A Study on Fingerprint Hash Code Generation Using Euclidean Distance for Identifying a User

Krishna Prasad K. [1] **& P. S. Aithal**[2]

[1] Research Scholar, College of Computer and Information Science, Srinivas University, Mangaluru-575001, Karnataka, INDIA.

[2] College of Computer and Information Science, Srinivas University, Mangaluru-575001, Karnataka, INDIA.

E-mail: karanikrishna@gmail.com

Type of the Paper: Research Paper.
Type of Review: Peer Reviewed.
Indexed in : OpenAIRE.
DOI :
Google Scholar Citation: IJMTS

How to Cite this Paper:
Krishna Prasad, K., Aithal, P. S. (2017). A Study on Fingerprint Hash Code Generation Using Euclidean Distance for Identifying a User. *International Journal of Management, Technology, and Social Sciences (IJMTS)*, 2(2), 116-126. DOI :

International Journal of Management, Technology, and Social Sciences (IJMTS)
A Refereed International Journal

A Study on Fingerprint Hash Code Generation Using Euclidean Distance for Identifying a User

Krishna Prasad K. [1] **& P. S. Aithal**[2]

[1] Research Scholar, College of Computer and Information Science, Srinivas University, Mangaluru-575001, Karnataka, INDIA.

[2] College of Computer and Information Science, Srinivas University, Mangaluru-575001, Karnataka, INDIA.

E-mail: karanikrishna@gmail.com

ABSTRACT

Biometrics innovation has ended up being a precise and proficient response to the security issue. Biometrics is a developing field of research as of late and has been dedicated to the distinguishing proof or authentication of people utilizing one or multiple inherent physical or behavioural characteristics. The unique fingerprint traits of a man are exceptionally exact and are special to a person. Authentication frameworks in light of unique fingerprints have demonstrated to create low false acceptance rate and false rejection rate, alongside other favourable circumstances like simple and easy usage strategy. But the modern study reveals that fingerprint is not so secured like secured passwords which consist of alphanumeric characters, number and special characters. Fingerprints are left at crime places, on materials or at the door which is usually class of latent fingerprints. We cannot keep fingerprint as secure like rigid passwords. In this paper, we discuss fingerprint image Hash code generation based on the Euclidean distance calculated on the binary image. Euclidean distance on a binary image is the distance from every pixel to the nearest neighbour pixel which is having bit value one. Hashcode alone not sufficient for Verification or Authentication purpose, but can work along with Multifactor security model or it is half secured. To implement Hash code generation we use MATLAB2015a. This study shows how fingerprints Hash code uniquely identifies a user or acts as index-key or identity-key.

Keywords: Fingerprint image, Fingerprint hashcode, Authentication, Multifactor authentication model, Euclidean distance.

1. INTRODUCTION

Biometrics is an investigation of checking and setting up the identity of an individual through physiological components or behavioural qualities. Even though biometric technologies differ in complexities, capacities and performance parameters, still all offer a few regular components like biometric sensor module, feature extractor module, a matching module, decision-making module and system database. Fingerprint biometric has been utilized in numerous areas together with entrance management and door-lock programs, smart cards, vehicle ignition control framework and fingerprint controlled access control system. Automatic Fingerprint Identification System (AFIS) consists of different steps like preprocessing, enhancement, segmentation, thinning, feature extraction, post-processing, minutiae orientation and alignment [1-6]. The distinctiveness of fingerprint is added forward by using ridge patterns and it has been proved that the information in small regions of friction ridges is in no way repeated. These friction ridges broaden in a human system all through the fetus level itself Fingerprint sensors or acquisition devices uses different types of

sensors to take input or to get fingerprint image into the system [7].

Commonly, all the profitable biometric systems shield the stored templates by using encrypting those using general cryptographic techniques. Either a public key cryptosystem like RSA (RSA laboratories, 1999) or a symmetric key cipher like AES (Advanced Encryption Standard, 2001) is usually used for template encryption.

One of the important challenges in biometric identification or verification system is keeping the biometric data or template safe and secure. A hash function is usually transformed functions, which converts or transform data or features from one form to another. Always transform function should be a one-way function or another way it should not be invertible [8].

A number of template protection strategies like fuzzy commitment [9], fuzzy vault [9], protecting functions [10] and distributed supply coding [11] can be considered as the key binding biometric cryptosystem. Different schemes for securing biometric templates along with those positioned forth in [12-15] also fall under this class.

