Watermarking Algorithm for Encrypting Fingerprint Image -A MatLab Implementation

Ms. Madhu Chauhan
Assistant Professor, IINTM
Research Scholar, Mewar University
Delhi, India
myself_madhu26@yahoo.com

Dr. R.P. Saxena
Prof MTU
Noida
rp_saxena@yahoo.com

Abstract - In today's world of electronic communication we use to send and receive data through internet. Apart from sending data to the correct recipient, the aspect of secure transmission also comes to picture i.e. data should remain unhampered. There are many techniques and algorithm coming in existence to provide secure transmission of data. Biometrics being one of the fast growing industries for identifying a person also needs to be secure. In this paper we are suggesting an algorithm to encrypt a fingerprint sample so that it can securely be transmitted over internet.

Keywords: *Encryption, Minutiae, Binarization, Thinning,*

*****

I. Introduction

Many problems are faced at the time of identifying a person. These can either be resolved by verification or by recognition. Recognizing a person means finding the identity of a person, verification on the other hand refers to matching the identity with stated one.
Biometrics can be defined as an area of science which deals with the study of physical and behavioral traits of human being. There are various traits which can be studied in this field like, finger print, hand geometry, iris, retina and face etc. Out of the above mentioned trait we are emphasizing on fingerprint.
A fingerprint is a pattern of ridges (dark lines) and valleys (light areas) designed on the finger. Apart from this, there are certain minutiae points i.e. ridge ending or ridge bifurcation. These points can act as a useful tool for judging the identity of a person. Technically speaking, fingerprint being one of the most reliable traits of a person is used in the applications of recognition and verification.
Acquiring Finger Print Samples
When the person put his fingerprint on the sensor, the image of that finger print is captured in that sensor. This sample can be extracted later on for any kind of processing.
In order to process the fingerprint, we have used the FVC2002 finger print image database. FVC2002 was the Second International Competition for Fingerprint Verification Algorithms

The details of the fingerprint database used by us are as follows:

Sensor Type	Image Size	Size of Set	Resolution
Optical Sensor: "TouchView II" by Identix	388x374 (142 k pixels)	10 users x 8 fingerprints per user	500 dpi

II. Processing of Fingerprint Sample

Before sending the fingerprint sample there is need to process and encrypt it. The processing of fingerprint is done in following phases:

- Loading of image
- Image Enhancement
- Binarization
- Image Thinning
- Minutiae Marking and Extraction

Loading Fingerprint image. The fingerprint images are used to be extracted from sensor and saved in database from which any random sample is selected for processing. Shown in figure 1.

Image Enhancement. Before, we actually start with the process the fingerprint sample undergoes a process of Image Enhancement in which the image is made clearer for smooth functioning of further operations. This process includes judging the quality of sensor, enhancing the contrast between ridges and valleys and connecting light and almost disappeared lines.

Fingerprint Image Binarization. In this method, the sample fingerprint image is divided into certain blocks, say (16X16). The intensity value of these blocks is analyzed and accordingly the pixel value is marked as 1 if intensity value of that pixel is greater than mean intensity of that block and 0 otherwise. Shown in figure 3.

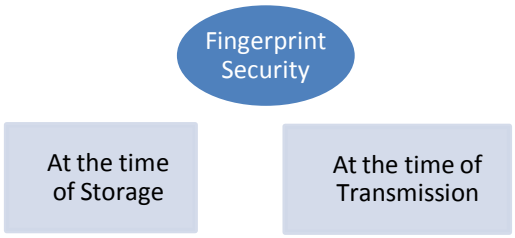
Fingerprint Ridge Thinning. As the name suggest, this is the process of thinning in which thickness of each line is reduced to a single line. The image is reduced to the level where no further removal of pixels is possible. Shown in figure 4.

Minutia Marking. In this step all the minutiae points are marked based on number of bifurcations and ridge end found. Shown in figure 5.

 ISSN: 2321-8169
1052 - 1055

III. Security of Fingerprint Sample

Apart from just capturing and storing the fingerprint sample, another aspect associated to it is "Secure Transmission". Only capturing and storing the sample in database will not solve the purpose. At times, there could be a need to send the sample. In these types of situations, it becomes very important to make the sample secure enough to be hacked on the transmission link.
In this paper we are only emphasizing upon securing a finger print at the time of transmission.



Whenever we talk about secure transmission of an image, in the first instance the concept of steganography, and encryption come to our mind. We will discuss these techniques in brief.
Steganography is the technique of hiding the confidential information in such a manner that intruder do not even get to know that something is actually hidden behind the image. So that any hacker doesn't even come to know that something is fishy.
This technique is often confused with encryption. Both are the methods of hiding the information.
Encryption is the process of encoding messages or information in such a way that only authorized parties can read it.

