Efficient and Privacy-preserving Online Fingerprint Authentication Scheme Over Outsourced Data

Hui Zhu, *Member, IEEE,* Qing Wei, *X*iaopeng Yang, Rongxing Lu, *Senior Member, IEEE, and* *Hui Li, Member, IEEE*

Abstract—With the pervasiveness of mobile devices and the development of biometric technology, biometric identification, which can achieve individual authentication relies on personal biological or behavioral characteristics, has attracted widely considerable interest. However, privacy issues of biometric data bring out increasing concerns due to the highly sensitivity of biometric data. Aiming at this challenge, in this paper, we present a novel privacy-preserving online fingerprint authentication scheme, named e-Finga, over encrypted outsourced data. In the proposed e-Finga scheme, the user's fingerprint registered in trust authority can be outsourced to different servers with user's authorization, and secure, accurate and efficient authentication service can be provided without the leakage of fingerprint information. Specifically, an improved homomorphic encryption technology for secure Euclidean distance calculation to achieve an efficient online fingerprint matching algorithm over encrypted FingerCode data in the outsourcing scenarios. Through detailed security analysis, we show that e-Finga can resist various security threats. In addition, we implement e-Finga over a workstation with a real fingerprint database, and extensive simulation results demonstrate that the proposed e-Finga scheme can serve efficient and accurate online fingerprint authentication.

Index Terms—privacy-preserving, online authentication, fingerprint, outsource.

1 Introduction

Biometric-based identification which relies on personal biological or behavioural characteristics is receiving more and more attention as a convenient method of identifying people [1] [2]. Owing to the universality, uniqueness and permanence of biometric data [3], biometric recognition systems have been widely used in a multitude of applications without concerning about lost, stolen or forgotten, offer greater convenience than the traditional methods, e.g., PINs, passwords, ID cards. Recently, more banks are ramping up efforts to incorporate biometric technology(iris scanner, fingerprint readers, etc) into their systems [4] [5]. Considering the online payment and other access control scenarios, biometric authentication is always used as double authentication or two-factor authentication, to further certification. As the most popular biometric technology, fingerprint identification has been widely used in not only the criminal identification and police work, but also civilian applications like access control, online payment verification and driver license applications [6]–[8].

Despite the proliferation of fingerprint authentication, there are also increasing concerns over its associated privacy and legal issues, since the fingerprint data is highly sensitive and is impossible to be revoked and replaced once leaked [9], [10]. For example, if a fingerprint used as a password is compromised, it can never be used again because the fingerprint can not be changed like traditional passwords. Moreover, we might use the same fingerprint in different applications since we have a limited number of fingers, which means that a fingerprint stolen from one application could be misused in some other applications [11]. According to the CNN news, hackers stole 5.6 million government fingerprints in 2015, which means that millions of people can no longer rely on their fingerprints as security mechanism, given that smartphones and buildings are increasingly use biometric scanners to grant access [12]. Hence, appropriate security and privacy protection mechanism should be in place to defend against disclosure or misuse of fingerprint data.

In this sense, the privacy-preserving of online fingerprint authentication is still a challenging work considering the requirements of practical systems on security and efficiency. Since fingerprint identification allows some uncertainty or distortion, fingerprint data is inappropriate to be encrypted by Hash algorithm which has the extremely high "avalanche effect". To address the challenge, several typical schemes including Fuzzy Vault and BioHashing have been proposed [13] [14], which can achieve the privacy of templates during the storage and matching process. However, in the above schemes, the servers are considered trusted, and these privacy-preserving techniques will affect the accuracy of the underling identification system. Moreover, homomorphic encryption and searchable encryption technique are introduced to solve the problem [15]–[19]. These relevant schemes have high time complexities or only support basic arithmetic, cannot support the multiple complex computation online fingerprint matching service [20] [21]. Considering the outsourced scenario, some schemes [22]–[24] exploits matrix-based encryption so that it avoids heavy computation overhead while allowing the cloud to locate the best match without decryption. However, these schemes assumes the database is trusted which is not conform the actual situation.

In this paper, aiming at the above challenges, we propose a novel efficient and privacy-preserving online fingerprint authentication scheme, named e-Finga, over encrypted outsourced

This work was supported in part by National Key Research and Development Program of China(2017YFB0802201), National Natural Science Foundation of China(61672411, U1401251 and 81600574), and China 111 Project(B16037).

H. Zhu, Q. Wei, X. Yang and H. Li are with the State Key Laboratory of Integrated Services Networks, Xidian University, Xi'an710071, China(e-mail: zhuhui@xidian.edu.cn; qing9308@gmail.com; xiaopengyang2015@gmail.com;lihui@mail.xidian.edu.cn).

R. Lu is with Faculty of Computer Science, University of New Brunswick, New Brunswick, Canada(e-mail:rlu1@unb.ca).

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCC.2018.2866405, IEEE Transactions on Cloud Computing

data. As to the template used to represent the users' fingerprint, we adopt the FingerCode representation [25]. Considering the server is honest-but-curious, and the online authentication need quick responses, the proposed e-Finga is characterized by employing an improved homomorphic encryption technology for secure Euclidean distance calculation under composite order group to protect users' fingerprint information and the confidentiality of the matching templates with low overhead in computation. Specifically, the main contributions of this paper are threefold.

- First, our proposed e-Finga scheme provides secure and privacy-preserving online fingerprint authentication service. With e-Finga, the users can access fingerprint authentication service without leaking their sensitive fingerprint data. The matching templates are encrypted and stored in a trusted authority to ensure the storage security. Besides, the communication packages are encrypted and signed to ensure data security in the transmission.
- Second, the scheme provides the efficient and accurate fingerprint matching service. Different from other time-consuming homomorphic encryption techniques, we construct a special homomorphic encryption algorithm over an efficient filter-based fingerprint identification method. By using Pollard's lambda method and constructing trapdoor, the proposed scheme will not affect the accuracy of the underling biometric identification system, that is, it can provide efficient and accurate fingerprint authentication service.
- Third, to evaluate the effectiveness of the proposed scheme, we also develop a custom simulator built in Java with the FVC2006(Forth Fingerprint Verification Competition) database [26]. Performance evaluation demonstrates that our proposed e-Finga can provide an efficient and privacy-preserving fingerprint authentication in real life.

The remainder of this paper is organized as follows: In Section 2, we introduce the system model, security requirements, and design goal. In Section 3, we recall the bilinear pairings, 2DNF cryptosystem and FingerCode-based id matching as preliminaries, and present our e-Finga scheme in Section 4. The security analysis and performance evaluation are followed in Section 5 and 6, respectively. We also review some related works in Section 7. Finally, we draw the conclusions in Section 8.

2 System model, Security requirements and design goals

In this section, we formalize the system model, security requirements, and identify our design goals.

2.1 System Model

In our system model, we mainly focus on how servers provide accurate and efficient fingerprint authentication with encrypted user queries and templates. In particular, the system consists of three parts: trusted authority (TA), online authentication servers of Internet service ($OASers$) and users, as shown in Fig. 1.

- TA is a trusted authority, such as a government department, bootstraps the system initialization by generating and sending system parameters to registered $OASers$ and users respectively. TA is responsible for encryption and storage of sensitive fingerprint templates collected from users. Moreover, TA sends certain encrypted templates to registered $OASers$ with users' authorization. TA performs two functions: system initialization and encrypted template authorization.
- $OASers$ provide personal authentication, such as enterprises like Amazon, Alibaba, etc. $OASers$ should register in TA in advance to be qualified to provide fingerprint authentication service. $OASers$ receive users' register information and requests to TA for related templates. After receiving the encrypted fingerprint templates, $OASers$ can provide personal authentication by using fingerprint matching technique over ciphertext.
- After fingerprint acquisition by TA, the users can register in $OASers$ and query the privacy-preserving online fingerprint authentication service by their fingerprints. Considering the fingerprints contain sensitive information of users, and sending the query in plaintext to $OASers$ may lead to privacy leakage, the users should perform some encryption operations during the process of generating query.

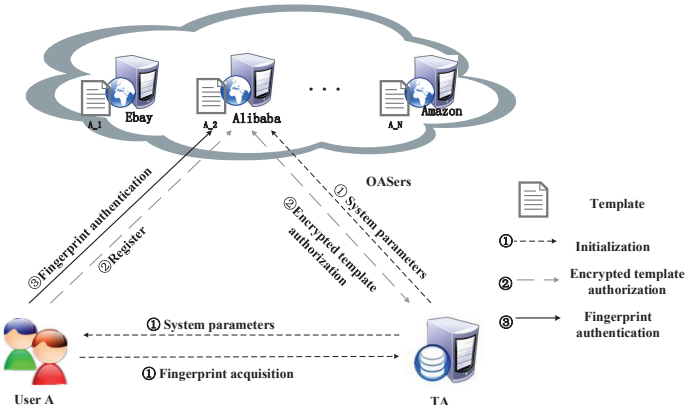


Fig. 1. System model under consideration.