In this study, we calculate Euclidean distance for a binary fingerprint image, which is a straight line distance from a pixel with value zero to the pixel with value non-zero, which is one in a binary image using Euclidean norm. The Euclidean distance is calculated for all the pixels of the binary fingerprint image. The two points k and l in two-dimensional Euclidean spaces and k with the coordinates (k1, k2), l with the coordinates (l1, l2). The line segment with the endpoints of k and l will form the hypotenuse of a right-angled triangle. The space among factors k and l is defined as the square root of the sum of the squares of the differences among the corresponding coordinates of the points. In a two-dimensional Euclidean geometry Euclidean distance between two points k = (kx, ky) and l = (lx, ly) is given as follows;

$$d(k, l) = \sqrt{(lx - kx)^2 + (ly - ky)^2}$$

For example consider a 3×3 sized matrix with values as follows

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

The Euclidean distance for each point is calculated as follows

$$\begin{bmatrix} 1.4142 & 1.0000 & 1.4142 \\ 1.0000 & 0 & 1.0000 \\ 1.4142 & 1.0000 & 1.4142 \end{bmatrix}$$

The most natural or common matrix for finding distance matrix in the binary image is Euclidean distance [16-18]. Due to the lack of efficient algorithms in the field of Euclidean distance led to the development of many types of research in this field in order to define, elaborate and also to use some other methods to find the distance using other methods like the city block, chessboard or chamfer [18-20]. The Euclidean distance transform is global operation and the calculation of Euclidean distance is most common and simple operation and amount of calculation required is always directly proportional to the size of the entire image because this is calculated for every pixel.

Fingerprints are a half-secret if passwords are leaked or hacked, it easily revocable using another password. But in a biometric security system, which uses only biometric features, is not easy to change fingerprint key or fingerprint are static biometric, which never change much throughout the lifespan. Fingerprints are left at the car, door or anyplace where every person goes and places his finger [21]].

Fingerprint Hash code is not used for full security or authentication purpose but it can be combined with other security mechanisms like password or OTP in order to enhance security. Fingerprint Hash code acts as the key, which can uniquely identify every person. So it can be replaceable with user-id or username and can work along with text-based or picture based or pattern based passwords. The fingerprint hash code is not constant with biometric sensors or reader [22].

This paper has sections. Section-1 explains about introductory information of fingerprint and Euclidean distance, and template protection. Section-2, explain about objective and methodology of the study. Section-3 explains

about Algorithm of Hash code generation, Section-4 depicts a flowchart of Hash code generation. Section-5 explains Results and Discussions. Section-6 concludes the paper.

2. OBJECTIVES AND METHODOLOGY

There are many types of research are carried out translation and rotation invariant fingerprint hash code generation but even small or pixel changes cause difference in Hash code. So this research does not concentrate on developing fingerprint hash code which is translation and rotation invariant. Fingerprint alone not gives full security, in order to improve the security of the system fingerprint acts one factor along with OTP, password, or any other biometric psychological or behavioral traits. The main objectives of this study are given below.

- To Study a Fingerprint Hash code generation using Euclidean distance value calculated for each pixel of the binary image.
- To verify the uniqueness of fingerprint Hash code using FVC ongoing 2002 benchmark dataset.

Figure 1 explains the methodology used in this research work.

