EAI Endorsed Transactions
on Pervasive Health and Technology Research Article **EAI.EU**

Privacy Preserving Fingerprint Authentication at the Cloud Server for eHealth Services

S Rajarajan[1],*, R Venkata Kausik[1], M Sree Charan[1], PLK. Priyadarsini[1]

[1]SASTRA Deemed to be University, India

Abstract

Background: Restricting entry and access to highly sensitive systems like e-Health services is vital to protect the privacy and confidentiality of customers. User authentication is implemented predominantly done using Bio-metric schemes. But there is a challenge in securely storing the fingerprint templates. Since fingerprints are permanent, fingerprints getting stolen will be a serious issue.
Objective: The objective of this paper is to design a scheme to carry out the fingerprint match at the cloud server that will not compromise the fingerprints.
Methods: When users input their fingerprint, a unique key is generated and it is sent to the server for verification.
Results: We evaluated our proposed scheme and ascertained that it generates unique Fingerprint Identification Number for different fingerprints and the false rejection rate of our proposed scheme is within acceptable limit.
Conclusion: The proposed scheme overcomes the security threats to the stored fingerprints of users

Keywords: Biometrics, Fingerprints, Privacy-preserving, Cancellable-biometrics, Cloud, Image feature extraction, SURF

Received on 13 April 2019, accepted on 07 May 2019, published on 15 May 2019

doi: 10.4108/eai.13-7-2018.162688

1. Introduction

User authentication plays an important role in several applications. Particularly when the applications involve handling money such as Internet or mobile banking, authenticating the actual user is very important. Earlier passwords and PIN numbers were used as secret values to recognize authenticated users. But they are difficult to remember [1]. Biometric schemes offer superior convenience to users since they need not be remembered explicitly. But unlike passwords, biometric values cannot be changed when there is fear of leakage happened. To ensure protection of Biometric fields, cancellable biometric schemes are proposed. In those schemes the biometric data is distorted using some algorithm and then it is stored. During verification the distorted data is restored and matched against the user's biometric data. Another option to accomplish biometric authentication without the danger of losing the data is generating a unique number out of the biometric feed and storing only that instead of the actual biometric data. Cloud storage offers enormous space for storing data of organizations [2]. In this paper we are presenting our scheme that generates a unique number for a fingerprint template using a feature extraction scheme called SURF (Speeded Up Robust Features) scheme. The generated FINs are stored on cloud storage. Every time user wants to access his eHealth system, he has to feed his fingerprint which is instantly converted into a FIN, forwarded to the cloud server and it is matched against the stored FIN. Our implementation on MatLab provided satisfactory results that we could successfully authenticate users using the key value generated through our scheme.

1.1 Biometric Crypto Systems

*Corresponding author. Email:srajarajan@cse.sastra.edu

Because of the prevalence of networking and Internet usages, the volume of information being share communicated through communication channels is huge. There are also significant number of security attacks on the communications. Cryptography provides the means to protect the confidentiality of the information being shared on a public network like Internet. It transforms the actual message to some random text with the help of an algorithm and a key. When both sender and receiver use the same key it is termed as symmetric cryptography and when different keys are used for encryption and it is called asymmetric cryptography or public key cryptography. One of the challenges in cryptography implementation is the management of keys. Keys have to be secretly communicated to the sender and the sender must remember the key to use it for encrypting the messages. The amalgamation of Biometrics and cryptography offers the solution to this problem. Instead of remembering any key, the user just has to feed his biometric value through a biometric sensor. This biometric value is then transformed into a key with the help of an algorithm and it is applied on the encryption. There are several schemes developed to convert a biometric data into a key that is unique with respect to the biometric data. A major obstacle in the biometric key generation scheme is the possible variation in the biometric value due to environmental factors.

1.2 Authentication in eHealth Services

Authentication is crucial for eHealth services. Digital technologies are widely adopted for providing authentication. As more and more health services are getting moved to the cloud, we need more innovative and secured authentication schemes needs to be designed to meet the unique challenges of the cloud eHealth services [3]. eHealth services are also getting extended to mobile based deployments. But that increases the security vulnerabilities [4].