2.2 Security requirements

In our security model, we consider TA is trusted, but $OASers$ are honest-but-curious. Specifically, $OASer$ will honestly execute the operations to identify users' identity, but it also tries to analyse the encrypted templates received from TA and the queries received from users to obtain the original fingerprint data. Besides, an $OASer$ may have malicious behaviors like trying to impersonate another $OASer$ to offer service or have collusion behavior with other $OASers$. In addition, we assume an active adversary A who may eavesdrop on all communication links to obtain encrypted data, guess plaintext values and impersonate an legal user or an $OASer$.

The security of fingerprint data in storage, transmitting and calculation is crucial for the success of privacy-preserving fingerprint authentication scheme. Therefore, in order to guarantee the security of sensitive data and communication packages, the following security requirements should be satisfied.

- *Privacy.* On the one hand, the proposed scheme should protect the users' fingerprint information in the query request, i.e., even if an $OASer$ or an adversary obtains all the queries from users, it cannot obtain the original fingerprint information. In addition, when an $OASer$ performs the computing operation to determine whether the query fingerprint and the fingerprint template match, it

cannot get more information except the matching result. Moreover, the matching result should be protected from adversaries.

- *Confidentiality.* The proposed e-Finga scheme should keep the sensitive fingerprint templates assets. Even if an *OASer* or an adversary stores all the data about user U_i received from TA, it cannot get the original fingerprint information of the related template. Moreover, the scheme should prevent the collusion behavior between *OASers* or templates leak, even an illegal *OASer* gets the encrypted templates assets from another, but it cannot be able to offer fingerprint authentication service.
- *Authentication.* Authenticating an encrypted query/response that is really sent by a legal *OASer*/user and has not been altered during the transmission, e.g., if an illegal user forges a query or response, this malicious operation should be detected. In this sense, only the legal queries and responses can be accepted.

2.3 Design Goal

Under the aforementioned system model and security requirements, our design goal is to develop efficient and privacy-preserving online fingerprint authentication scheme with accurate matching results. Specifically, the following three objectives should be achieved.

- *The security requirements should be guaranteed.* As stated above, if the scheme does not consider the security, the users' sensitive fingerprint information could be disclosed. Therefore, the proposed system should achieve the confidentiality in storage, transmitting and calculation process.
- *The fingerprint authentication with high accuracy should be guaranteed.* The accuracy is the most critical aspects of personal authentication system, and cannot be lowered when protecting users' privacy. Therefore, the proposed scheme should also provide the highly precise and reliable fingerprint authentication.
- *Low communication overhead and low computation complexity should be guaranteed.* Considering the real-time requirements of online fingerprint authentication service, the proposed scheme should have low overhead in terms of communication and computation.

3 Preliminaries

In this section, we review the bilinear pairing technique and the 2DNF cryptosystem, and then describe the FingerCode-based id matching algorithm which will serve as the basis of our proposed scheme.

3.1 Bilinear Pairing

Let $\mathbb{G}$, $\mathbb{G}_T$ be two cyclic groups of the same finite order n, and g be a generator of $\mathbb{G}$. Suppose $\mathbb{G}$ and $\mathbb{G}_T$ are equipped with a pairing, and a non-degenerated and efficiently computable bilinear map $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$ has the following properties.

1) Bilinearity. For all $u, v \in \mathbb{G}$ and $a, b \in \mathbb{Z}_q^*$, we have $e(u^a, v^b) = e(u, v)^{ab}$;
2) Non-degeneracy. $e(g, g) \neq 1_{\mathbb{G}_T}$;
3) Computability. $e(u, v)$ can be computed efficiently for all $u, v \in \mathbb{G}$.

3.2 2DNF

The 2DNF Cryptosystem [27] can achieve the homomorphic properties, which is similar to the Paillier [28] encryption schemes. Concretely, the 2DNF Cryptosystem is comprised of three algorithms: key generation, encryption and decryption.

- Key Generation($Gen(l)$). Given a security parameter $l \in \mathbb{Z}^+$, two l-bit prime numbers q_1, q_2 are first chosen, and compute $N = q_1 \cdot q_2 \in \mathbb{Z}$. Generate a bilinear group $\mathbb{G}$ of order N, and let g, u be two generators of $\mathbb{G}$. Then, $h = u^{q_2}$ is calculated as a random generator of the subgroup of $\mathbb{G}$ with order q_1. Finally, private key $SK = q_1$ and public key $PK = (N, \mathbb{G}, \mathbb{G}_T, e, g, h)$ are outputted.
- Encryption. Assume that the message space consists of integers in the set $\{0, 1, ..., T\}$ with $T < q_2$, then, to encrypt a message m with public key PK, we select a random r from $\{0, 1, ..., N-1\}$ and the ciphertext can be calculated by $C = g^m \cdot h^r \in \mathbb{G}$.
- Decryption. To decrypt a ciphertext C with privacy key $K = q_1$, be aware of $C^{q_1} = (g^m \cdot h^r)^{q_1} = (g^{q_1})^m$, let $\hat{g} = g^{q_1}$. To achieve the corresponding message m, it suffices to compute the discrete logarithm of c^{q_1} base $\hat{g}$. Since $0 \leq m \leq T$ takes expected time $\hat{O}(\sqrt{T})$ using Pollard's lambda method to get the message m.

Note that the decryption time in scheme is the polynomial time in the size of the message space M. Hence, the cryptosystem obviously can be efficiently suitable for short messages.

3.3 FingerCode-Based ID Matching

The FingerCode-based id matching algorithm uses a bank of Gabor filters to capture both local and global details in a fingerprint as a compact fixed length FingerCode [25]. FingerCode for each fingerprint is a n-dimensional feature vector(topically $n = 640$), each element of which is an 8-bit integer. To match two fingerprints, the Euclidean distance between their corresponding FingerCodes are computed and compared with a threshold. For example, given two FingerCodes $x = (x_1, x_2, ..., x_n)$ and $y = (y_1, y_2, ..., y_n)$, their Euclidean distance is:

$$d_{xy} = \sqrt{\sum_{j=1}^{n} (x_i - y_i)^2}$$

If the Euclidean distance between the two FingerCodes is below the threshold Δ_d, the corresponding fingerprints can be considered from the same person. The equal error rate of this filter-based algorithm is in the range of 3-5%, and has a much lower computational complexity and more suitable for online fingerprint authentication.

4 Proposed e-Finga Scheme

In this section, we propose e-Finga, an efficient and privacy-preserving online fingerprint authentication scheme, which mainly consists of following four parts: *System Initialization*, *Encrypted Template Authorization*, *Authentication Query Generation*, *Fingerprint Matching*. Specially, TA bootstraps the system, and provides registration for users and *OASers*, and collects the registered users' fingerprints as matching templates in the *System Initialization* phase. After that, TA encrypts the collected templates, sends the templates of registered users to *OASer* with users' authorization in the *Encryption Template Authorization* phase. Then, the registered users generate queries to *OASer* in

the *Query Generation* phase. Finally, when $OASer$ receives the user's query, it process the matching calculation to judge whether the fingerprints match and response the result to the user in the *Fingerprint Matching* phase. Meanwhile, for easier expression, we give the description of notations used in e-Finga by the following subsections in Table. 1.