IV. Idea behind Algorithm designed

As we know that a finger print of a person can be identified on the number and location of minutiae points of the fingerprint. We capture a fingerprint sample from the input sensor. The minutiae points present on the boundaries of the image are not very clear and do not make any major change to the result. Therefore image has been cropped from all sides to remove less significant points, which reduce the number of pixels to 25-30% ppprox. Thereby reducing the processing time. Shown in figure 2.

After this we find the position of minutiae points in the form of (x,y) coordinates. Shown in figure 6 and 7. Then we try to tamper the these (x,y) coordinates by interchanging the values, calculating quotient and reversing the sequence of bits, which is discussed in detail in algorithm discussed later.
After following the operations, output is in the form of array which could be sent over the network. Shown in figure 8. At the destination the algorithm is reversed through which original values of (x,y) coordinates are recovered. These coordinates can then be plotted over and original minutiae points can be extracted.

V. Proposed Algorithm

Encryption

Firstly the values of x and y are interchanged, here x and y are coordinates of minutiae points.

```
temp=x
x=y
y=temp
```

After this, the values of x and y are divided among themselves which reduces the value to a very small number and hence contribute to fast calculations.

As we know that floating point numbers are calculated fast as compared to integer values, therefore smaller pixel value (out of x and y) is kept as dividend and larger pixel value is kept as divisor.
i.e. pixel with smaller value is divided by pixel with larger value.

Calculate quotient for all the pixel values as:

```
quo=y/x
set flag=1          } if x>y

quo=x/y             } if y>x
set flag as 0
```

Also instead of applying calculations on both x and y pixel values, only one value 'quo' is taken into consideration which leads to time saving. The quotient is converted to binary.

```
bin_quo= convert_to_binary(quo)
```

Alternate pixel values are flipped from left to right. Take a counter variable and check its value as even and odd using mod operator and flip the bit sequence accordingly.

```
final1=fliplr(bin_quo)  } if ctr is even
final1=bin_quo;         } otherwise
```

Decryption

```
rev_flip=fliplr(final1);  } if ctr is even
rev_flip=final1;          } otherwise
```

Extracting quo by converting back to decimal

```
quo= convert_to_decimal (bin_quo)
```

getting the values of x and y:

```
org_x1=y1./org_quo1;   } if flag(i)==1
org_y1=x1.*org_quo1;
```

org_x1=org_quo1.*y1; otherwise
org_y1=x1./org_quo1;

These values are again interchanges by using temp and original values are extracted.

At the time of sending data the code for encryption will be used. On the other end, at the time of decryption reverse process will be implemented.

VI. Implementation environment

The algorithm is implemented using MatLab Starter Application on Microsoft Windows 7 (32 bit operating system). The experiment was performed on Intel(R) Pentium(R) Dual CPU E2180 @2.00GHz and Ram of 2.00 GB.

VII. Finding and Conclusion

It has been observed that the during encryption the pixel values of x and y coordinates are changed to a single array which is sent over the network.
At the time of decryption, by following the reverse process original pixel values are retrieved.

VIII. References

Books

[1] A. Ross, K. Nandakumar, and A. K. Jain, Handbook of Multibiometrics. Springer, 2006.

[2] J. Wayman, A. Jain, D. Maltoni, D. Maio (Eds.), Biometric Systems: Technology, Design and Performance Evaluation, Springer, 2005.

Journals

[3] Feng Hao, R. Anderson, and J. Daugman. Combining crypto with biomet-rics effectively. Computers, IEEE Transactions on, 55(9):1081–1088, Sept. 2006.

[4] A. K. Jain, K. Nandakumar, and A. Nagar. Biometric template security. EURASIP Journal on Advances in Signal Processing,, Special Issue on Ad-vanced Signal Processing and Pattern Recognition Methods for Biometrics, 2008.

[5] A. Rahman, M. S. Azad and F. Anwar, "An Efficient Technique for Human Verification using Finger Stripe Geometry," International Journal of Soft Computing, vol. 2(3), pp. 445-449, 2007.

[6] Hide & Seek: An Introduction to Steganography: Niels Provos and Peter Honeyman, IEEE Security & Privacy Magazine, May/June 2003. http://niels.xtdnet.nl/papers/practical.pdf

Conference Proceedings

[7] M. Kaur, M. Singh, P.S. Sandhu, "Fingerprint Verification system using Minutiae Verification Technique", Proceedings of world Academy of Science, Engineering and Technology, vol. 36, 2008.

