Fusion Encryption Technique for Finger Print Matching with Text

Manju Mandot*, S.S. Sarangdevot**, and Sharad Verma***

***Abstract*—**A fingerprint in its narrow sense is an impression left by the friction ridges of a human finger. In a wider use of the term, fingerprints are the traces of an impression from the friction ridges of any part of a human or other primate hand. Various methods have been implemented for inducting effective and efficient results. It is a set theory approach to digital image processing based on finger prints. In this paper, we proposed a fusion encryption fingerprint matching algorithm, which is secure against side channel attacks. An algorithm based on the local structure of the minutiae is presented to match the fingerprints with text. The MATLAB code has been written for fusion encryption fingerprint matching algorithm.

***Keywords*—** Minutiae, Feature Extraction, Fingerprint Recognition, Secure Matching.

I. Introduction

BIOMETRIC recognition systems offer greater security and convenience than traditional methods of personal recognition [1,2]. Along with the rapid growing of this emerging technology, the system performance, such as accuracy and speed, is continuously improved. At the same time, the security of the biometric system itself is becoming more and more important. One of the most significant disadvantages of the biometric recognition system is that they cannot be easily recalled. Therefore the secure storage of the biometric template is becoming extremely important. In a traditional biometric recognition system, the biometric template, such as fingerprint, voice, etc. is usually stored on a central server during enrollment. The input biometric signal captured by the front-end sensor is sent to the server and the processing and matching steps are performed on the server [3]. Embedded biometric recognition systems try to solve this problem by moving the signal processing and matching engines from the server to the embedded device. In these systems, the biometric signals are processed and matched on the embedded device and only the result is transmitted to the server. This approach can avoid the attacks on communication and server. It also avoids that the biometric data needs to be stored on multiple servers for multiple applications. However, it is very easy to compromise the plain-text storage of the template in the embedded device. To make the storage more secure, the biometric template is encrypted using a secret key before being stored. As soon as the input signal has come, the matcher decrypts the template and performs the comparison. However, some dedicated attacks can still extract the secure key, and in turn, the template. The reason for this is that the physical implementation of an algorithm provides attackers with some important information.

*Associate Professor, JRN Rajasthan Vidyapeeth, University, Udaipur
** Vice-Chancellor, JRN Rajasthan Vidyapeeth, University, Udaipur
*** Research Scholar, Deptt of computer Science and IT
JRN, Rajasthan Vidyapeeth, University, Udaipur.

II. Fingerprint Matching

There are two basic types of fingerprint matching techniques:

Graph based and Minutiae based. For modern embedded fingerprint recognition systems, the minutiae-based matching is popular because, on the one hand, the minutiae of the fingerprint

are widely believed the most discriminating and reliable features, and on the other hand, the template size of the biometric information based on minutiae is much smaller and the processing speed is higher than that of graph-based fingerprint matching. These characteristics are very important for saving memory and energy on the embedded devices. Lots of work has been done for minutiae-based fingerprint matching. Some of them use the local structure of the minutiae to describe the characteristics of the minutiae set [5]. This approach has high processing speed and robustness to rotation and partial prints. However, the local structure usually has less distinct features because it only represents some parts of the whole minutiae set. Prints from different fingers may have quite a few similar local structures by coincidence while prints from the same finger may only have very few similar structures due to the presence of false minutiae and the absence of genuine minutiae. Alignment-based matching algorithms take use of the shape of the ridge connected to minutiae [6,8]. This might improve the system accuracy. However, this approach results in a larger template size because the associated ridges for each minutia must be saved. Some other researches combine the local and global structures [7,8]. The local structure is used to find the correspondence of two minutiae sets and increase the reliability of the global matching. The global structure of minutiae reliably determines the uniqueness of a fingerprint.

III. Fingerprint Recognition Techniques

Most of the existing fingerprint techniques in literature are based on minutiae points which are represented using their

co-ordinate locations in the image[4]. When test fingerprint image is rotated with respect to enrolled image or partially available, these techniques face problem in matching due to change in the co-ordinate locations of the minutiae points and perform very poorly. These two cases are discussed below.

3.1. Proposed Approach

Here three algorithms are presented with their implementation code for the encryption. This encryption technique is based on the fingerprint, name and date of birth of the candidate. This algorithm encrypts the fingerprint by the following algorithms that are given below and then for more security he writes the text (combination of candidate name and date of birth) on encrypted image. This text is look like a watermark on the image.

IV. Encryption Algorithm for Fingerprint Image with Text

The algorithm is designed for encryption of the image with text. In this read a input image with the candidate name & date of Birth. Firstly input image is convert into gray scale mode and then in mat2 gray image. Then apply the erosion & dilution operation on mat2gray image. After this operation subtract the erosion operation result and dilution operation result. This subtraction result is rotated in to both positive and negative direction with theta (θ) angle. Rotated image in both directions (positive and negative) is fused with the candidate name and DOB.