TABLE 1
Variables and their descriptions

Variables	Description
l	the secure parameter chosen by TA
q_1, q_2	parameters of bilinear groups
$\mathbb{G}, \mathbb{G}_T$	the bilinear groups with order N
$H_1(), H_2()$	the secure cryptographic hash function
$E()$	the secure asymmetric encryption algorithm, such as ECC
SB	$SB = g^{q_1}$
PB	$PB = e(g,g)^{q_1}$
Δ_d	the threshold of matching judgement
TA	a trusted authority, such as a government department
$OASer$	an online authentication server of Internet service
SK_{U_i}, PK_{U_i}	the private key and public key of the user U_i
SK_{TA}, PK_{TA}	the private key and public key of TA
SK_S, PK_S	the private key and public key of $OASer$
IC_S	the identification code set of $OASers$
RDS	the reference evaluation data set

4.1 System Initialization

We consider TA is a trusted authority bootstraps the system. In the system initialization phase, TA first chooses a security parameter l (l is more than 512) to obtain $(\mathbb{G}, \mathbb{G}_T, q_1, q_2, e, g, h, N = q_1 \cdot q_2)$ by running $Gen(l)$, and computes two secret bases, $SB = g^{q_1}$ and $PB = e(g,g)^{q_1}$. Then, TA chooses a secure asymmetric encryption algorithm $E()$, e.g., ECC, and two secure cryptographic hash functions $H_1()$ and $H_2()$, where $H_1 : \{0,1\}^* \rightarrow \mathbb{G}$, $H_2 : \{0,1\}^* \rightarrow \mathbb{Z}_{q_2}^*$. In addition, TA chooses a random number as its private key $SK_{TA} \in \mathbb{Z}_N^*$ and computes its public key $PK_{TA} = g^{SK_{TA}}$. Finally, TA keeps the $\langle q_1, SK_{TA} \rangle$ secretly, and publishes the system parameters $\langle \mathbb{G}, \mathbb{G}_T, e, g, h, N, PK_{TA}, E(), H_1(), H_2() \rangle$.

When registering in TA, $OASer$ chooses a random number $SK_S \in \mathbb{Z}_N^*$ as private key, computes its corresponding public key $PK_S = g^{SK_S}$, and submits its registering information and PK_S to TA for signature, but it cannot get $\langle SB, PB \rangle$ from TA. And TA distributes a pseudorandom identification code for every registered $OASer$, and the identification code set of $OASers$ is expressed as IC_S.

When registering in TA, user U_i chooses a random number as his/her private key $SK_{U_i} \in \mathbb{Z}_N^*$, computes and submits its public key $PK_{U_i} = g^{SK_{U_i}}$ and his/her information to TA for signature. Then, TA chooses $k_i \in \mathbb{Z}_N^*$ as nuisance parameter for U_i and sends $\langle SB, PB, k_i, IC_S \rangle$ to U_i, where IC_S is the identification code set of $OASers$, and publishes the registered users lists and corresponding public key PK_{U_i}.

Then, U_i's fingerprint information should be collected by TA. After the image and vector extraction of Gabor filters, the FingerCode of a fingerprint can be generated as $X_{U_i} = (x_1, x_2, \cdots, x_n)$, where x_i is an 8-bit integer, $0 < i < n$.

For each user's FingerCode vector, TA executes as follows.

- TA obtains the n-dimensional FingerCode $X_{U_i} = (x_1, x_2, \cdots, x_n)$ of U_i, and computes $x_1' = x_1 + H_2(k_i + c_S)$, $x_2' = x_2 + H_2(k_i + c_S), \cdots, x_n' = x_n + H_2(k_i + c_S)$, where $c_S \in IC_S$, k_i and c_S is only known by registered users and TA, which can resist the exhaustive attack.
- TA chooses n random numbers $r_1, r_2, \cdots, r_n \in \mathbb{Z}_N^*$ for every FingerCode $X_{U_i} = (x_1, x_2, \cdots, x_n)$, and processes as follows.

$$\begin{cases} f_{x_1} = g^{x_1'} \cdot h^{r_1} \\ f_{x_2} = g^{x_2'} \cdot h^{r_2} \\ \vdots \\ f_{x_n} = g^{x_n'} \cdot h^{r_n} \\ f_x' = PB^{(x_1'^2 + x_2'^2 + \cdots + x_n'^2)} \end{cases}$$

For FingerCode vector $X_{U_i} = (x_1, x_2, \cdots, x_n)$ of user U_i, TA obtains $F_{U_i} = (f_{x_1}, f_{x_2}, \cdots, f_{x_n}, f_x')$. Note that F_{U_i} is different related to different $OASers$ because of c_S is a identification code of an $OASer$.

TA also computes $RD_i = PB^i$, where $0 \leq i \leq \Delta_d^2$, Δ_d is the threshold of matching judgement of two FingerCodes' Euclidean distance. Based on the reference data set $RDS = \{RD_0, RD_1, ...RD_i, ..., RD_{\Delta_d^2}\}$, TA creates a Bloom filter BF_{RDS} and uploads BF_{RDS} to all the registered $OASers$ [29].

4.2 Encrypted Template Authorization

If user U_i wants to use the fingerprint authentication service from an $OASer$, he/she should register in the $OASer$ first, then the $OASer$ requests the templates from TA. The Fig. 2 illustrates *Encrypted Templates Authorization* phase.

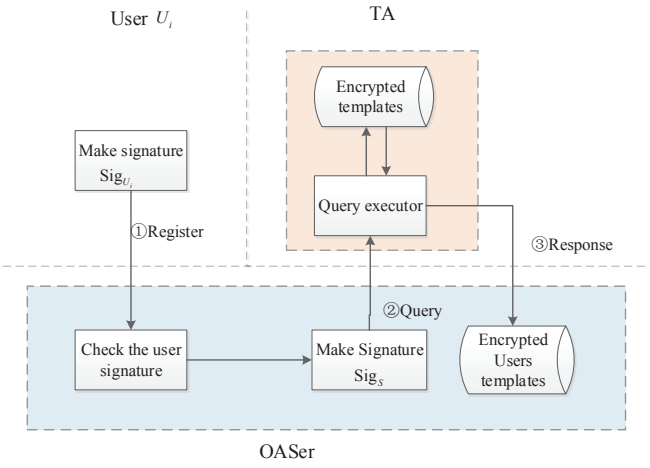


Fig. 2. Encryption Template Authorization.

- *Users Register in $OASer$.* When U_i registering in an $OASer$ for fingerprint authentication service, he/she should make the signature $Sig_{U_i} = H_1(ID_{U_i} \| PK_S \| TS_1)^{SK_{U_i}}$ by using his/her private key SK_{U_i}, where TS_1 is the current time stamp to resist potential replay attack, and ID_{U_i} is the user's identify information. Then, U_i sends $\langle ID_{U_i} \| TS_1 \| Sig_{U_i} \rangle$ to $OASer$.
- *$OASer$ Request to TA.* $OASer$ first checks the user's information and the time stamp TS_1 is within valid term and verifies the signature Sig_{U_i} whether $e(g, Sig_{U_i}) = e(PK_{U_i}, H_1(ID_{U_i} \| PK_S \| TS_1))$.

If it does hold, the signature is accepted, since $e(g, Sig_{U_i}) = e(g, H_1(ID_{U_i}||PK_S||TS_1))^{SK_{U_i}} = e(PK_{U_i}, H_1(ID_{U_i}||PK_S||TS_1))$. Then the $OASer$ makes the signature $Sig_S = H_1(ID_S||TS_2)^{SK_S}$ by using private key SK_S, where TS_2 is the current time stamp and ID_S is the $OASer$'s identify information. Then sends the authorization item $\langle ID_{U_i}||TS_1||Sig_{U_i}||ID_S||TS_2||Sig_S\rangle$ to TA for user U_i's template authorization.

- *Response to $OASer$.* After receiving $\langle ID_{U_i}||TS_1||Sig_{U_i}||ID_S||TS_2||Sig_S\rangle$ from $OASer$, TA first checks the user's information and the $OASer$'s information, and checks the time stamps TS_1 and TS_2 are within valid term, then verify the signatures Sig_{U_i} and Sig_S whether $e(g, Sig_{U_i}) = e(g, H_1(ID_{U_i}||PK_S||TS_1))^{SK_{U_i}} = e(PK_{U_i}, H_1(ID_{U_i}||PK_S||TS_1))$ and $e(g, Sig_S) = e(g, H_1(ID_S||TS_2))^{SK_S} = e(PK_S, H_1(ID_S||TS_2))$. If both equations do hold, the signatures are accepted. Then TA makes a signature $Sig_{TA} = H_1(ID_{U_i}||F_{U_i}||TS_3)^{SK_{TA}}$ using the private key SK_{TA}, where ID_{U_i}, F_{U_i} is the user's information and related encrypted templates, TS_3 is the current time stamp. Then, TA sends the encrypted template information $\langle ID_{U_i}||F_{U_i}||TS_3||Sig_{TA}\rangle$ relate to $OASer$ in response.
- *Storage in $OASer$.* After receiving the $\langle ID_{U_i}||F_{U_i}||TS_3||Sig_{TA}\rangle$ from TA, $OASer$ first checks the time stamp TS_3, then verifies the signature Sig_{TA} whether $e(g, Sig_{TA}) = e(PK_{TA}, H_1(ID_{U_i}||F_{U_i}||TS_3))$. If it does hold, the signature is accepted, the $OASer$ will save $\langle ID_{U_i}||F_{U_i}\rangle$ pairs in database.

