2016 IEEE 14th Intl Conf on Dependable, Autonomic and Secure Computing, 14th Intl Conf on Pervasive Intelligence and Computing, 2nd Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress

Privacy-Preserving Palm Print Authentication using Homomorphic Encryption

Jong-Hyuk Im, JinChun Choi, DaeHun Nyang and Mun-Kyu Lee
Department of Computer and Information Engineering
INHA University
Incheon 22212, Korea
imjhyuk@gmail.com, noodlejin@isrl.kr, {nyang, mklee}@inha.ac.kr

Abstract— **Biometric verification systems have security issues regarding the storage of biometric data in that a user's biometric features cannot be changed into other ones even when a system is compromised. To address this issue, it may be safe to store the biometrics data on a reliable remote server instead of storing them in a local device. However, this approach may raise a privacy issue. In this paper, we propose a biometric verification system where the biometric data are stored in a remote server in an encrypted form and the similarity of the user input to the registered biometric data is computed in an encrypted domain using a homomorphic encryption. We evaluated the performance of the proposed system through an implementation on an Android-based smartphone and an i7-based server.**

Keywords- palm print; homomorphic encryption; biometrics; authentication; mobile device

I. Introduction

Recent advances in mobile technologies enabled people to do various jobs using their mobile devices. These jobs include dealing with private and financial data. In order to protect these critical data, many mobile device manufacturers and software companies use biometrics such as fingerprint, palm print, and iris for user authentication. However, biometric verification has some security issues which were not the case for traditional passwords. That is, leakage of biometric data may result in a critical privacy problem, because the biometric features are unchangeable and unique to each individual. A possible countermeasure is to store biometric data on a reliable remote server. In this case, however, the remote server can learn user's biometric data, which raises another privacy issue. In addition, if the server is compromised by an attacker, massive biometric data of multiple users may be leaked.

In this paper, we propose a solution to this problem by presenting a biometric verification system where the biometric data are stored in a remote server in an encrypted form and the biometric input by the claimed user is compared with the registered biometric data in a ciphertext domain using a homomorphic encryption algorithm. Homomorphic encryption is an encryption scheme where arithmetic operations can be performed on ciphertexts. It is frequently considered in the context of privacy-preserving big data analysis and secure cloud computing [1]. If the operation is addition (multiplication), we call the encryption scheme an additive (multiplicative) homomorphic encryption scheme. A fully homomorphic encryption (FHE) scheme is a scheme which supports an arbitrary number of homomorphic additions and multiplications on ciphertexts [2-4]. Whereas FHE schemes are very useful, they are significantly less efficient than additive homomorphic encryption (AHE) schemes such as the Paillier scheme [5]. Although AHE schemes have a clear limit that they only support homomorphic additions, a recently proposed transformation method [6] can transform an AHE to a scheme where degree-2 functions can be evaluated, i.e., it becomes possible to do a limited number of multiplications. We combine this technique with the Paillier AHE to enable homomorphic comparison of encrypted biometric feature vectors.

II. Preliminaries

A. Biometric Verification

Biometric verification is an authentication process using uniquely identifiable biological characteristics such as human face, fingerprint, palm print, vein, etc. Also footstep and voice can be used for authentication [7]. To authenticate a user using the above biometric data, various linear classification techniques such as Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA) [8], and Random Projection (RP) [9] are being used. In these methods, projection matrices are used to extract a feature vector from the user's biometric information. RP does not need any information on a specific user to create a projection matrix, which is in contrast to PCA and LDA, but it guarantees similar performance to that of LDA. In order to compare the feature vectors extracted from many users' biometric data, distance metrics such as Manhattan distance and Euclidean distance are frequently used [10]. In this paper, we are interested in palm print authentication among various biometric verification schemes.

B. Transformation for Multiplicative Homomorphism

To compute the Euclidean distance between two feature vectors, multiplications and additions between vector elements are necessary. Recently in [6], a transform function was proposed which takes as input an additive homomorphic encryption (AHE) scheme and builds a multiplicative

978-1-5090-4065-0/16 $31.00 © 2016 IEEE
DOI 10.1109/DASC-PICom-DataCom-CyberSciTec.2016.150

homomorphic encryption scheme where homomorphic evaluation of degree-2 functions is possible. Most additive homomorphic encryption schemes are based on the same framework where plaintext messages are dealt with in an exponent. Namely, they are encryption schemes based on a discrete log trapdoor modulo a large integer. Fig. 1 is the details of the transform function in [6].