Result of Encryption Algorithm for fingerprint image

Fig. 1: Input image

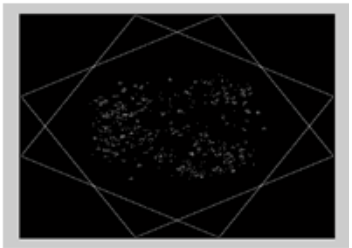

Fig. 2: Fused image in mean mode

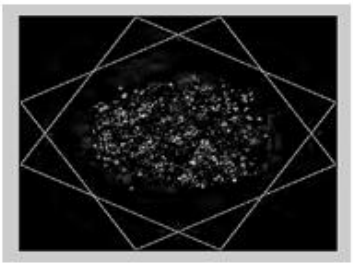

Fig. 3: Fused image in max mode

V. Encryption Algorithm for fingerprint image and text using Binarization

The algorithm is designed for the binarized image, name and date of birth of candidate. In this take the input of candidate finger print, name and DOB form the user. Then convert the fingerprint, image into gray scale mode and then perform mat2gray conversion. This mat2gray image is convert into binarization and perform dilation and erosion operation with binarized result. Subtract erosion and dilation result and rotate subtract result into positive and negative direction with theta (θ) angle. The Result of the positive and negative rotation is fused and stores it. The fused result is again encrypted with text as candidate name and DOB.

Result of Encryption Algorithm for fingerprint image and text using Binarization.

Fig. 4 : Input image

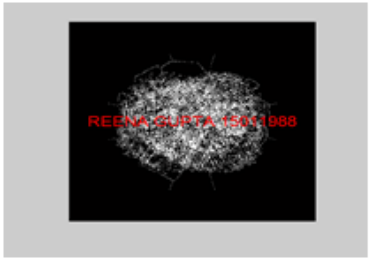

Fig. 5 : BXFUSmean-XFUSmean

Fig. 6: XFUSmean- BXFUSmean

VI. Fusions Fingerprint Image with Text Algorithm

This algorithm is fusion of first two algorithms. In this approach, take a candidate fingerprint as input image and their name and date of birth. Then convert input image into gray scale mode and then convert it into mat2 gray image now perform all the first algorithm action with mat2gray image. Like erosion, dilation, subtraction of it rotation and fusion. Now convert the mat2gray image into binarized image. Then perform all the second algorithm action with binarized image. Like erosion, dilation, subtraction of it, rotation and fusion. Now subtract the final result of first algorithm and final result of second algorithm and store the

result into 'end result' variable. Then write the text (name and DOB) son the 'end result' variable.

Algorithm:

Step 1: Read the input image.
Step 2: Enter the name and date of birth of candidate.
Step 3: Convert the RGB image into gray scale
Step 4: Convert Gray scale image into Mat2gray form
Step 5:FinalDiffuseimg1
Step 6: FinalDiffuseimg2
Step 7: Subtract FinalDiffuseimg1 and FinalDiffuseimg2.
Step 8: Write the text on the resultant image.

MATLAB CODE:

```
function [Result1, Result2]= cybersecurity(name, dob, image)
employeename=name;
employeedob=dob;
detail=[employeename ' ' employeedob];
img=imread(image);
figure,imshow(img);
K=imfinfo(image);
if K.ColorType == 'truecolor'
grayimg=rgb2gray(doubleimg);
figure, imshow(grayimg);
m2gimg=mat2gray(grayimg);
figure, imshow(m2gimg);
else
m2gimg=mat2gray(doubleimg);
figure, imshow(m2gimg);
end
BWim = bwperim(m2gimg);
figure, imshow(BWim);
BWmor = bwmorph(m2gimg,'skel',Inf);
figure, imshow(BWmor);
sub2=BWim - BWmor;
figure,imshow(sub2);
imgM60=imrotate(sub2, -60);
figure,imshow(imgM60);
imgA60=imrotate(sub2, +60);
figure,imshow(imgA60);
XFUSmean =
wfusimg(imgA60,imgM60,'db2',5,'mean','mean');
figure,imshow(XFUSmean);
```