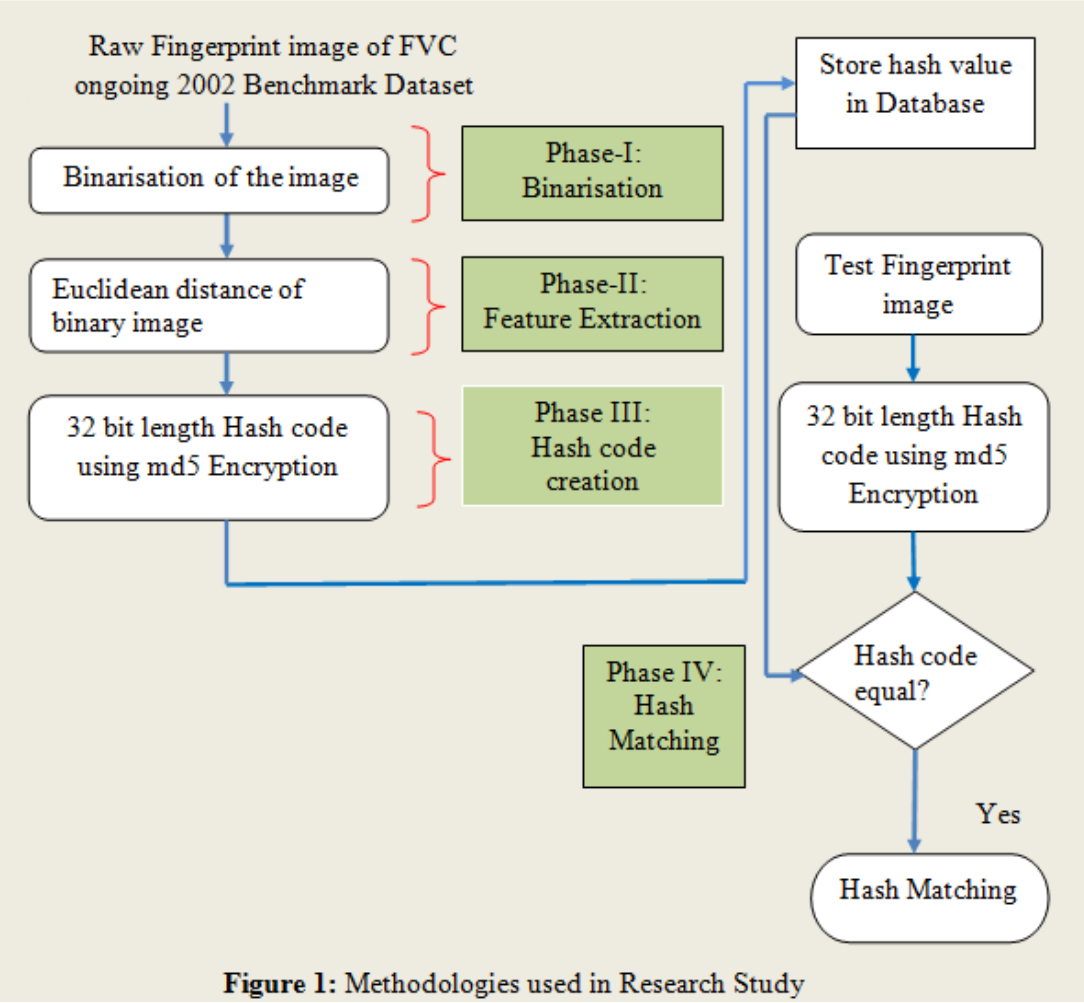


Figure 1: Methodologies used in Research Study

Here initially FVC ongoing 2002 benchmark dataset is considered for testing the hash code. The benchmark dataset image is binarised and Euclidean distance of the image is calculated for each pixel. The distinct values of the Euclidean distance matrices values are considered and 32-

bit length hash code is generated. Distinct Euclidean distance value summation, mean value and standard deviation values are considered for generating Hash code.

3. ALGORITHM OF HASHCODE GENERATION USING EUCLIDEAN DISTANCE

This section explains step by step procedure to develop Hashcode by making use of Euclidean distance matrix on a binary fingerprint image. The steps of the algorithm are explained below. The algorithm also shows the pseudo code.

Step 1: Input Grayscale fingerprint image
read (input_image)
Step 2: Convert input image into 256 × 256 sized two-dimensional image
resized_image = image_resize (input_image, [256, 256])
Step 3: Convert 256 × 256 sized grayscale image into binary image
binary_image = convert_to_binary(resized_image)
Step 4: Find the Euclidean distance of the image
euclidean_image = Euclidean_distance(binary_image)
Step 5: Find the distinct value of the Euclidean distance
distinct_eucllidean_value = distinct_value(euclidean_image)
Step 6: Find the distinct value summation
For i=1 to size(distinct_eucllidean_value)
euclidean_sum = distinct_eucllidean_value (i)
end for
Step 7: Find the mean of the distinct Euclidean value
euclidean_mean = mean(distinct_eucllidean_value)
Step 8: Find the standard deviation of the distinct Euclidean value
std_deviation = standard_deviation(distinct_eucllidean_value)
Step 9: Combine the value of Step-6, Step-7, and Step-8
combine_value = combine(euclidean_sum, euclidean_mean, std_deviation)
Step 10: Pass the value of Step-9 as parameter for MD5 Hash function
hash_value = MD5_DataHash(combine_value)

4. FLOWCHART OF HASHCODE GENERATION USING EUCLIDEAN DISTANCE

The above algorithm is explained using flowchart. The different process or work flow are listed below. With an intension to make the MD5 Hashcode more robust and to get the advantage of salting Euclidean distance sum, mean, and standard deviation are combined and passed to the MD5 algorithm.