$\boldsymbol{Input\ HE\ scheme}$:

Let $\widehat{\boldsymbol{HE}} = (\widehat{\boldsymbol{KeyGen}}, \widehat{\boldsymbol{Enc}}, \widehat{\boldsymbol{Eval}}, \widehat{\boldsymbol{Dec}})$ be an AHE scheme.

Let $\boldsymbol{M}, \widehat{\boldsymbol{C}}$ be the message and ciphertext spaces of $\widehat{\boldsymbol{HE}}$, respectively.

$\boldsymbol{Key\ Generation}$:

$\widehat{\boldsymbol{KeyGen}}$ is run to get $(\boldsymbol{pk}, \boldsymbol{sk})$, and output $(\boldsymbol{pk}, \boldsymbol{sk})$

$\boldsymbol{Encryption}$:

Given a message $\boldsymbol{m} \in \boldsymbol{M}$, output $\widehat{\boldsymbol{C}} = \widehat{\boldsymbol{Enc}}(\boldsymbol{pk}, \boldsymbol{m})$

$\boldsymbol{Add_1}$:

Given two ciphertexts $\boldsymbol{C_1}, \boldsymbol{C_2} \in \widehat{\boldsymbol{C}}$,

output $\boldsymbol{C} = \boldsymbol{C_1} \boxplus \boldsymbol{C_2} \in \widehat{\boldsymbol{C}}$, where $\boxplus$ is homomorphic addition, which is actually multiplication of two ciphertexts.

$\boldsymbol{Mult}$:

Given two ciphertexts $\boldsymbol{C_1^{'}}, \boldsymbol{C_2^{'}} \in \widehat{\boldsymbol{C}}$,

choose (at random) $\boldsymbol{a_1}, \boldsymbol{a_2} \in \boldsymbol{M}$,

compute $\boldsymbol{C_i^{''}} = \widehat{\boldsymbol{Enc}}(\boldsymbol{pk}, -\boldsymbol{a_i})$, and set $\boldsymbol{\beta_i} =$ $\boldsymbol{Add_1}(\boldsymbol{C_i^{'}}, \boldsymbol{C_i^{''}})(\boldsymbol{i} = \boldsymbol{1}, \boldsymbol{2})$

Finally, output $\boldsymbol{C} = (\boldsymbol{\alpha}, \boldsymbol{\beta})$, where

$\boldsymbol{\alpha} = \widehat{\boldsymbol{Enc}}(\boldsymbol{pk}, \boldsymbol{a_1} \cdot \boldsymbol{a_2}) \boxplus \boldsymbol{a_1} \cdot \boldsymbol{\beta_2} \boxplus \boldsymbol{a_2} \cdot \boldsymbol{\beta_1}$ and $\boldsymbol{\beta} = (\boldsymbol{\beta_1}, \boldsymbol{\beta_2})^{\top}$.

$\boldsymbol{Add_2}$:

Given two ciphertexts $\boldsymbol{C_1}, \boldsymbol{C_2}$ $\left(\boldsymbol{C_i} = (\boldsymbol{\alpha_i}, \boldsymbol{\beta_i})\right)$,

output $\boldsymbol{\alpha} = \boldsymbol{\alpha_1} \boxplus \boldsymbol{\alpha_2}, \boldsymbol{\beta} = [\boldsymbol{\beta_1}, \boldsymbol{\beta_2}]$ (concatenation).

$\boldsymbol{cMult}$:

Given a constant $\boldsymbol{c} \in \boldsymbol{M}$, a ciphertext $\boldsymbol{C}$

If $\boldsymbol{C} = \boldsymbol{\beta} \in \widehat{\boldsymbol{C}}$, then $\boldsymbol{C}^{'} = \boldsymbol{c} \cdot \boldsymbol{\beta} \in \widehat{\boldsymbol{C}}$

If $\boldsymbol{C} = (\boldsymbol{\alpha}, \boldsymbol{\beta})$, where $\boldsymbol{\beta} = \left[(\boldsymbol{\beta_{1,1}}, \boldsymbol{\beta_{2,1}})^{\top}, \ldots, (\boldsymbol{\beta_{1,l}}, \boldsymbol{\beta_{2,l}})^{\top}\right]$,

then, output $\boldsymbol{\alpha}^{'} = \boldsymbol{c} \cdot \boldsymbol{\alpha}, \boldsymbol{\beta}^{'} = \left[(\boldsymbol{c} \cdot \boldsymbol{\beta_{1,1}}, \boldsymbol{\beta_{2,1}})^{\top}, \ldots, (\boldsymbol{c} \cdot \boldsymbol{\beta_{1,l}}, \boldsymbol{\beta_{2,l}})^{\top}\right]$