4.3 Authentication Query Generation

After registering in an $OASer$, U_i can securely send his/her query request to the $OASers$ and avoid exposing the original fingerprint data by the following procedure.

- U_i firstly obtains his/her fingerprint image through his/her smart terminal. After the image and vector extraction of Gabor filters, the client generates a n-dimensional FingerCode vector $Y_{U_i} = (y_1, y_2, \cdots, y_n)$. Then computes $y'_1 = y_1 + H_2(k_i + c_S), y'_2 = y_2 + H_2(k_i + c_S), \cdots, y'_n = y_n + H_2(k_i + c_S)$, where k_i and c_S is only known by TA and registered users, and c_S is the $OASer$'s identification code.
- U_i uses the the threshold of matching judgement Δ_d processed as follows.

$$\begin{cases} rq_{y_1} = SB^{2\cdot y'_1} \\ rq_{y_2} = SB^{2\cdot y'_2} \\ \vdots \\ rq_{y_n} = SB^{2\cdot y'_n} \\ rq'_y = PB^{y'^2_1+y'^2_2+\cdots+y'^2_n-\Delta_d^2} \end{cases}$$

For the FingerCode vector $Y_{U_i} = (y_1, y_2, \cdots, y_n)$, the user obtains $RQ_{U_i} = (rq_{y_1}, rq_{y_2}, \cdots, rq_{y_n}, rq'_y)$.

- U_i makes a signature $Sig_i = H_1(RQ_{U_i}||ID_{U_i}||TS_4)^{SK_{U_i}}$ by using his/her private key SK_{U_i}, where TS_4 is the current time stamp, which can resist the potential replay attack.
- U_i sends the authentication request $\langle RQ_{U_i}||ID_{U_i}||TS_4||Sig_i\rangle$ to $OASer$.

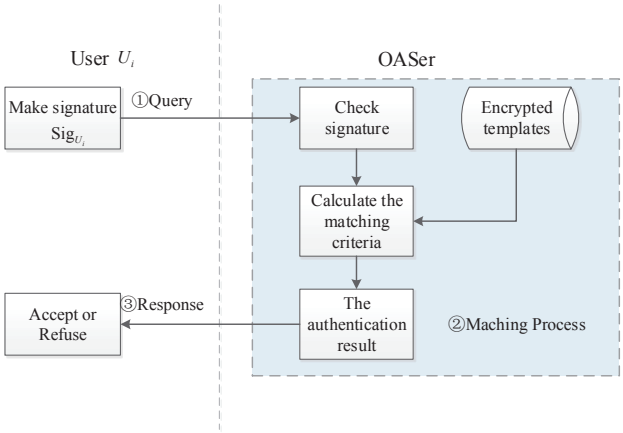


Fig. 3. Fingerprint Matching.

4.4 Fingerprint Matching

Upon receiving U_i's request $\langle RQ_{U_i}||ID_{U_i}||TS_4||Sig_i\rangle$, the $OASer$ provides fast fingerprint authentication service by the following procedure as showed in Fig. 3.

- $OASer$ first checks the time stamp TS_3 is within its valid term, then verifies the signature Sig_i whether $e(g, Sig_i) = e(PK_{U_i}, H_1(RQ_{U_i}||ID_{U_i}||TS_4))$. If it does hold, the signature is accepted.
- According to the user's information ID_{U_i}, $OASer$ searches the related $\langle ID_{U_i}||F_{U_i}\rangle$ pairs in database to get its encrypted template $F_{U_i} = (f_{x_1}, f_{x_2}, \cdots, f_{x_n}, f'_x)$.
- According to the encrypted user query $RQ_{U_i} = (rq_{y_1}, rq_{y_2}, \cdots, rq_{y_n}, rq'_y)$ and encrypted template $F_{U_i} = (f_{x_1}, f_{x_2}, \cdots, f_{x_n}, f'_x)$, $OASer$ computes the matching criteria M_d which are implicitly formed by

$$\begin{aligned} M_d &= \frac{e(f_{x_1}, rq_{y_1}) \cdot e(f_{x_2}, rq_{y_2}) \cdots e(f_{x_n}, rq_{y_n})}{f'_x \cdot rq'_y} \\ &= \frac{e(g^{x'_1} \cdot h^{r_1}, SB^{2\cdot y'_1}) \cdots e(g^{x'_n} \cdot h^{r_n}, SB^{2\cdot y'_n})}{PB^{(x'^2_1+x'^2_2+\cdots+x'^2_n)} \cdot PB^{y'^2_1+y'^2_2+\cdots+y'^2_n-\Delta_d^2}} \\ &= \frac{e(g^{x'_1} \cdot h^{r_1}, g^{q_1\cdot 2\cdot y'_1}) \cdots e(g^{x'_n} \cdot h^{r_n}, g^{q_1\cdot 2\cdot y'_n})}{PB^{(x'^2_1+x'^2_2+\cdots+x'^2_n+y'^2_1+y'^2_2+\cdots+y'^2_n-\Delta_d^2)}} \\ &= \frac{e(g^{x'_1\cdot q_1}, g^{2\cdot y'_1}) \cdots e(g^{x'_n\cdot q_1}, g^{2\cdot y'_n})}{PB^{(x'^2_1+x'^2_2+\cdots+x'^2_n+y'^2_1+y'^2_2+\cdots+y'^2_n-\Delta_d^2)}} \\ &= \frac{e(g,g)^{2q_1\cdot x'_1\cdot y'_1} \cdots e(g,g)^{2q_1\cdot x'_n\cdot y'_n}}{PB^{(x'^2_1+x'^2_2+\cdots+x'^2_n+y'^2_1+y'^2_2+\cdots+y'^2_n-\Delta_d^2)}} \\ &= PB^{\Delta_d^2-((x'_1-y'_1)^2+\cdots+(x'_n-y'_n)^2)} \\ &= PB^{\Delta_d^2-((x_1-y_1)^2+\cdots+(x_n-y_n)^2)} \end{aligned}$$

- $OASer$ runs BF.Test algorithm [29] with Bloom filter BF_{RDS} to judge whether M_d is an element of the set RDS. If M_d is an element of the set RDS, the two fingerprints meet the matching requirement, the authentication result RS is true, otherwise, RS is false, where authentication result RS is a boolean value.
- $OASer$ encrypts RS with the secure asymmetric encryption algorithm $E()$ and U_i's public key PK_{U_i}, and makes a signature $Sig_R = H_1(E_{PK_{U_i}}(RS)||TS_5)^{SK_S}$ by using its private key SK_S, where TS_5 is the current time stamp, and sends $\langle E_{PK_{U_i}}(RS)||TS_5||Sig_R\rangle$ to U_i.

- After receiving $\langle E_{PK_{U_i}}(RS)||TS_5||Sig_R\rangle$ from the server $OASer$, U_i checks the time stamp TS_5 is within valid term, and verifies the signature Sig_R by verifying whether $e(g, Sig_R) = e(PK_S, H_1(E_{PK_{U_i}}(RS)||TS_5))$. Then, U_i decrypts $E_{PK_{U_i}}(RS)$ with U_i's secret key SK_{U_i} to obtain the authentication result. If the authentication succeed, the user can continue accessing $OASer$, if not, the user can choose to authenticate again or abandon access.

  Correctness of the fingerprint comparison algorithm. As the exponential of search criteria $M_d = PB^{\Delta_d^2-((x_1-y_1)^2+\cdots+(x_n-y_n)^2)}$, PB is a generator of a cyclic group with order q_2(q_2 is more than 512-bits), $(x_1 - y_1)^2 + \cdots + (x_n - y_n)^2$ is the square of the Euclidean distance between the two FingerCodes. If the two fingerprints meet the matching criterion, $0 \le \Delta_d^2 - ((x_1 - y_1)^2 + \cdots + (x_n - y_n)^2) \le \Delta_d^2$ and the matching criteria M_d must be an element of the set RDS.

5 Security Analysis

In this section, we analyze the security properties of the proposed e-Finga scheme. In particular, following the security requirements discussed earlier, our analysis will focus on how the proposed privacy-preserving fingerprint authentication scheme can achieve the users' fingerprint privacy, templates confidentiality, and authentication of the query request and response.