- Converting input image to 256 × 256 sized grayscale image
- Converting to binary image
- Finding Euclidean distance
- Finding distinct value of the Euclidean distance
- Finding the sum of the distinct Euclidean distance
- Finding the mean of the distinct Euclidean distance
- Finding the standard deviation of the distinct Euclidean distance
- Generating MD5 Hashcode using combined sum, mean, and standard deviation of distinct Euclidean distance value

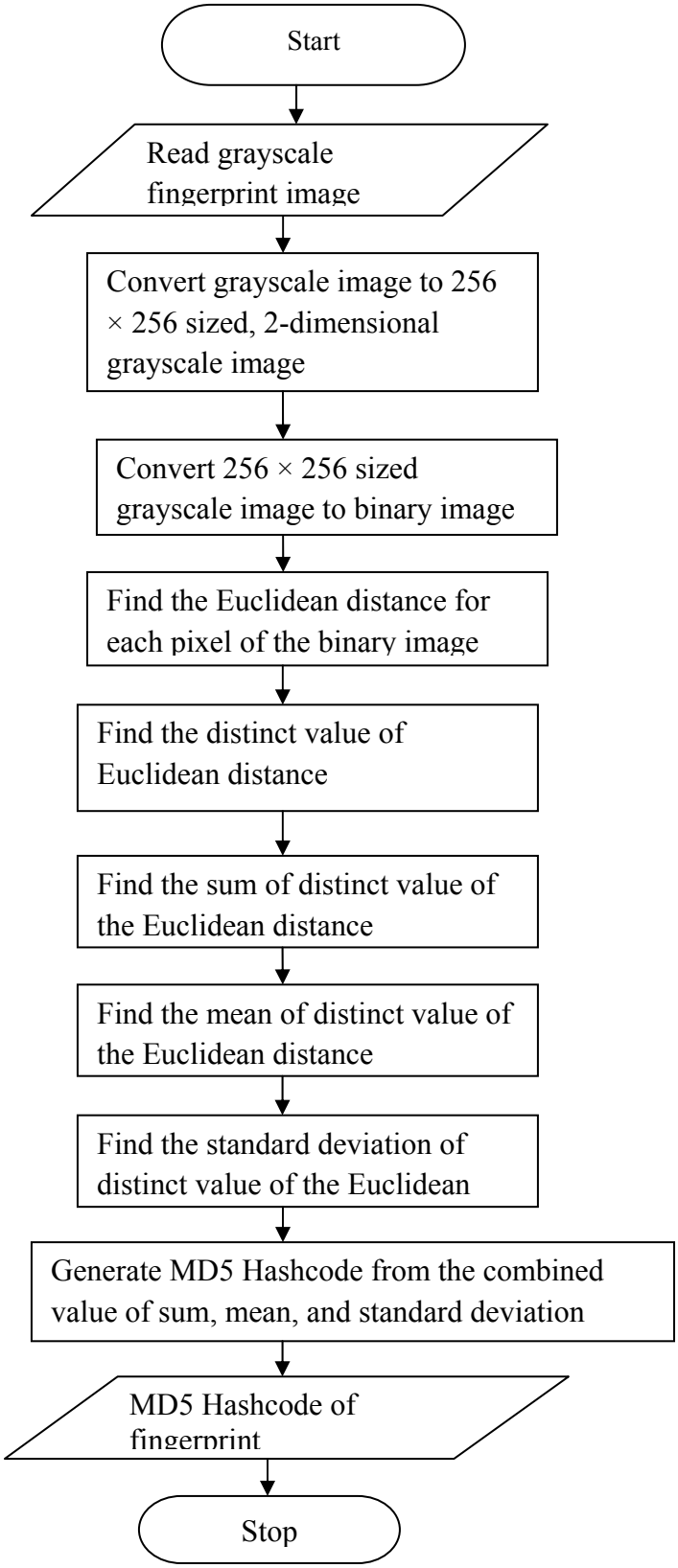


Figure 2: Flowchart of Hash code generation using Euclidean Distance

The process of the MD5 algorithm is disused below.
Input: Extracted Features
Output: Hash Code
Step-1: Attach the padded bits
Step-2: Append the length of the initial input to the result of the previous step-1
Step-3: Initialize MD buffer as A, B, C, D.
A four-word buffer (A, B, C, D) was used to evaluate the message digest. Here each of A, B, C, D is a 32- bit register
Step-4: Process message in 16-word blocks
Step-5: Finally, we get the 32-bit Hashcode as output