Fig. 7: Input Image

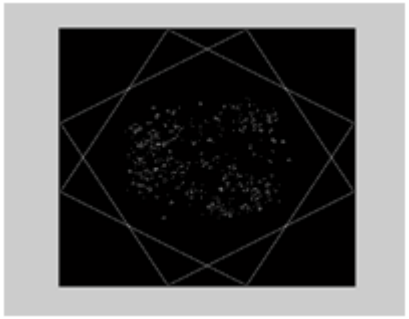

Fig. 8 : Fused image in mean mode with rotation

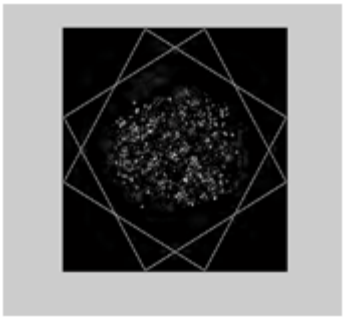

Fig. 9 : Fused image in max mode with rotation

```
XFUS = wfusimg(imgA60,imgM60,'sym4',5,'max','max');
figure,imshow(XFUS);
binaryimg=im2bw(m2gimg);
figure,imshow(binaryimg);
BBWim = bwperim(binaryimg);
figure, imshow(BBWim);
BBWmor = bwmorph(binaryimg,'skel',Inf);
figure, imshow(BBWmor);
Bsub2=BBWim - BBWmor;
figure,imshow(Bsub2);
BimgM60=imrotate(Bsub2, -60);
figure,imshow(BimgM60);
BimgA60=imrotate(Bsub2, +60);
figure,imshow(BimgA60);
BXFUSmean =
wfusimg(BimgA60,BimgM60,'db2',5,'mean','mean');
figure,imshow(BXFUSmean);
BXFUS =
wfusimg(BimgA60,BimgM60,'sym4',5,'max','max');
figure,imshow(BXFUS);
```

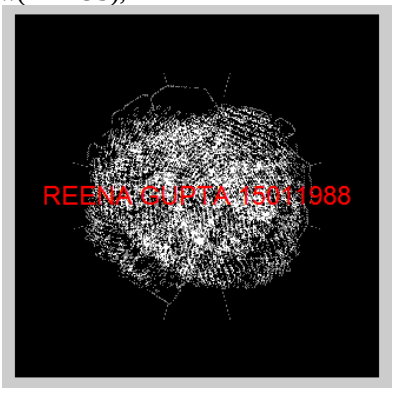

Fig.10: Difference image of mat2gray image and binary image

Fig. 11: Difference image of
binary image & mat2gray image

```
diff2=XFUSmean-BXFUSmean;
figure, imshow(diff2);
Diff3= BXFUSmean-XFUSmean;
        figure,imshow(Diff3);
        Bcenter=size(Diff3)/2;
text(Bcenter(1),Bcenter(2),['\fontsize{16}\color{red}',detail],
'HorizontalAlignment','center');
```

VII. Conclusion

In this paper, we present a novel secure fingerprint recognition system, in which the minutiae-based matching algorithm is robust against relative low quality of input fingerprint images and minutiae detection. The exact features of the finger prints and other images of the objects can be extracted with Morphological operations that include Erosion, dilation, boundary Extraction, skeletons, convex hull, morphological filtering, Thinning, pruning operations. Fusion Encryption Algorithm was developed and MATLAB coding written for template matching is used for figure print matching.

References

[1] A. K. Jain, A. Ross, and S. Prabhakar, "An introduction to biometric recognition," IEEE Trans. Circuits Syst. Video Technology, Special Issue Image- and Video-Based Biomet., vol. 14, no. 1, pp. 4–20, Jan.2004.

[2] R. M. Bolle, J. H. Connell, S. Pankanti, N. K. Ratha, and A. W. Senior, Guide to Biometrics. New York: Springer, 2004.

[3] J. L. Wayman, A. K. Jain, D. Maltoni, and D. Maio, Eds., Biometric Systems: Technology, Design and Performance Evaluation. New delhi.

[4] D. Maltoni, D. Maio, A. K. Jain, and S. Prabhakar, Handbook of Fingerprint Recognition. New York: Springer Verlag, Jun. 2003.

[5] Hrechak, AK and McHugh, JA. Automated fingerprint recognition using structural matching, Pattern Recognition, vol.23, no.8, 1990, pp.893-904. UK.

[6] Jain, A., Lin, H. and Bolle, R., On-line fingerprint verification, IEEE Transactions on Pattern Analysis & Machine Intelligence, vol.19, no.4, April 1997, pp.302-14. Publisher: IEEE Comput. Soc, USA.

[7] Jiang, X., Yau, W., Fingerprint minutiae matching based on the local and global structures, Proceedings 15th International Conference on Pattern Recognition. ICPR- 2000. IEEE Comput. Soc. Part, vol.2, 2000, pp.1038-41 vol.2. Los Alamitos, CA, USA.

[8] Shrivsubramani Krishnamoorthy, K P Soman," Implementation and Comparative Study of Image Fusion Algorithms" .International Journal of Computer Applications (0975 – 8887) Volume 9– No.2, November 2010.