- *The privacy of users' original fingerprint data item.* In the scheme, the user's request is $\langle RQ_{U_i}||ID_{U_i}||TS_4||Sig_i\rangle$. In our security model, the adversary A who may eavesdrop on all communication links and the $OASer$ could get the user's encrypted FingerCode data $RQ_{U_i} = (rq_{y_1}, rq_{y_2}, \cdots, rq_{y_n}, rq'_y)$ which can be implicitly expressed as

$$\begin{cases} rq_{y_1} = SB^{2\cdot y'_1} \\ rq_{y_2} = SB^{2\cdot y'_2} \\ \vdots \\ rq_{y_n} = SB^{2\cdot y'_n} \\ rq'_y = PB^{y'^2_1+y'^2_2+\cdots+y'^2_n-\Delta_d^2} \end{cases}$$

  Note that $\langle SB, PB\rangle$ are only known by TA and legal users, vector $(y'_1, y'_2, \cdots, y'_n)$ cannot be computed. In addition, to avoid the exhaustive attack against $(rq_{y_1}, rq_{y_2}, \cdots, rq_{y_n}, rq'_y)$ by Pollard's lambda method, the sample space of user's original FingerCode vector $Y_{U_i} = (y_1, y_2, \cdots, y_n)$ is increased by computing $y'_1 = y_1 + H_2(k_i + c_S), y'_2 = y_2 + H_2(k_i + c_S), \cdots, y'_n = y_n + H_2(k_i + c_S)$, where k_i and c_S are only known by TA and legal users, the adversary is unable to recover any useful information.

  Besides, in our security model, an $OAser$ may tries to impersonate another $OASer$ to offer service or have collusion behavior with other $OASers$. c_S is added to increase the sample space of FingerCode vector which is a pseudorandom identification code of an $OASer$. It means that the same user creates different request to different $OASers$, and an $OASer$ cannot impersonate another $OASer$ to offer service.

  Moreover, in our security model, $OASers$ will honestly execute the operations but curious about the user's FingerCode information. The matching calculation in $OASer$ is based on encrypted user's query $RQ_{U_i} = (rq_{y_1}, rq_{y_2}, \cdots, rq_{y_n}, rq'_y)$ instead of user's FingerCode data. The matching criteria M_d can be expressed as $M_d = PB^{\Delta_d^2-((x_1-y_1)^2+\cdots+(x_n-y_n)^2)}$, where PB is unknown by $OASers$. $OASer$ runs BF.Test algorithm with Bloom filter BF_{RDS} to judge whether M_d is an element of the set RDS and gets the matching result RS. In the process, $OASer$ can only know the matching result instead of the user's query FingerCode vector $Y_{U_i} = (y_1, y_2, \cdots, y_n)$ and the square of Euclidean distance $(x_1 - y_1)^2 + \cdots + (x_n - y_n)^2$. In specific, the authentication result RS is encrypted with a secure asymmetric encryption $E()$ and U'_i public key PK_{U_i}. Since only the user U_i has his/her private key SK_{U_i}, the adversary A could not get the authentication result RS.

  In conclusion, the proposed e-Finga scheme can protect the privacy of users' fingerprint information and the authentication result.
- *The confidentiality of templates.* In our security model, the adversary A who may eavesdrop on all communication links and the $OASer$ could get related encrypted templates data. In e-Finga scheme, the encrypted templates are stored in pairs $\langle ID_{U_i}||F_{U_i}\rangle$. The user U_i's encrypted template is $F_{U_i} = (f_{x_1}, f_{x_2}, \cdots, f_{x_n}, f'_x)$, which can be implicitly expressed as

$$\begin{cases} f_{x_1} = g^{x'_1} \cdot h^{r_1} \\ f_{x_2} = g^{x'_2} \cdot h^{r_2} \\ \vdots \\ f_{x_n} = g^{x'_n} \cdot h^{r_n} \\ f'_x = PB^{(x'^2_1+x'^2_2+\cdots+x'^2_n)} \end{cases}$$

  Note that $\langle g, h\rangle$ is a published parameter, random numbers $r_1, r_2, \cdots, r_n \in \mathbb{Z}_N^*$ are chosen to add confounding factors. Since the random numbers $r_1, r_2, \cdots, r_n \in \mathbb{Z}_N^*$ are chosen by TA, and PB is only known by TA and legal users, vector $(x'_1, x'_2, \cdots, x'_n)$ cannot be computed. In addition, to avoid the exhaustive attack by Pollard's lambda method, the sample space of template vector $X_{U_i} = (x_1, x_2, \cdots, x_n)$ is increased by computing $x'_1 = x_1 + H_2(k_i + c_S), x'_2 = x_2 + H_2(k_i + c_S), \cdots, x'_n = x_n + H_2(k_i + c_S)$, where k_i and c_S are only known by TA and legal users, the adversary is unable to recover any useful information.

  Besides, in our security model, an $OAser$ may tries to impersonate another $OASer$ to offer service or have collusion behavior with other $OASers$. c_S is added to increase the sample space of FingerCode vector which is a pseudorandom identification code of an $OASer$. It means that the same template is encrypted to different ciphertext and sent to different $OASers$. Even if the $OASers$ have collusion behaviors, an $OASer$ still cannot impersonate another $OASer$ to offer service.

  Moreover, in our security model, $OASers$ will honestly execute the operations but curious about the templates information. The matching calculation in $OASer$ is based on encrypted template $F_{U_i} = (f_{x_1}, f_{x_2}, \cdots, f_{x_n}, f'_x)$ instead of the original template data. The matching criteria M_d can be expressed as $M_d = PB^{\Delta_d^2-((x_1-y_1)^2+\cdots+(x_n-y_n)^2)}$, where PB is unknown by $OASers$. $OASer$ runs BF.Test algorithm with Bloom filter BF_{RDS} to judge whether M_d is an element of the set RDS and gets the matching result RS. In the process, $OASer$ can only know the matching result instead

of the template vector $X_{U_i} = (x_1, x_2, \cdots, x_n)$ and the square of Euclidean distance $(x_1 - y_1)^2 + \cdots + (x_n - y_n)^2$.
Above all, the proposed e-Finga scheme can achieve confidentiality of templates.

- *The authentication of the query request and response.* In our security model, we assume an active adversary A who may eavesdrop on all communication links to obtain encrypted data, guess plaintext values and impersonate an legal user or an $OASer$. Thus, every encrypted query/response are signed by Boneh-Lynn-Shacham(BLS) short signature [30]. Since the BLS short signature is provably secure under the computational Diffie-Hellman problem in the random oracle model, the data authentication can be guaranteed. The senders signature the query with their private key, which the recipient can authenticate the validity of the sender. Moreover, we add the current time stamp as a section of signature which can resist potential replay attack. As a result, the query and response messages are verified within valid term and signed by legal party, which can resist potential replay attack and counterfeit attack.

From the above analysis, we can conclude that the proposed e-Finga scheme is secure and privacy-preserving, and achieves our security design goal.

6 Performance Evaluation

In this section, we evaluate the performance of the proposed e-Finga scheme in terms of the computation and communication costs. Then we implement e-Finga and deploy it in real environment to evaluate its integrated performance.

6.1 Evaluation Environment

In order to measure the integrated performance of e-Finga in real environment, we implement e-Finga on a workstation with a real fingerprint database. Specially, a workstation with two 2.3GHz 6-core processor, 64GB RAM, Windows 7, was chosen to simulate the process on TA, $OASer$ and users. Based on e-Finga scheme, an e-Finga application built in Java, named e-Finga.exe, is installed on the workstation, and the simulator for TA and $OASer$ is deployed in the workstation. Users who registered in TA can obtain online fingerprint authentication by e-Finge.exe. In particular, when a user inputs the fingerprint data by e-Finga.exe, the client sends a query request to the $OASer$ and get the response. In addition, we choose one real dataset to evaluate the efficiency and accuracy, we test our scheme on the FVC2006(Forth Fingerprint Verification Competition) database [26].

6.2 Computation and Communication Costs

The proposed e-Finga scheme can offer online efficient fingerprint authentication service. Specifically, we assume the FingerCode is n-dimensional feature vector. When TA generates the encrypted template $F_{U_i} = (f_{x_1}, f_{x_2}, \cdots, f_{x_n}, f'_x)$, it requires $2n+1$ exponentiation operations and $2n$ multiplication operations. When the user generates the encrypted fingerprint information $RQ_{U_i} = (rq_{y_1}, rq_{y_2}, \cdots, rq_{y_n}, rq'_y)$ in the *Query Generation* phase, it requires $n+1$ exponentiation operations and $2n$ multiplication operations. In the *fingerprint matching* phase, it will cost $OASer$ n pairing operations and $n+1$ multiplication operations. Denote the computational costs of an exponentiation operation, a multiplication operation and a pairing operation by C_e, C_m, C_p, respectively. Then, totally for TA, the user and $OASer$, the computational cost will be $(2n+1) * C_e + 2n * C_m$, $(n+1) * C_e + 2n * C_m$ and $n * C_p + (n+1) * C_m$ in e-Finga.