$\boldsymbol{Decryption}$:

If $\boldsymbol{C} = \boldsymbol{\beta} \in \widehat{\boldsymbol{C}}$, output $\widehat{\boldsymbol{Dec}}(\boldsymbol{sk}, \boldsymbol{\beta})$

If $\boldsymbol{C} = (\boldsymbol{\alpha}, \boldsymbol{\beta})$, where $\boldsymbol{\beta} = \left[(\boldsymbol{\beta_{1,1}}, \boldsymbol{\beta_{2,1}})^{\top}, \ldots, (\boldsymbol{\beta_{1,l}}, \boldsymbol{\beta_{2,l}})^{\top}\right]$,

output $\boldsymbol{m} = \widehat{\boldsymbol{Dec}}(\boldsymbol{\alpha}) + \left(\sum_{i=1}^{l} \widehat{\boldsymbol{Dec}}(\boldsymbol{sk}, \boldsymbol{\beta_{1,i}}) \cdot \widehat{\boldsymbol{Dec}}(\boldsymbol{sk}, \boldsymbol{\beta_{2,i}})\right)$.

Figure 1. Description of transform function in [6].

C. Paillier Cryptosystem

The above framework in [6] can be applied to almost any AHE. In this paper, we chose the Paillier scheme [5] as the underlying AHE scheme. The scheme in [5] is described in Fig. 2.

Parameters:

Select prime numbers $\boldsymbol{p}, \boldsymbol{q}$.

Calculate $\boldsymbol{n} = \boldsymbol{pq}$ and $\boldsymbol{\lambda} = \boldsymbol{lcm}(\boldsymbol{p} - \boldsymbol{1}, \boldsymbol{q} - \boldsymbol{1})$.

Let $\boldsymbol{g} \in \mathbb{Z}_{n^2}^{*}$ be a generator.

Public key: $\boldsymbol{n}, \boldsymbol{g}$

Private key: $\boldsymbol{p}, \boldsymbol{q}$ (or equivalently $\boldsymbol{\lambda}$)

Encryption:

Given a message $\boldsymbol{m} < \boldsymbol{n}$, select a random $\boldsymbol{r} \in \boldsymbol{Z}_{\boldsymbol{n}}^{*}$.

output $\boldsymbol{c} = \boldsymbol{g}^{\boldsymbol{m}}\boldsymbol{r}^{\boldsymbol{n}}\ \boldsymbol{mod}\ \boldsymbol{n}^{\boldsymbol{2}}$.

Decryption:

Given a ciphertext $\boldsymbol{c} < \boldsymbol{n}^{\boldsymbol{2}}$,

output $\boldsymbol{m} = \left(\frac{L(c^{\lambda}\ mod\ n^2)}{L(g^{\lambda}\ mod\ n^2)}\right)\ \boldsymbol{mod}\ \boldsymbol{n}$, where $\boldsymbol{L}(\boldsymbol{u}) = \frac{\boldsymbol{u}-\boldsymbol{1}}{\boldsymbol{n}}$.

Figure 2. Description of the Paillier scheme [5].

Homomorphic addition of two ciphertexts $c_1 = Enc(m_1)$ and $c_2 = Enc(m_2)$, where m_1 and m_2 are plaintexts, and Enc is encryption, is done by multiplying two ciphertexts:

$c_1 \boxplus c_2 \equiv g^{m_1}r_1^n \cdot g^{m_2}r_2^n \equiv g^{m_1+m_2} \cdot (r_1r_2)^n\ mod\ n^2$

Similarly, multiplication of a ciphertext c by a constant a is done as follows:

$c^a = (g^m r^n)^a = g^{ma}(r^a)^n\ mod\ n^2$.

III. Proposed Method

A. Proposed Palm Print Authentication Method

In this paper, we focus on biometric verification of a mobile device user. Most mobile devices are equipped with a built-in camera which can be used to obtain biometric data. Because a palm print can be obtained easily on a mobile device compared to other biometric data such as a fingerprint and iris, we consider palm print for biometric verification.