5. RESULTS AND DISCUSSIONS

In this study, WampServer is used to create a database. This database table contains two fields as id and Hashcode. The Hashcode generation using Euclidean distance is implemented using MATLAB2015a. The configuration of the system used to implement this study is given in Table 1.

Table 1: Configuration of System used for finding Execution Time

Sr. No.	Parameters	System Details
1	Model	Compaq 435
2	Processor	AMD E-350 processor 1.60 GHz
3	Installed Memory	3 GB (2 GB usable)
4	System Type	32-bit Operating System
5	Operating System	Windows 7 Starter
6	Software	MATLAB 2015a 32-bit

The execution time for different randomly selected images of FVC ongoing 2002 dataset is shown in Table 2.

Table 2: Execution time of the training phase

Method Name	Image name	Execution Time (in seconds)	Average
Method-1	101_1	0.507921	0.144420
	101_5	0.245508	
	102_2	0.146157	
	103_3	0.108258	
	104_4	0.102478	
	104_7	0.056262	
	104_8	0.068901	
	105_8	0.080591	
	106_6	0.114282	
	109_3	0.117105	
	109_8	0.109788	
	110_3	0.104671	
	110_8	0.115539	

The average execution time of the fingerprint Hashcode generation using Euclidean distance is very good and it is approximately 0.144420. Here we only consider the training phase. The testing phase includes around 0.44 seconds more than training phase. If the configuration of the system increases definitely execution time also increases. Table-3 shows the Hashcode generated based on an MD5 algorithm using Euclidean distance.

Table-3: Hash code generated using Euclidean distance

Serial No.	Hashcode
1	e06c186b309ba7351d716b519d7c73b2
2	6e621fa2509d451735cc3a6371ddb5bc
3	58700de96bb19d7fae96279f37e2f134
4	63fd88581c026148ff47df64ccb1d070
5	1dae3325e72f45e183431fb2bbd79377
6	2ce72a2f2b342594c2333f607b8da5f5
7	52fee94aa0ae0bacd8450613997f181d
8	d9d9fa4f656ce9f56cc09aaf4633a588
9	5c3035d3ae414d8ebf7bf117de58c21c
10	911cd7041e72ae334477ef593b217666
11	1d00b55914c7559b8cab2569c9a035a3
12	5fb8835210967067ce0612ae341222ce
13	375eaf11d2909e267ea8e37012895d0b

The screenshots of the grayscale fingerprint image capture is shown using Figure 3.