1.3 SURF Algorithm

SURF is one of the effective feature extraction algorithm for object recognition [5]. SURF makes use of integer approximation of the Hessian blob detector. Square shaped filters which are an approximation of Gaussian smoothing is used in SURF.

$$S(x, y) = \sum_{i=0}^{x} \sum_{j=0}^{y} I(i, j)$$

Since Hessian matrix based blob detector is used, the determinant enables identifying the local change surrounding the point and points where determinant is higher. Considering a point p=(x, y) in an image I, the Hessian matrix H(p, σ) at point p and scale σ, is:

$$H(p, \sigma) = \begin{pmatrix} L_{xx}(p, \sigma) & L_{xy}(p, \sigma) \\ L_{yx}(p, \sigma) & L_{yy}(p, \sigma) \end{pmatrix}$$

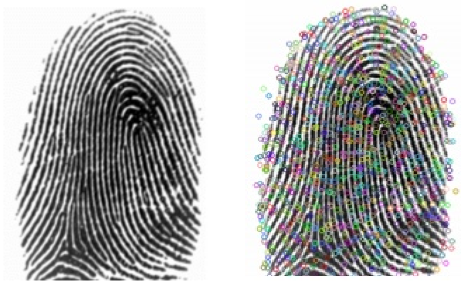

Figure 1: (a). Raw fingerprint image (b). Features extracted using SURF

In section 2 some of the related works found in literature are summarized, section 3 introduces the proposed approach, section 4 presents the experimental results of the sample implementation and section 5 concludes the paper.

2. Related Works

Lot of research has been done on implementing privacy-preserving biometric identification. The primary objective of privacy-preserving biometric identification is to ensure that the user's biometric data is matched against the stored biometric value while ensuring that the biometric stored data is never revealed [6]. Some of the implementations are based on the homomorphic techniques. A light-weight privacy preserving biometric scheme for the energy constrained IOT devices is developed based on block logic operation in [7]. Teoh and Kim [8] proposed a dynamic randomized quantization transformation that binarizes the fingerprint features retrieved through a multichannel Gabor filter. A scheme based on behavioural biometrics with fuzzy commitment is proposed in [9]. In [10], a fuzzy commitment scheme for iris data is proposed. Reed Solomon code is used for tackling errors. A scheme to protect speech template at the databases using ECC is presented in [11]. The scheme makes use of lesser key size without compromising on security. There are numerous research works carried out in biometric crypto systems. Soutar et al proposed one of the earliest biometric crypto system. A new innovative scheme called fuzzy vector signature which minimizes the usage of bandwidth for conducting the biometric authentication is proposed. [12]. Following their model, Hao et al. applied the FCS scheme to iris pattern to generate cryptographic key. Zhang et al. incorporate reed-solomon code to build a crypto system with the help of FCS and iris pattern [13]. Nandakumar et al. have successfully implemented the fuzzy vault based crypto system with the help of fingerprints [14]. Feng et al. demonstrated a biometric cryptosystem using facial image [15]. A privacy preserving authentication scheme for eHealth networks was proposed in [16]. Their scheme replaces the centralized authentication of users by one-to-one authentication involving the users directly.

Most of the existing schemes were designed to generate a key from the Biometric data which could be used as encryption key of a cryptographic system. Whereas our scheme is an authentication scheme for a eHealth system which generates unique alpha-numeric passwords based on the fingerprints of the users.

3. Proposed Approach

Our proposed scheme makes use of image feature extraction algorithm to distinguish fingerprint images and to generate a unique number for each finger print image. Unlike many of the proposed biometric key generation schemes, our scheme takes the entire finger print image for generating a key, not just the minutiae of the fingerprints. The speeded up robust features (SURF) extraction algorithm is used to distinguish the fingerprint images. We implemented our proposed approach on Matlab. In this algorithm we generate unique FINs using fingerprints. The sample fingerprint datasets were downloaded from FVC2000 which contains a repository of fingerprint images [17].

Firstly, we input the images from the dataset into a variable named “dinfo” which extracts each image information from the dataset using dir command(which takes us to the specified location where the dataset exists). We loop through dinfo to access each image through its name parameter i.e we store it in a variable

(thisimage = dinfo(K).name)

We use the matlab command “imread” for reading the image, imread acquires a greyscale or colour image whose name is mentioned under the filename argument. In the absence of the file at the specified location, we are required to specify the complete path of image’s location. Now , we convert the RGB image read by imread into grayscale image using the function rgb2gray. With the help of I =rgb2gray(RGB), image is converted into a grayscale image. This is done by removing the hue and saturation details while retaining the luminance. Next, we will be detecting the SURFFeatures in the image read.

points =detectSURFFeatures(I)
returns a SURFPoints object, points..

Now we use extractFeatures function which takes two input parameters – Image and points that were detected earlier using detectSURFFeatures function. extractFeatures function returns features and valid_points of the input image.