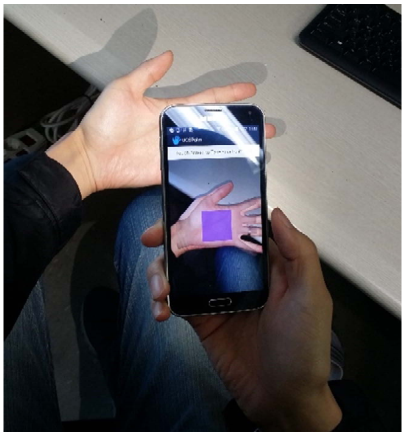

Figure 3. The palm print authentication phase. (taking a picture)

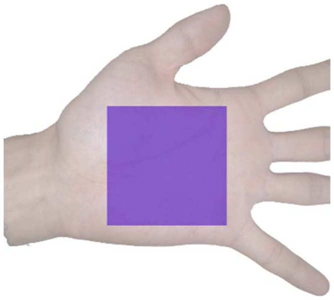

Figure 4. The guideline template for palm print authentication.

In general, the area in an image used for biometric verification is called a Region of Interest (ROI) [11]. In order to effectively extract an ROI, it is useful to provide a guideline template for a user so that the user can take a palm print picture easily [12].

In our method, we display a virtual hand as a guideline template as in Fig. 4 and ask the user to align his/her own hand to this guideline when taking a picture. (See Fig. 3.) This procedure helps the user to take a fixed region in a hand, i.e., ROI which is shown as the square region in Fig. 4, as consistently as possible.

After the mobile device captures the ROI as described above, the RP method is applied to extract a biometric feature vector. First, the resolution of a captured square region, i.e., the ROI, is reduced to $n \times n$, and the resulting image is transformed to a grayscale image. Let x be an n^2-dimensional vector generated by serializing this image. Then, an $m \times n^2$ projection matrix U is applied to obtain an m-dimensional vector $y = Ux^T$. The matrix U is calculated in the setup stage and used for all users. The elements in U were chosen according to the same distribution as [12]. The vector $y = (y_1, \ldots, y_m)$ is used as a feature vector, and user authentication is done by computing the distance of two feature vectors. As the distance metric, we use the Euclidean distance which is more appropriate for homomorphic evaluation than the Manhattan distance.

B. Privacy-Preserving Palm Print Authentication Using HE

1) Requirments and Threat Model

In our authentication system, the user's biometric data, i.e., palm prints, are stored in a remote server instead of the mobile device so that the data may be safe even when the device is compromised. To keep the user's privacy, the data are stored in an encrypted form and operations required to palm print authentication are done homomorphically over the ciphertext. Therefore, we consider three parties; user, mobile device and server. A user is a party who has original palm print data, and wants to be authenticated by his/her mobile device. A mobile device is a party which authenticates its owner using biometrics. The mobile device encrypts a user's biometric data using an HE scheme and sends its ciphertexts to a server. The private and public keys for HE are stored in the device. The server stores encrypted palm print data, and performs homomorphic distance computation. The server is assumed to be honest but curious. That is, it follows the protocol although it may try to obtain some information on the original palm print from the stored ciphertext.

2) Proposed Palm Print Authentication Protocol

The protocol consists of two phases; registration and authentication, which are shown in Fig. 5 and 6, respectively.

Registration (Fig. 5): When the user starts registration procedure, the mobile device generates a key pair and requests the user to input biometric data (palm print). The user takes a picture of his/her palm print using the camera on the device. The device extracts a feature vector y from the image using the RP method. After encrypting the elements in y with the public key, the device sends the ciphertext to the server, and the server stores it.

Authentication (Fig. 6): When the user requests authentication, the mobile device requests the user to take a picture of his/her palm print. The device extracts a feature vector from the image using the RP method. After encrypting the y' elements in with the public key, the device sends the ciphertext y' to the server. After authenticating the device via any secure authentication protocol, the server calculates the encrypted distance between y and $y'$by evaluating

$$\sum_{i=1}^{m} \boldsymbol{Mult}\left(\begin{array}{c}\boldsymbol{Add_1}\left(\widehat{\boldsymbol{Enc}}(pk, y_i^{'}), \widehat{\boldsymbol{Enc}}(pk, -y_i)\right), \\ \boldsymbol{Add_1}\left(\widehat{\boldsymbol{Enc}}(pk, y_i^{'}), \widehat{\boldsymbol{Enc}}(pk, -y_i)\right)\end{array}\right)$$

where $\sum$ is repeated application of $\boldsymbol{Add_2}$. The server sends the result to the mobile device. Finally, the mobile device decrypts this result using the private key, producing the square of Euclidean distance between y and y'i.e., $\sum_{i=1}^{m}(y' - y_i)^2$. Then, the device decides whether or not to permit the user's access according to the decrypted value.