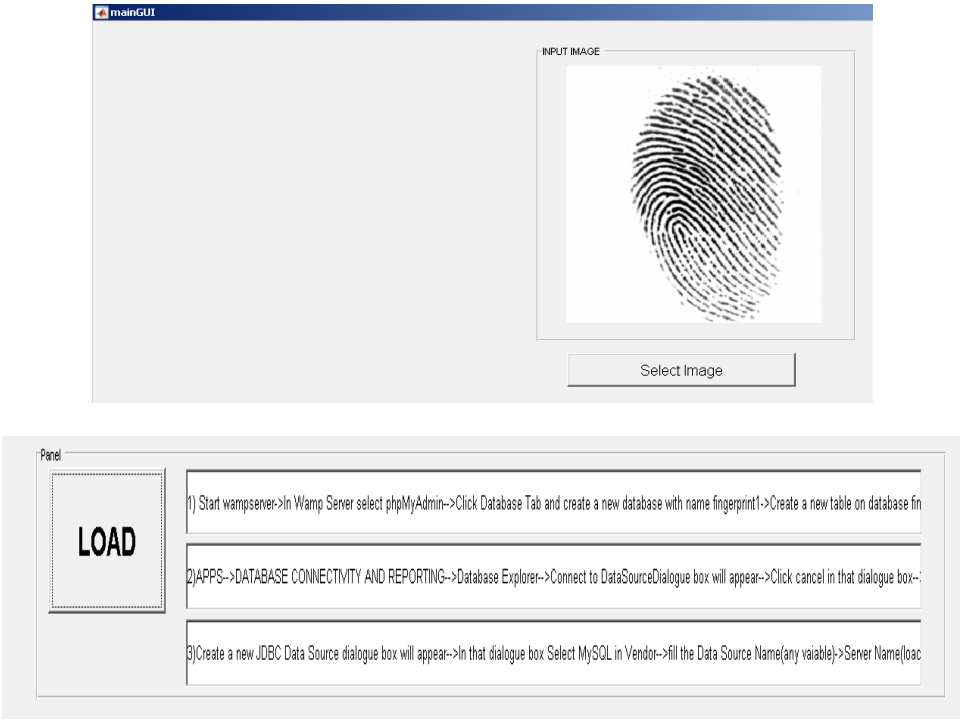
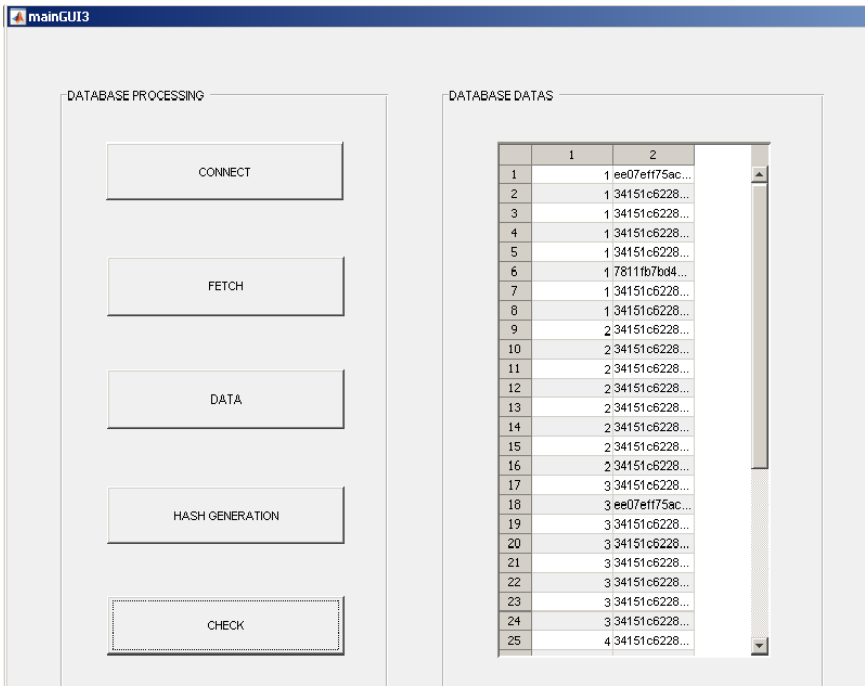


Figure 3: Screenshots of Fingerprint image capture

The screenshot or Figure 3 contains two push buttons. One push button is used to select grayscale fingerprint image. Another one gives instruction to create WampServer and to connect this from MATLAB2015a. The screenshots of Database processing and status is shown using Figure 4.



