp.259–276.

Cappelli, R., Ferrara, M., Maltoni, D. and Tistarelli, M. (2010) 'MCC: a baseline algorithm for fingerprint verification in FVC-ongoing', in *2010 11th International Conference on Control Automation Robotics & Vision*, IEEE, pp.19–23.

Cappelli, R., Maio, D., Maltoni, D., Wayman, J.L. and Jain, A.K. (2005) 'Performance evaluation of fingerprint verification systems', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 28, No. 1, pp.3–18.

Cavoukian, A., Stoianov, A. and Carter, F. (2008) 'Keynote paper: biometric encryption: technology for strong authentication, security and privacy', in *Policies and Research in Identity Management*, pp.57–77, Springer.

Dodis, Y., Ostrovsky, R., Reyzin, L. and Smith, A. (2008) 'Fuzzy extractors: how to generate strong keys from biometrics and other noisy data', *SIAM Journal on Computing*, Vol. 38, No. 1, pp.97–139.

Dodis, Y., Reyzin, L. and Smith, A. (2004) 'Fuzzy extractors: how to generate strong keys from biometrics and other noisy data', in *International Conference on the Theory and Applications of Cryptographic Techniques*, pp.523–540, Springer.

Ferrara, M., Maltoni, D. and Cappelli, R. (2012) 'Noninvertible minutia cylinder-code representation', *IEEE Transactions on Information Forensics and Security*, Vol. 7, No. 6, pp.1727–1737.

Fu, X., Liu, C., Bian, J., Feng, J., Wang, H. and Mao, Z. (2013) 'Extended clique models: a new matching strategy for fingerprint recognition', in *2013 International Conference on Biometrics (ICB)*, IEEE, pp.1–6.

Gomez-Barrero, M., Galbally, J., Rathgeb, C. and Busch, C. (2017) 'General framework to evaluate unlinkability in biometric template protection systems', *IEEE Transactions on Information Forensics and Security*, Vol. 13, No. 6, pp.1406–1420.

Gomez-Barrero, M., Rathgeb, C., Galbally, J., Busch, C. and Fierrez, J. (2016) 'Unlinkable and irreversible biometric template protection based on bloom filters', *Information Sciences*, Vol. 370, No. 2016, pp.18–32.

Inuma, M. (2014) 'A relation between irreversibility and unlinkability for biometric template protection algorithms', *Josai Mathematical Monographs*, Vol. 7, No. 2014, pp.55–65.

ISO/IEC (2013) *Information Technology – Security Techniques–Information Security Management Systems – Requirements*, ISO/IEC, J.T.C.I.J.S.S., No. 27.

Jain, A.K. and Dubes, R.C. (1988) *Algorithms for Clustering Data*, Prentice-Hall, Englewood Cliffs, New Jersey.

Jain, A.K., Feng, J., Nagar, A. and Nandakumar, K. (2008a) 'On matching latent fingerprints', in *2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, IEEE, pp.1–8.

Jain, A.K., Nandakumar, K. and Nagar, A. (2008b) 'Biometric template security', *EURASIP Journal on Advances in Signal Processing*, Vol. 2008, No. 1, pp.1–17.

Jin, A.T.B., Ling, D.N.C. and Goh, A. (2004) 'Biohashing: two factor authentication featuring fingerprint data and tokenised random number', *Pattern Recognition*, Vol. 37, No. 11, pp.2245–2255.

Jochem, W.C., Bird, T.J. and Tatem, A.J. (2018) 'Identifying residential neighbourhood types from settlement points in a machine learning approach', *Computers, Environment and Urban Systems*, Vol. 68, No. 1, pp.104–113.

Juels, A. and Sudan, M. (2006) 'A fuzzy vault scheme', *Designs, Codes and Cryptography*, Vol. 38, No. 2, pp.237–257.

Juels, A. and Wattenberg, M. (1991) 'A fuzzy commitment scheme', in *Proceedings of the 6th ACM Conference on Computer and Communications Security*, pp.28–36.

Khanyile, N.P., de Kock, A. and Mathekga, M.E. (2014) 'Similarity score computation for minutiae-based fingerprint recognition', in *IEEE International Joint Conference on Biometrics*, IEEE, pp.1–8.

Kumar, G., Tulyakov, S. and Govindaraju, V. (2010) 'Combination of symmetric hash functions for secure fingerprint matching', in *2010 20th International Conference on Pattern Recognition*, IEEE, pp.890–893.

Lee, C., Choi, J-Y., Toh, K-A., Lee, S. and Kim, J. (2007) 'Alignment-free cancelable fingerprint templates based on local minutiae information', *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, Vol. 37, No. 4, pp.980–992.

Maltoni, D. (2005) 'A tutorial on fingerprint recognition', *Advanced Studies in Biometrics*, Vol. 3161, No. 1, pp.43–68.

Medina-Perez, M.A., Garcia-Borroto, M., Gutierrez-Rodriguez, A.E. and Altamirano-Robles, L. (2012) 'Improving fingerprint verification using minutiae triplets', *Sensors*, Vol. 12, No. 3, pp.3418–3437.

Murmu, N. and Otti, A. (2009) *Fingerprint Recognition*, PhD thesis.

Nagar, A. and Jain, A.K. (2009) 'On the security of non-invertible fingerprint template transforms', in *2009 First IEEE International Workshop on Information Forensics and Security (WIFS)*, IEEE, pp.81–85.

Nagar, A., Nandakumar, K. and Jain, A.K. (2010) 'Biometric template transformation: a security analysis', in *Media Forensics and Security II*, Vol. 7541, p.75410, International Society for Optics and Photonics.

Ratha, N.K., Chikkerur, S., Connell, J.H. and Bolle, R.M. (2007) 'Generating cancelable fingerprint templates', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 29, No. 4, pp.561–572.

Ratha, N.K., Connell, J.H. and Bolle, R.M. (2001) 'Enhancing security and privacy in biometrics-based authentication systems', *IBM Systems Journal*, Vol. 40, No. 3, pp.614–634.

Sabir, M. (2018) 'Sensitivity and specificity analysis of fingerprints based algorithm', in *2018 International Conference on Applied and Engineering Mathematics (ICAEM)*, IEEE, pp.1–5.

Tulyakov, S., Farooq, F., Mansukhani, P. and Govindaraju, V. (2007) 'Symmetric hash functions for secure fingerprint biometric systems', *Pattern Recognition Letters*, Vol. 28, No. 16, pp.2427–2436.

Uludag, U., Pankanti, S., Prabhakar, S. and Jain, A.K. (2004) 'Biometric cryptosystems: issues and challenges', *Proceedings of the IEEE*, Vol. 92, No. 6, pp.948–960.

Wieclaw, Å. (2009) 'A minutiae-based matching algorithms in fingerprint recognition systems', *Journal of Medical Informatics & Technologies*, Vol. 13, No. 1, pp.65–71.