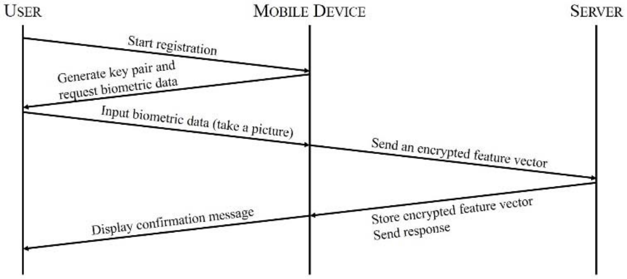
USER
MOBILE DEVICE
SERVER
Start registration
Generate key pair and request biometric data
Input biometric data (take a picture)
Send an encrypted feature vector
Display confirmation message
Store encrypted feature vector
Send response

Figure 5. The registration phase of our proposed protocol.

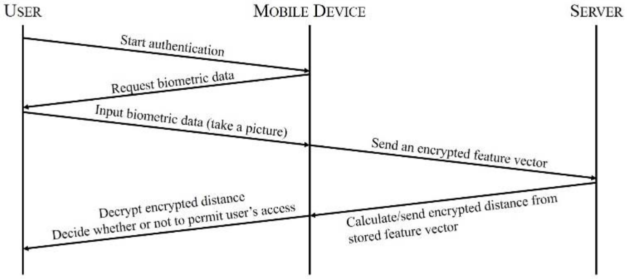
USER
MOBILE DEVICE
SERVER
Start authentication
Request biometric data
Input biometric data (take a picture)
Send an encrypted feature vector
Decrypt encrypted distance
Decide whether or not to permit user's access
Calculate/send encrypted distance from stored feature vector

Figure 6. The authentication phase of our proposed protocol.

For homomorphic distance computation of two feature vectors, we use the combination of the Paillier scheme [5] and the transformation scheme in [6]. Although there are other HE schemes where degree-2 functions can be computed, e.g., [13], the transformation scheme in [6] has better performance.

IV. Experimental Results

We implemented a homomorphic palm print authentication application on Samsung Galaxy S5 smartphone with a Qualcomm Snapdragon 801 chipset, 1,600 million pixel camera and Android 4.4.2 Kitkat OS. We also implemented a server application on an Intel® Core™ i7-4770 @ 3.40GHz CPU with 24GB RAM and a Windows 10 OS. The codes were written in Java using JDK 1.7. We used the security parameter, 1,024 bits, for the Paillier scheme [5].

For the RP method, we used a 100×30^2 projection matrix U, i.e., $m = 100, n = 30$. This parameter value was chosen to balance authentication success rate and calculation time. We designed an experiment with two goals. The first goal was to evaluate the Equal Error Rate (EER) which represents the performance of a biometric verification method. We recruited five volunteers for the experiment. Four of them were male. Every participant repeated palm print authentication ten times. An authentication was regarded as a success when $\sum_{i=1}^{m}(y_i' - y_i)^2$ was smaller than a predefined threshold. Fig. 7 shows the False Acceptance Rate (FAR) and False Rejection Rate (FRR) according to various threshold values. According to the experimental results shown in Fig. 7, EER was 15.20% when the threshold was set as 177,300. The second goal of our experiment was to measure the timings to execute the proposed protocol. We measured the timings for (M1) key pair generation for homomorphic encryption; (M2) image processing and feature vector extraction (computation of y or y_i'); (M3) encryption of y or y_i'; and (M4) decryption of square of Euclidean distance between y and y' on the mobile device, and (S1) computation of encrypted score, i.e., homomorphic evaluation of the encrypted value of $\sum_{i=1}^{m}(y_i' - y_i)^2$ on the server. Table 1 is the average time for each operation. The amount of time for other operations such as data communication and file I/O were excluded in Table 1, because they were negligible. According to Table 1, the registration phase, which is composed of (M1), (M2), and (M3), consumes about 5.91 seconds, and the authentication phase, which is composed of (M2), (M3), (S1), and (M4), consumes about 18.25 seconds. We also remark that these figures can be improved by applying various optimization techniques.