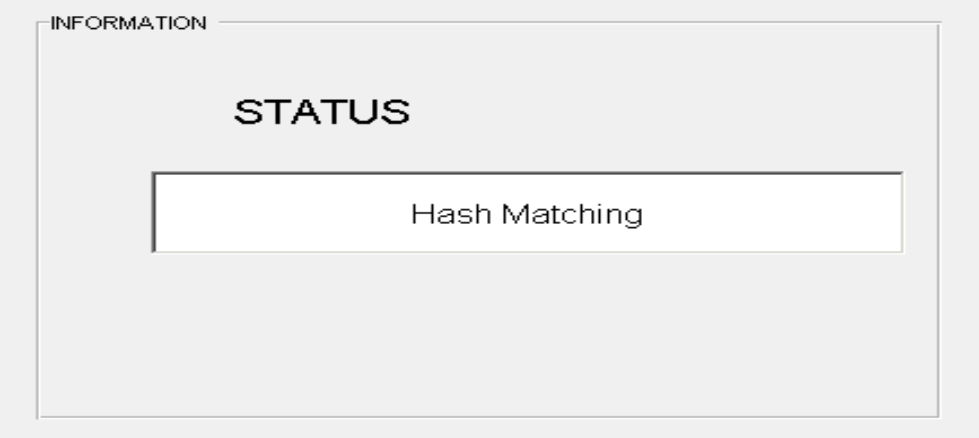


Figure 4: Screenshots of Database processing and Status

The data processing control of figure 4 consists of five push buttons as Connect, Fetch, Data, Hash Generation, and Check. Connect button is used to connect to the database, Fetch button is used to fetch records from the database, Data button is used to show data in tables, hash generation is used to generate Hashcode for sample input fingerprint, and Check is used to check sample input Fingerprint image is matching or not matching with already stored hash code.

Advantages of Hash value produced using Euclidean distance

- Hash code produced using Euclidean distance matrices are noninvertible
- Hash code takes very small amount of memory
- Hash code Hides original information of fingerprint image from the intruder
- The execution time of Hash code generation using Euclidean distance is very good.
- It is unique for each fingerprint of the same person means ten fingerprints will be having ten different Hash codes.

Benefits of Hash value produced using Euclidean distance

- Hash code is used as identity-key or index-key for unique identification purpose of a user.
- Easily we can append salting in order to make the Hash code more robust.
- Fingerprint Hash code is a transformed function, which does not reveal original minutiae details.
- Fingerprint Hash code consumes very less time for training phase.
- Unlike another fingerprint matching, this study does not use scoring level. It uses only binary value either matching or not matching.

Constraints of Hash value produced using Euclidean distance

- Small changes in fingerprint hash code make large differences.
- Fingerprint generation using Euclidean distance is translation and rotation variant which is not having much scope when the fingerprint is used for identification purpose rather than security purpose.

Disadvantages of Hash value produced using Euclidean distance

- Fingerprint hash code cannot be solely used for security or authentication purpose.
- If fingerprint image of same finger input is taken through any type of solid and robust sensors in consecutive two intervals, still fingerprint hash code generates different hash code.
- Even though developed fingerprint Hash code is invariant to translation and rotation, if the user presses hardly into one reader or sensor, or swipe the finger in a different orientation, or a cut in the finger, for a

successive two capture, produces different Hash code.

6. CONCLUSION

Even though fingerprints are most common and easily usable and many research contribution available areas of biometrics, which is having some flaws like which left by a human being at many places like door, wall, on the car and many more places are easily mimicked by fraud or intruder. The fingerprint does not get matched when the finger has some cut or wound and sensors are not able to recognize in some weather conditions like winter season. The fingerprint is effective as identity or index key and not as a full security feature. It works well with multifactor biometrics authentication as one major factor.

In this paper, we developed a Hash code based on an MD5 Hash function by making use of Euclidean distance of binary fingerprint image. This Hashcode can be effectively used as Index-key or identity-key. This method shows considerably better execution time. This method also gives 100% accurate matching as far as input fingerprint image is once captured and stored static digital fingerprint image. If we capture through sensor each time this gives different Hash code. So this method is not suitable for solely security purpose.

REFERENCES

[1] Krishna Prasad, K. & Aithal, P.S. (2017). A Conceptual Study on Image Enhancement Techniques for Fingerprint Images. *International Journal of Applied Engineering and Management Letters (IJAEML),* 1(1), 63-72. DOI: http://dx.doi.org/10.5281/zenodo.831678.

[2] Krishna Prasad, K. & Aithal, P.S. (2017). Literature Review on Fingerprint Level 1 and Level 2 Features Enhancement to Improve Quality of Image. *International Journal of Management, Technology, and Social Sciences (IJMTS),* 2(2), 8-19. DOI: http://dx.doi.org/10.5281/zenodo.835608.

[3] Krishna Prasad, K. & Aithal, P.S. (2017). Fingerprint Image Segmentation: A Review of State of the Art Techniques. *International Journal of Management, Technology, and Social Sciences (IJMTS),* 2(2), 28-39. DOI: http://dx.doi.org/10.5281/zenodo.848191.

[4] Krishna Prasad, K. & Aithal, P.S. (2017). A Novel Method to Contrast Dominating Gray Levels during Image contrast Adjustment using Modified Histogram Equalization. *International Journal of Applied Engineering and Management Letters (IJAEML),* 1(2), 27-39. DOI: http://dx.doi.org/10.5281/zenodo.896653.