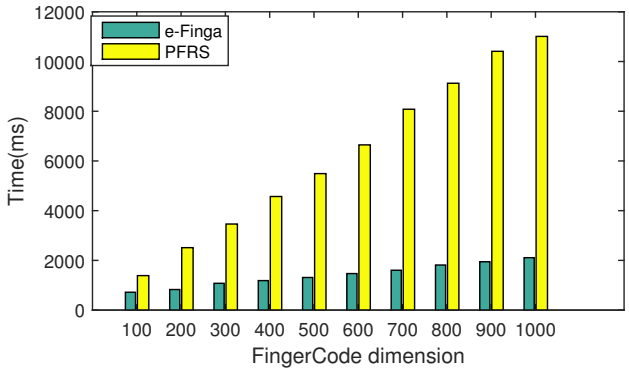


Fig. 4. Query and response time in e-Finga and PFRS.

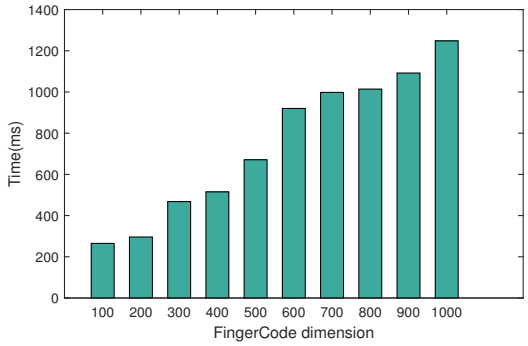
Different from many of time-consuming fully and partially homomorphic encryption techniques, the proposed e-Finga uses lightweight multi-party random masking and polynomial aggregation techniques, which can provide efficient online fingerprint authentication while preserves the privacy of the fingerprint information with low overhead in computation. In the following, we select a fingerprint recognition system as comparison, which based on homomorphic encryption and FingerCode templates [15], called PFRS in the rest of paper for the sake of simplicity. PFRS achieves two-party privacy-preserving by using Paillier encryption technique. We assume there are n components for each FingerCode($n = 640$ in PFRS). And the corresponding computational costs of the user and the server in PFRS are $(3n+1) * C_e + (n+1) * C_m$and $(2n+2) * C_e + (n+3) * C_m$ respectively. We present the computation complexity comparison of e-Finga and PFRS in Table 2.

TABLE 2
Comparison of Computation Complexity

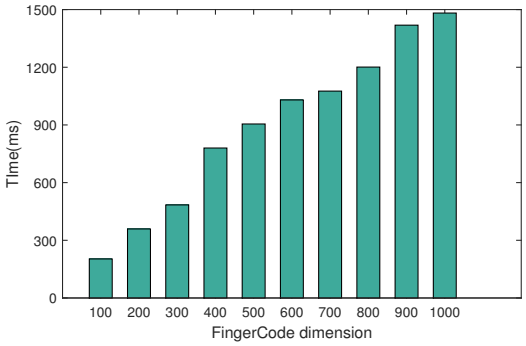
	e-Finga	PFRS
User	$(n+1)*C_e+2n*C_m$	$(3n+1) * C_e + (n+1) * C_m$
Server	$n*C_p+(n+1)*C_m$	$(2n+2) * C_e + (n+3) * C_m$
Cost time	1.5s	7.0s

For better comparison, we implement the proposed e-Finga and PFRS in JAVA. In specific, we test on the FVC2006 DB1 [26]. Fig. 4 depicts the computation overhead varying of e-Finga and PFRS with the number of FingerCode dimensions, and we can find that with the increasing of the numbers of dimensions, the computation overhead of PFRS significantly increases and it is much higher than that of our proposed e-Finga scheme. Since the dimension of FingerCode feature vector is 640 topically, the average cost time of query and response in a workstation is 1.5s in e-Finga, which is acceptable for online application. By comparison, e-Finga is more efficient and more secure in outsourcing scenarios over PFRS.

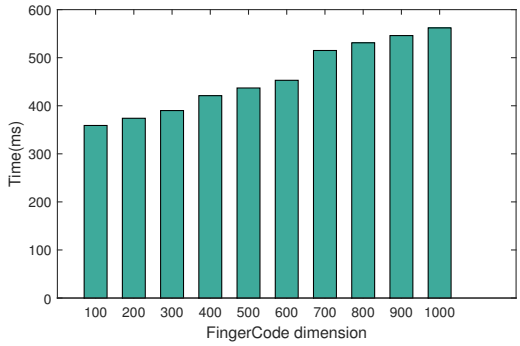
In addition, we have made a comparison of communication costs between e-Finga and PFRS. In e-Finga, the user's query is



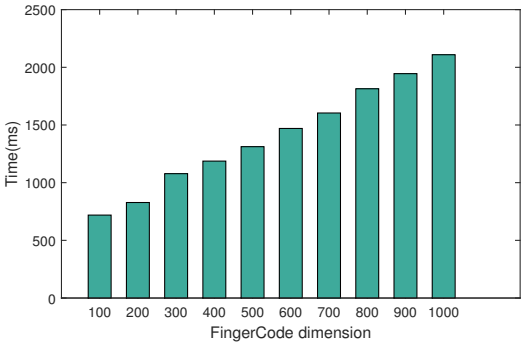
(a) Computation cost of $OASer$ with different FingerCode dimensions.



(b) Computation cost of TA with different FingerCode dimensions.



(c) Computation cost of user with different FingerCode dimensions.



(d) Query and response time in real time

Fig. 5. Computation complexity of e-Finga.

$E_{U_i} = E_{PK_S}(RQ_{U_i}||U_i||TS_3)$, and the response is the authentication result RS, where RS is a boolean value. We calculate the size of the query package and response package of the above two schemes and the results are shown in Table 3. Since the query package is 20KB less than 164KB in PFRS while the dimension is 640, our proposed e-Finga scheme can accomplish better efficiency in terms of communication overhead.

TABLE 3
Comparison of Communication Costs(n=640)

	e-Finga	PFRS
User	20KB	164KB
Server	1B	1B

6.3 Experimental Evaluation

1) *Efficiency Evaluation:* According to the system model, the performances of e-Finga in $OASer$, TA and the users are mainly determined by the computation complexity.

- *$OASer$:* In our proposed e-Finga scheme, the factors which may impact the computation complexity in $OASer$ is the dimension of FingerCode, which affects the length of a template and a user query, and further affect the computation complexity in $OASer$. Therefore, we choose different numbers of the dimension to illustrate the computation cost of $OASer$. We select the dimension of FingerCode from 100 to 1000(each element is an 8-bits integer). As shown in Fig. 5.(a), it is obviously that the computation cost of $OASer$ linearly increases with the increasing of the dimension of FingerCode. The reason is that, when the dimension of FingerCode increases, $OASer$ computes the criteria M_d using bilinear pairing for the increasing length of encrypted FingerCode vectors, which will spend more time in the matching process.
- *TA:* In e-Finga sheme, the compute operations are mainly in encrypting the collected templates in the phase *System Initialization*. Therefore, different dimensions of the FingerCode are chosen to illustrate the computation cost of TA. As shown in Fig. 5.(b), the dimension of the FingerCode is selected from 100 to 1000. It is obvious that increasing the dimension of FingerCode linearly increases the computation cost of TA. The reason is that, when TA publishes the dataset of a user's fingerprint template to $OASer$, in the form of $F_{U_i} = (f_{x_1}, f_{x_2}, \cdots, f_{x_n}, f'_x)$, which will spend more time with the increasing of the dimension of FingerCode.
- *the users:* The compute operations in the client are in the phase *User Query Generation*. Therefore, different dimensions of FingerCode are chosen to illustrate the computation cost of the client. To observe the computation cost of the user, the dimension of FingerCode are selected from 100 to 1000. As shown in Fig. 5.(c), the computation overhead of users increases with the increasing of the dimension of FingerCode. When a user publishes the dataset $RQ_{U_i} = (rq_{y_1}, rq_{y_2}, \cdots, rq_{y_n}, rq'_y)$ to $OASer$, he/she

will spend more time with the increasing of dimensions of FingerCode.