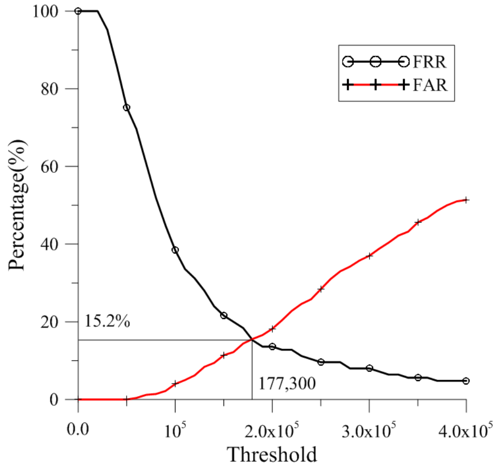

Figure 7. The result of experiment to measue FRR and FAR.

TABLE I. AVERAGE TIMINGS FOR EACH OPERATIONS (SECONDS)

Subject	***Operation***	***Average time***
Mobile Device	(M1) Key pair generation	0.33
	(M2) Image processing and feature vector extraction	2.37
	(M3) Encryption of feature vector	3.21
	(M4) Decryption	6.15
Server	(S1) Computation of encrypted score	6.52

V. DISCUSSION AND CONCLUSION

In this paper, we proposed a privacy-preserving palm print authentication scheme based on the RP method and homomorphic encryption. It will be a promising future research issue to develop an effective way to enable the mobile device to obtain a more accurate palm print image. In addition, it will also be possible to optimize the implementation of homomorphic encryption scheme, or to adopt AHE schemes other than the Paillier scheme [5].

ACKNOWLEDGMENT

This research was supported in part by Basic Science Research Program through the NRF, Korea, funded by the MoE (grant number: 2014R1A1A2058514), in part by MSIP, Korea, under the ITRC support program (IITP-2016-H8501-16-1008) supervised by the IITP, and in part by the Information Security R&D program of MSIP/IITP [B0717-16-0114, Development of Biometrics-based Key Infrastructure Technology for On-line Identification].

REFERENCES

[1] M. Naehrig, K. Lauter, and V.Vaikuntanathan, “"Can homomorphic encryption be practical?,” Proceedings of the 3rd ACM Cloud Computing Security Workshop (CCSW 2015), ACM, pp. 113-124, 2011.

[2] C. Gentry, “Fully Homomorphic Encryption Using Ideal Lattices”, STOC ’09, pp. 169-178, 2010.

[3] M. Dijk, C. Gentry, S. Halevi, and V. Vaikuntanathan, “Fully homomorphic encryption over the integers,” EUROCRYPT ’10, LNCS 6110, pp. 24-42, 2010.

[4] Z. Brakerski, C. Gentry, and V. Vaikuntanathan, “(Leveled) Fully Homomorphic Encryption without Bootstrapping,” ITCS ’12, pp. 309-325, 2012.

[5] Pascal Paillier, “Public-Key Cryptosystems Based on Composite Degree Residuosity Classes,” EUROCRYPT ’99, pp. 223-238, 1999.

[6] Dario Catalano, and Dario Fiore, “Using Linearly-Homomorphic Encryption to Evaluate Degree-2 Functions on Encrypted Data,” CCS ’15, pp. 1518-1529, 2015.

[7] Jain, Anil, Patrick Flynn, and Arun A. Ross, eds. Handbook of biometrics. Springer Science & Business Media, 2007.

[8] J. Yang, D. Zhang, J.-Y. Yang, and B. Niu, “Globally maximizing, locally minimizing: unsupervised discriminant projection with applications to face and palm biometrics.,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 29, no. 4, pp. 650–64, Apr. 2007.

[9] E. Bingham and H. Mannila, “Random projection in dimensionality reduction,” Proc. seventh ACM SIGKDD Int. Conf. Knowl. Discov. data Min. - KDD ’01, pp. 245–250, 2001.

[10] Adams Kong, David Zhang, and Mohamed Kamel, “A survey of palmprint recognition,” Pattern Recognition, vol. 42, issue 7, pp.1408-1418, 2009.

[11] C. C. Han, H. L. Cheng, C. L. Lin, and K. C. Fan, “Personal authentication using palm-print features,” Pattern Recognit., vol. 36, no. 2, pp. 371–381, 2003.

[12] S. Lee, S. Kang, D. Nyang, and K. Lee, “Effective Palm Print Authentication Guideline Image with Smart Phone,” (In Korean) The Journal of Korean Institute of Communications and Information Sciences, vol. 39, no. 11, pp. 994–999, 2014.

[13] Boneh, Dan, Eu-Jin Goh, and Kobbi Nissim. “Evaluating 2-DNF formulas on ciphertexts,” Theory of cryptography. Springer Berlin Heidelberg, pp. 325-341, 2005.