[5] Krishna Prasad, K. & Aithal, P.S. (2017).Two Dimensional Clipping Based Segmentation Algorithm for Grayscale Fingerprint Images. *International Journal of Applied Engineering and Management Letters (IJAEML),* 1(2), 51-65. DOI: http://dx.doi.org/10.5281/zenodo.1037627.

[6] Krishna Prasad, K. & Aithal, P.S. (2017). A conceptual Study on Fingerprint Thinning Process based on Edge Prediction. *International Journal of Applied Engineering and Management Letters (IJAEML),* 1(2), 98-111. DOI: http://dx.doi.org/10.5281/zenodo.1067110.

[7] Krishna Prasad, K. (2017). A Critical Study on Fingerprint Image Sensing and Acquisition Technology. *International Journal of Case Studies in Business, IT and Education (IJCSBE),* 1(2), 86-92. DOI: http://dx.doi.org/10.5281/zenodo.1130581.

[8] Tulyakov, S., Farooq, F., Mansukhani, P., & Govindaraju, V. (2007). Symmetric hash functions for secure fingerprint biometric systems. *Pattern Recognition Letters*, *28*(16), 2427-2436.

[9] Juels, A. (2002). M. Sudan 'A fuzzy vault scheme'. In *Proceedings of the 2002 IEEE International Symposium on Information Theory* (Vol. 408).

[10] Tuyls, P., Akkermans, A. H., Kevenaar, T. A., Schrijen, G. J., Bazen, A. M., & Veldhuis, R. N. (2005, July). Practical biometric authentication with template protection. In *AVBPA* (Vol. 3546, pp. 436-446).

[11] Holst, J. C., & Draper, D. A. (1999). *U.S. Patent No. 5,999,039*. Washington, DC: U.S. Patent and Trademark Office.

[12] Davida, G. I., Frankel, Y., Matt, B., & Peralta, R. (1999). On the relation of error correction and cryptography to an online biometric based identification scheme. In *Workshop on coding and cryptography*.

[13] Hao, F, Anderson, R & Daugman, J 2006, _Combining Crypto with Biometrics Effectively', IEEE Transactions on Computers, vol. 55, pp. 1081-1088.

[14] Kelkboom, E. J., Gökberk, B., Kevenaar, T. A., Akkermans, A. H., & van der Veen, M. (2007, August). "3D face": biometric template protection for 3D face recognition. In *International Conference on Biometrics* (pp. 566-573). Springer, Berlin, Heidelberg.

[15] Connie, T., Teoh, A., Goh, M., & Ngo, D. (2005). PalmHashing: a novel approach for cancelable biometrics. *Information processing letters*, *93*(1), 1-5.

[16] Das, P. P., Chakrabarti, P. P., & Chatterji, B. N. (1987). Distance functions in digital geometry. *Information Sciences*, *42*(2), 113-136.

[17] Yamada, H. (1984). Complete Euclidean distance transformation by parallel operation. In *Proc. of 7th Int. Conf. on Pattern Recognition, Montreal* (Vol. 1, pp. 69-71).

[18] Borgefors, G. (1986). Distance transformations in digital images. *Computer vision, graphics, and image processing*, *34*(3), 344-371.

[19] Danielsson, P. E. (1980). Euclidean distance mapping. *Computer Graphics and image processing*, *14*(3), 227-248.

[20] Yamashita, M., & Ibaraki, T. (1986). Distances defined by neighborhood sequences. *Pattern Recognition*, *19*(3), 237-246.

[21] https://hackaday.com/2015/11/10/your-unhashable-fingerprints-secure-nothing/, Last Accesses Date: 05-12-2017.

[22] https://security.stackexchange.com/questions/42384/is-there-any-way-to-cryptographically-hash-a-human-thumbprint, Last Accesses Date: 05-12-2017.

[23] Aithal, P. S. (2016). A Review on Advanced Security Solutions in Online Banking Models, International Journal of Scientific Research and Modern Education (IJSRME), 1(1), 421-429. DOI : http://doi.org/10.5281/zenodo.160971.

[24] Aithal, P. S. (2015). Biometric Authenticated Security Solution to Online Financial Transactions. *International Journal of Management, IT and Engineering (IJMIE)*, 5(7), 455-464, DOI : http://doi.org/10.5281/zenodo.268875.

****