As a consequence, our proposed scheme can achieve privacy-preserving fingerprint verification with low computation complexity in $OASer$, TA and the users.

2) *Integrated performance in real environment:* In order to evaluate the integrated performance of our proposed scheme, e-Finga is deployed in a real environment. In specific, we test our system on the FVC2006 DB1 [26], and the data set of FingerCode are extracted by Matlab Fingerprint Recognition System V2 [31]. In addition, the client and $OASer$ are connected through a 802.11g WLAN, and client will send a query request to $OASer$ and get the response through WLAN. Therefore, we evaluate performance of e-Finga with different dimensions of FingerCode in real environment. To observe the integrated performance of e-Finga, the dimension of FingerCode are selected from 100 to 1000. As shown in Fig. 5.(d), the average query and response time of e-Finga increases with the increasing of the dimension of FingerCode. We can find that the entire overhead for once whole fingerprint authentication service query and response time is approximate to 1.5s in the real environment(640-dimension FingerCode).
3) *Accuracy analysis:* The subsequent privacy transformation will not affect the accuracy of the underling biometric identification system. In the FingerCode-based id matching algorithm, the Euclidean distance between their corresponding FingerCodes are computed and compared with a threshold. Given two FingerCodes $x = (x_1, x_2, ..., x_n)$ and $y = (y_1, y_2, ..., y_n)$, their Euclidean distance is $d_{xy} = \sqrt{\sum_{j=1}^{n}(x_i - y_i)^2}$, then judge whether the Euclidean distance between the two FingerCodes below the threshold Δ_d. In our e-Finga scheme, the matching criteria M_d is formed by $M_d = PB^{\Delta_d^2-((x_1-y_1)^2+\cdots+(x_n-y_n)^2)}$, where $(x_1-y_1)^2+\cdots+(x_n-y_n)^2$ is the square of the Euclidean distance. The evaluation data set $RDS = \{RD_0, RD_1, ...RD_i, ..., RD_{\Delta_d^2}\}$, where $RD_i = PB^i$, $0 \le i \le \Delta_d^2$. If the inequality $(x_1 - y_1)^2 + \cdots + (x_n - y_n)^2 \le \Delta_d^2$ holds, $OASer$ can identify M_d is an element of set RDS with Bloom filter BF_{RDS}, which is the same matching condition as the original FingerCode-based id matching algorithm. The equal error rate of the FingerCode algorithm is in the range of 3-5% [25], and the false positive rate of Bloom Filter BF_{RDS} is $0.6185^{m/t}$, where t is the number of elements have been added into the Bloom filter, m is the length of the Bloom filter [32]. Specifically, the false positive rate is 3.54×10^{-11}, where we choose $t = \Delta_d^2 = 100$ and $m = 5000$. In the real environment, the equal error rate of e-Finga scheme is still approximately in the range of 3-5%.

7 Related Works

The idea of direct key generation from biometric data was first raised in 1994 [33], but the privacy issues of biometric data bring out increasing concerns recently. In this section, some related works on privacy-preserving fingerprint authentication are briefly discussed.

Jin [14] proposed a novel two-factor authenticator based on iterated inner products between pseudo-random numbers and the user fingerprint feature which is named "BioHashing". The main drawback of this method is the low performance when an impostor steals the Hash key or the pseudo-random numbers of a party and tries to authenticate as the party [11]. Juels and Sudan [13] introduce the idea of Fuzzy Vault to formalize the use of error correcting codes for such applications. Many improved versions of fuzzy vault have been proposed, and apply in many applications scenarios [34] [35]. But these privacy transformation will affect the accuracy of the underling fingerprint identification system. Moreover, these fingerprint template protection frameworks cannot realize privacy-preserving in our scenario where the server is honest-but-curious, since the template data is known by servers in the schemes above [36].

The security problems with the outsourced databases can be solved if the critical data are encrypted. However, it leads to the problem how the data center can perform computation on encrypted data. Homomorphic encryption is a promising solution for this problem. Blanton and Gasti provided privacy-preserving protocols for minutia-based fingerprint representations [37], which utilized homomorphic encryption and garbled circuit evaluation. Moreover, Barni et al. [15] proposed a privacy-preserving FingerCode authentication based on homomorphic cryptosystem, where privacy transformation would not affect the accuracy of the underling biometric identification system. Kang designed a protocol based on homomorphic that allows the server and the user jointly computing the Euclidean distance between the template data and the query data [38]. However, the existing homomorphic encryption schemes, as mentioned in [20], are still not practical for arbitrary arithmetic computation over encrypted data due to the so-called bootstrapping that results in increasing computation overhead, and are not suit for online application.

The searchable encryption technique was introduced to solve the search problem over encrypted data. D.X Song [39] carried out the first significant work on the encrypted search in symmetric setting. But only the search word is exactly the same as the predefined word can users successfully search. Full homomorphic encryption seems to be a promising option for it can at least compute arbitrary mathematical function with encrypted data without having the decryption key [16], but it has many disadvantages as we discussed above. Even though searchable encryptions have been widely regarded as standard techniques to secure search over encrypted database, most existing schemes, as discussed in [21], can only support equality test and cannot support more complex arithmetic operations in the fingerprint matching system.

Yuan et al. [22] proposed a privacy-preserving cloud-based fingerprint identification scheme, based on matrix operation. To improve the security of the above scheme, Wang et al. [23] proposed a security-enhanced matrix-based scheme in the cloud, called CloudBI. However, these schemes assume the database owner encrypts all the fingerprint data and outsources the database the database to the cloud. In the identification phase, the users' queries are first sent to the database owner for encryption. The database owner is considered trusted, which is not conform the actual situation.

Different from all of the above works, the proposed e-Finga scheme aims at the efficiency and privacy issues, and based on an improved homomorphic encryption technology for secure Euclidean distance calculation over composite order group. In particular, the privacy transformation will not affect the accuracy of the underling fingerprint identification system.

8 Conclusion

In this paper, we have proposed an efficient and privacy-preserving online fingerprint authentication scheme, called e-Finga, over encrypted outsourced data. Based on an improved homomorphic encryption technology for secure Euclidean distance calculation over composite order group, the proposed e-Finga can achieve the privacy of user fingerprint and confidentiality of matching templates. Specifically, *OASer* can directly compute the matching criteria on ciphertext without decryption, and the accuracy of the underling fingerprint identification system will not be compromised. Meanwhile, the matching result can also only be decrypted by the registered user. Thus, the user can get secure and accurate fingerprint authentication without divulging his/her fingerprint information. Detailed security analysis shows its security strength and privacy-preserving ability, and extensive experiments are conducted to demonstrate its efficiency.

Availability

The implementation of the proposed e-Finga scheme and relevant information can be downloaded at `https://xdzhuhui.com/demo/e-Finga`.

References

[1] Q. Wei, H. Zhu, R. Lu, and H. Li, "Achieve efficient and privacy-preserving online fingerprint authentication over encrypted outsourced data," in *2017 IEEE International Conference on Communications*. IEEE, 2017, pp. 1–6.

[2] N. T. Hoang T, Choi D, "Gait authentication on mobile phone using biometric cryptosystem and fuzzy commitment scheme," *International Journal of Information Security*, vol. 14, no. 6, pp. 549–560, 2015.

[3] Z. F. Yang W, Huang X, "Comparative competitive coding for personal identification by using finger vein and finger dorsal texture fusion," *Information sciences*, vol. 268, pp. 20–32, 2014.

[4] N. Yeap, "Your finger is about to replace your bank password," http://money.cnn.com/2015/06/05/technology/bank-fingerprint-reader/, July 2015.

[5] "Hsbc offers voice and fingerprint id system to customers," http://www.bbc.com/news/business-35609833, February 2016.

[6] S. C. Alliance, "Smart card alliance.," *INSIDE Contactless Offers Free, Downloadable, Open NFC API and Source Code on SourceForge*, 2015.

[7] Z. X. Kong C, Zhao J, "A fingerprint payment system based on hbase," *Scientific Journal of Information Engineering*, vol. 73, no. 2, pp. 61–67, 2016.

[8] G. K. Gowda V R C, "Real time vehicle fleet management and security system," *Intelligent Computational Systems (RAICS), 2015 IEEE Recent Advances in. IEEE*, pp. 417–421, 2016.

[9] H. Zhu, X. Liu, R. Lu, and H. Li, "Efficient and privacy-preserving online medical prediagnosis framework using nonlinear svm," *IEEE journal of biomedical and health informatics*, vol. 21, no. 3, pp. 838–850, 2017.