[8] E.-C. Chang, R. Shen, and F. W. Teo, "Finding the original point set hidden among chaff," in Proc. the 2006 ACM Symposium on Information, computer and communications security, June 2009.

[9] U. Uludag, S. Pankanti, S. Prabhakar and A.K. Jain. "Biometric Cryptosystems: Issues and Challenges". Proceedings of the IEEE. 92(6):948-960. 2004

[10] Kulwinder Singh, Kiranbir Kaur,Ashok Sardana,"Fingerprint Feature Extraction"in IJCST Vol. 2,Issue 3,September 2011.

[11] Mary Lourde R and Dushyant Khosla "Fingerprint Identification in Biometric Security Systems"in International Journal of Computer and Electrical Engineering, Vol.2, No.5,October,2010.

Figure 1: Loading the Finger print image

Figure 2: Cropped image

Figure 3: Image Binarization

ISSN: 2321-8169
1052 - 1055

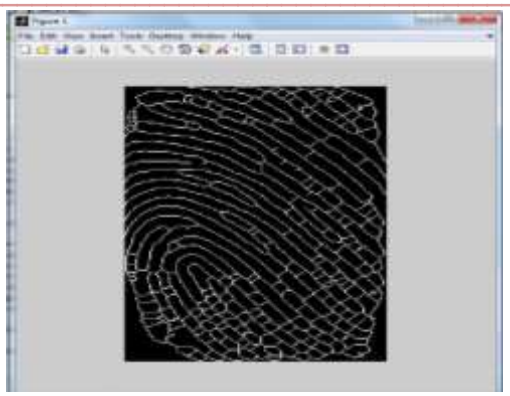


Figure 4: Image Thinning

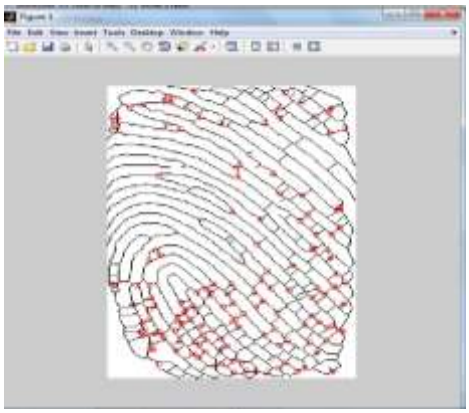


Figure 5: Minutiae Extraction

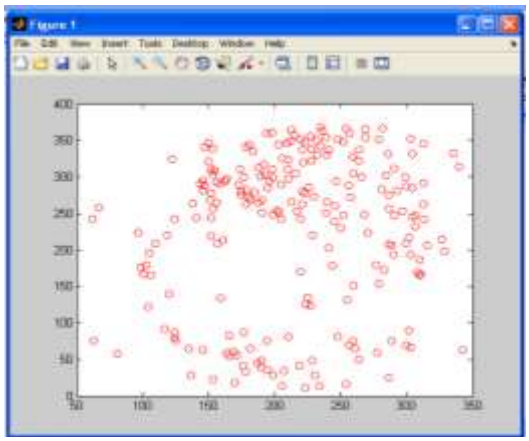


Figure 6: Finding (x,y) coordinates

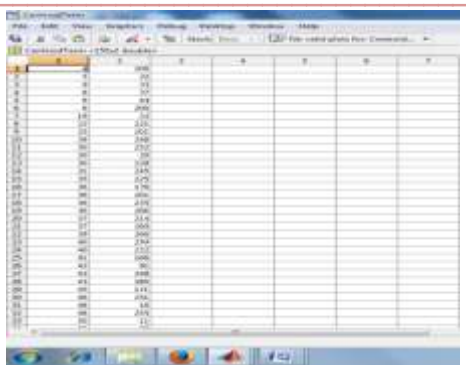


Figure 7: List of (x,y)coordinates

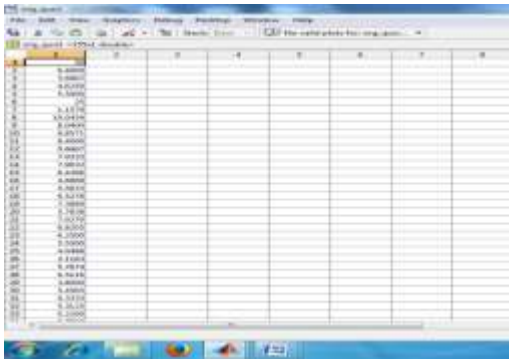


Figure 8: Quotient Calculated