[features,validPoints] = extractFeatures(I,points)

This function obtains the feature vectors and their respective locations from an image. We display the grayscale image using figure, imshow and “holdon” commands in matlab. Imshow takes image as input parameter. The command figure() generates a figure window using the default values. The resulting figure becomes the current figure. imshow(I) shows the grayscale image. “hold_on” function keeps plots in the current axes so that new plots attached to the axes will not overwrite existing plots. We extract the size of features vector using size function in MatLab. Size function provides a row vector that contain the length of the corresponding dimension of the feature vector. Since we get a 2 x 1 vector when we use size function , we access the element of the 2 x 1 vector which gives the count of number of rows of features vector. For example , let A be a 2x1 vector , the first element can be accessed using A(1). Here , we accessed the row count of features vector using siz_feat(1).

3.1 Key Generation

Next , we generate a random number ranging from 1 to siz_feat(1).This is achieved through randperm function in matlab.This function doesn’t repeat a random number in a given range.

p = randperm(n)

It returns a row vector consisting of the random permutation of the integers through 1 to n inclusive. Now we take the first element of the vector returned by randperm function and store it in a variable s. Next, we store first “s-1” x-coordinates, y-coordinates of the valid points in variables X and Y respectively.

X=valid_points.Location(1:s-1,1);
Y=valid_points.Location(1:s-1,2);

Next we find the angle between two consecutive valid_points in degrees using “atan” function and store it in a variable “a”. We find the length between two consecutive points and store it in a variable “l”, using “norm” function. In order to obtain an integer we will round it off using “round” function (for both a and l).
n =norm(v) returns the Euclidean norm of vector v. This norm is also called the 2-norm, vector magnitude, or Euclidean length. Now ,we apply bitxor between angle and length which we calculated in the above step. The inputs given in bitxor are explicitly converted from “double” to “int64”.

k3=bitxor(int64(l),int64(a),'int64');
bitxor(X,Y) returns the bit-wise XOR.

The two arguments must be in either same or compatible sizes. The value obtained frm bitxor is added to a variable ”f” every time the loop executes. Also we add the value at “c”th row of features vector to a variable “sum”(where “c” is the iterator which ranges from 1 to “s-2”).

```
f=f+k3;
sum=sum + features(c);
```

Next we multiply the variables “sum” and “f” obtained above and convert them explicitly to double data type. Now we convert this double value into int16 data type using typecast.We now convert this result into binary using dec2bin function in MatLab.

We could get the character vector of a binary represented value using the dec2bin(d)function. The argument to dec2bin must be a nonnegative integer. If the value passed is larger than the return value of flintmax, then dec2bin will not provide an exact representation.

Next we consider the binary string obtained from dec2bin function as 8 bit characters(we convert the string into 8 bit characters using unit8 function) and pass the result to sha256 hashing algorithm which coverts it into a vector of decimal values.

```
sha256hasher=System.Security.Cryptography.SHA256Managed;
sha256 = uint8(sha256hasher.ComputeHash(uint8(val)));
```

Now we convert the decimal-valued vector returned by hashing algorithm into hexadecimal-valued character vector using dec2hex function. dec2hex(X) is used to obtain the hexadecimal representation in a character vector format. X must be a positive integer. If X is larger than the value that flintmax returns, then function might not give an exact representation.

We store the hexdecimal character vector returned by dec2hex function in a variable “x”. From “x” , we form an FIN (Fingerprint Identification Number) of length 6. The first character of the FIN is the first character of first element of x. Second character of FIN is the first character of last element of x. Third character of FIN is the first character of centre element of x. Fourth character of FIN is the second character of first element of x. Fifth character of FIN is the second character of last element of x. Sixth character of FIN is the second character of centre element of x.

4. Experimental Results

We implemented our testing system on MatLab. The fingerprint images that we used for conducting this evaluation were obtained from FVC2000 which contains the database of fingerprints [18]. We made use of 165 fingerprints. A sample set of our experimental result is presented in Table 1. The results confirms that our proposed scheme is able to distinguish the fingerprints and generate unique FINs for each fingerprint Table 2 presents the accuracy of the verification of enrolled users whose FINs are stored at the database. We can see from the table that the false rejection rate of the proposed scheme is within the acceptable range. The time taken is also less.

The major limitation of our proposed scheme is that it is vulnerable to variation in fingerprints. In a conventional fingerprint authentication system the matching algorithm could tolerate some amount of variation in the fingerprint inputs without rejecting the authentication. But since our proposed scheme makes use of feature extraction algorithm, it expects the fingerprint data to be precisely the same as the one enrolled earlier.