[10] M. Shen, B. Ma, L. Zhu, R. Mijumbi, X. Du, and J. Hu, "Cloud-based approximate constrained shortest distance queries over encrypted graphs with privacy protection," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 4, pp. 940–953, 2018.

[11] A. B. J. T. Ngo, David Chek Ling, *Biometric Security*. Cambridge Scholars Publishing, 2015.

[12] J. Pagliery, "Hackers stole 5.6 million government fingerprints - more than estimated," http://money.cnn.com/2015/09/23/technology/opm-fingerprint-hack/, September 2015.

[13] A. Juels and M. Sudan, "A fuzzy vault scheme," *Designs, Codes and Cryptography*, vol. 38, no. 2, pp. 237–257, 2006.

[14] A. T. B. Jin, D. N. C. Ling, and A. Goh, "Biohashing: two factor authentication featuring fingerprint data and tokenised random number," *Pattern Recognition*, vol. 37, no. 11, pp. 2245–2255, 2004.

[15] M. Barni, T. Bianchi, D. Catalano, and M. D. Raimondo, "A privacy-compliant fingerprint recognition system based on homomorphic encryption and fingercode templates," in *IEEE BTAS 2010*, 2010, pp. 1–7.

[16] C. Gentry, "A fully homomorphic encryption scheme," *Dissertations and Theses - Gradworks*, 2009.

[17] B. K. Maral V, Kale S, "Homomorphic encryption for secure data mining in cloud," *International Journal of Engineering Science*, 2016.

[18] Y. Rahulamathavan, S. Veluru, R.-W. Phan, J. Chambers, and M. Rajarajan, "Privacy-preserving clinical decision support system using gaussian kernel-based classification," *Biomedical and Health Informatics, IEEE Journal of*, vol. 18, no. 1, pp. 56–66, Jan 2014.

[19] X. Liu, R. Lu, J. Ma, L. Chen, and B. Qin, "Privacy-preserving patient-centric clinical decision support system on naive bayesian classification," *IEEE journal of biomedical and health informatics*, vol. 20, no. 2, pp. 655–668, 2016.

[20] M. Naehrig, K. Lauter, and V. Vaikuntanathan, "Can homomorphic encryption be practical?" in *Proceedings of the 3rd ACM workshop on Cloud computing security workshop*. ACM, 2011, pp. 113–124.

[21] N. Cao, C. Wang, M. Li, K. Ren, and W. Lou, "Privacy-preserving multi-keyword ranked search over encrypted cloud data," *IEEE Transactions on parallel and distributed systems*, vol. 25, no. 1, pp. 222–233, 2014.

[22] J. Yuan and S. Yu, "Efficient privacy-preserving biometric identification in cloud computing," in *INFOCOM, 2013 Proceedings IEEE*. IEEE, 2013, pp. 2652–2660.

[23] Q. Wang, S. Hu, K. Ren, M. He, M. Du, and Z. Wang, "Cloudbi: Practical privacy-preserving outsourcing of biometric identification in the cloud," in *European Symposium on Research in Computer Security*. Springer, 2015, pp. 186–205.

[24] C. Zhang, L. Zhu, and C. Xu, "Ptbi: An efficient privacy-preserving biometric identification based on perturbed term in the cloud," *Information Sciences*, vol. 409, pp. 56–67, 2017.

[25] A. K. Jain, S. Prabhakar, L. Hong, and S. Pankanti, "Filterbank-based fingerprint matching," *Image Processing, IEEE Transactions on*, vol. 9, no. 5, pp. 846–859, 2000.

[26] "Fvc2006: the forth international fingerprint verification competition," http://bias.csr.unibo.it/fvc2006/default.asp.

[27] D. Boneh, E.-J. Goh, and K. Nissim, "Evaluating 2-dnf formulas on ciphertexts," in *Theory of Cryptography Conference*. Springer, 2005, pp. 325–341.

[28] P. Paillier *et al.*, "Public-key cryptosystems based on composite degree residuosity classes," in *Eurocrypt*, vol. 99. Springer, 1999, pp. 223–238.

[29] B. H. Bloom, "Space/time trade-offs in hash coding with allowable errors," *Communications of the ACM*, vol. 13, no. 7, pp. 422–426, 1970.

[30] D. Boneh, B. Lynn, and H. Shacham, "Short signatures from the weil pairing, asiacrypt 01," *Journal of Cryptology*, vol. 17, no. 4, pp. 297–319, 2001.

[31] "Matlab fingerprint recognition system v2," http://matlab-recognition-code.com.

[32] A. Broder and M. Mitzenmacher, "Network applications of bloom filters: A survey," *Internet mathematics*, vol. 1, no. 4, pp. 485–509, 2004.

[33] A. Bodo, "Method for producing a digital signature with aid of a biometric feature," *German patent DE*, vol. 42, no. 43, p. 908, 1994.

[34] B. M. R. Dellys H N, Benadjimi N, "Fingerprint fuzzy vault chaff point generation by squares method," in *2015 7th International Conference of Soft Computing and Pattern Recognition (SoCPaR)*. IEEE, 2015, pp. 357–362.

[35] A. A. Nasiri and M. Fathy, "Alignment-free fingerprint cryptosystem based on multiple fuzzy vaults," in *Artificial Intelligence and Signal Processing (AISP), 2015 International Symposium on*. IEEE, 2015, pp. 251–255.

[36] J. Hartloff and V. Govindaraju, "Security analysis for fingerprint fuzzy vaults," *Proceedings of SPIE - The International Society for Optical Engineering*, vol. 8712, no. 4, pp. 1–12, 2013.

[37] G. P. Blanton M, "Secure and efficient iris and fingerprint identification," *Biometric Security*, 2015.

[38] Z. J. F. Q. HE Kang, LI Mengxing, "Fingercode based remote fingerprint authentication scheme using homomorphic encryption," *Computer Engineering and Applications*, vol. 49, no. 24, pp. 78–82, 2013.

[39] D. X. Song, D. Wagner, and A. Perrig, "Practical techniques for searches on encrypted data," *IEEE Symposium on Security and Privacy*, pp. 44–55, 2000.

Hui Zhu (M'13) received his B.Sc. degree from Xidian University in 2003, M.Sc. degree from Wuhan University in 2005, and Ph.D. degrees from Xidian University in 2009. In 2013, he was with School of Electrical and Electronics Engineering, Nanyang Technological University as a research fellow.

Since 2016, he has been the professor in the School of Cyber Engineering, Xidian University, China. His research interests include the areas of applied cryptography, data security and privacy.

Qing Wei received the B.Sc. degree from Xidian University in 2015.

She is current working toward the Master's degree with the School of Cyber Engineering, Xidian University, China. Her research interests are in the areas of applied cryptography, cyber security and privacy.

Xiaopeng Yang received the B.Sc. from xidian university in 2014, M.Sc from xidian university in 2017.

He is current working toward the Doctors degree with the school of Cyber Engineering, Xidian University, China. His interests are in the areas of applied cryptography, data security and privacy.

Rongxing Lu (S'09-M'10-SM'15) has been an assistant professor at the Faculty of Computer Science, University of New Brunswick (UNB), Canada, since August 2016. Before that, he worked as an assistant professor at the School of Electrical and Electronic Engineering, Nanyang Technological University (NTU), Singapore from April 2013 to August 2016. Rongxing Lu worked as a Postdoctoral Fellow at the University of Waterloo from May 2012 to April 2013. He was awarded the most prestigious "Governor General's Gold Medal", when he received his PhD degree from the Department of Electri-cal and Computer Engineering, University of Waterloo, Canada, in 2012; and won the 8th IEEE Communications Society (ComSoc) Asia Pacific (AP) Outstanding Young Researcher Award, in 2013. He is presently a senior member of IEEE Communications Society.

His research interests include applied cryptography, privacy enhancing technologies, and IoT-Big Data security and privacy. Dr. Lu currently serves as the Secretary of IEEE ComSocCIS-TC.

Hui Li (M'10) Received his B.Sc. degree from Fudan University in 1990, M.Sc. and Ph.D. degrees from Xidian University in 1993 and 1998, respectively.

Since 2005, he has been the professor in the school of Telecommunication Engineering, Xidian University, China. His research interests are in the areas of cryptography, wireless network security, information theory and network coding.

Dr. Li served as TPC co-chair of ISPEC 2009 and IAS 2009, general co-chair of E-Forensic 2010, ProvSec 2011 and ISC 2011,honorary chair of NSS 2014, ASIACCS 2016.