Table 1: Sample FINs generated from Fingerprints

Finger Print Image	FIN Generated
	62C150
	80C14B
	DD9B00
	8B2DD6
	98728F

Table 2: Performance of verification of Biometric data

	Verification
No. of samples	165
False rejections	11
Accuracy	93%
Average time (in seconds)	0.73

5. Conclusion

Today the usage of fingerprints for mobile phone authentication and unlocking is very common. But extending fingerprint based authentication to authentications with remote serves is a big challenge due to the vulnerability of fingerprints. In this paper we have designed a cloud based fingerprint authentication scheme to convert a fingerprint into a unique FIN and conducting the verification at the cloud. But it is impossible to regenerate the fingerprint with the help of the FIN. We evaluated the proposed scheme and ascertained the validity of the proposed scheme. As a future work we wish to tackle the variations in the fingerprints. Another potential future work is to experiment the proposed scheme on iris data. Since iris value remains same without variation, it might be a better candidate than the fingerprints.

References

[1] Journal Article: Rajarajan, S., & Priyadarsini, P. (2019). UTP: A Novel PIN Number Based User Authentication Scheme. *International Arab Journal of Information Technology*, *16*(5), 904-913.

[2] Jangiti, S., Sriram, E., Jayaraman, R., Ramprasad, H., & Sriram, V. S. (2019). Resource ratio based virtual machine placement in heterogeneous cloud data centres. *Sādhanā*, *44*(12), 236.

[3] Journal article: Sultan, N. (2014). Making use of cloud computing for healthcare provision: Opportunities and challenges. *International Journal of Information Management*, *34*(2), 177-184.

[4] Conference: Çoban, Ç., & Tüysüz, M. F. (2019, September). E-Health and Privacy: Risks, Opportunities and Solutions. In *2019 4th International Conference on Computer Science and Engineering (UBMK)* (pp. 554-559). IEEE.

[5] Journal article: Awad, A. I. (2013). Fingerprint local invariant feature extraction on GPU with CUDA. *Informatica*, *37*(3).

[6] Journal article: Liu, C., Hu, X., Zhang, Q., Wei, J., & Liu, W. (2019). An Efficient Biometric Identification in Cloud Computing With Enhanced Privacy Security. *IEEE Access*, *7*, 105363-105375.

[7] Journal: Yang, W., Wang, S., Zheng, G., Yang, J., & Valli, C. (2019). A privacy-preserving lightweight biometric system for internet of things security. *IEEE Communications Magazine*, *57*(3), 84-89.

[8] Journal article: Teoh, A. B. J., & Kim, J. (2007). Secure biometric template protection in fuzzy commitment scheme. *IEICE Electronics Express*, *4*(23), 724-730.

[9] Journal article: Maiorana, E., & Campisi, P. (2009). Fuzzy commitment for function based signature template protection. *IEEE Signal Processing Letters*, *17*(3), 249-252.

[10] Conference: Rathgeb, C., & Uhl, A. (2009, June). Systematic construction of iris-based fuzzy commitment schemes. In *International Conference on Biometrics* (pp. 940-949). Springer, Berlin, Heidelberg.

[11] Journal: Sasikaladevi, N., Geetha, K., Revathi, A., Mahalakshmi, N., & Archana, N. (2019). SCAN-speech biometric template protection based on genus-2 hyper elliptic curve. *Multimedia Tools and Applications*, 1-23.

[12] Journal: Seo, M., Hwang, J. Y., Lee, D. H., Kim, S., Kim, S. H., & Park, J. H. (2019). Fuzzy Vector Signature and Its Application to Privacy-Preserving Authentication. *IEEE Access*.

[13] Conference: Zhang, L., Sun, Z., Tan, T., Hu, S. (2009). Robust biometric key extraction based on iris cryptosystem. In *Proceedings of The 3rd international conference of biometrics, ICB'09* (pp. 1060–1069).

[14] Journal article: Nandakumar, K., Jain, A. K., & Pankanti, S. (2007). Fingerprint-based fuzzy vault: Implementation and performance. *IEEE transactions on information forensics and security*, *2*(4), 744-757.

[15] Journal article: Feng, Y. C., Yuen, P. C., & Jain, A. K. (2009). A hybrid approach for generating secure and discriminating face template. *IEEE transactions on information forensics and security*, *5*(1), 103-117.

[16] Guo, L., Zhang, C., Sun, J., & Fang, Y. (2012, June). Paas: A privacy-preserving attribute-based authentication system for ehealth networks. In 2012 IEEE 32nd International Conference on Distributed Computing Systems (pp. 224-233). IEEE.

[17] Web site: http://bias.csr.unibo.it/fvc2000/

[18] Journal article: Maio, D., Maltoni, D., Cappelli, R., Wayman, J. L., & Jain, A. K. (2002). FVC2000: Fingerprint verification competition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *24*(3), 402-412.