Peer-to-Peer Networking and Applications (2021) 14:2171–2186
https://doi.org/10.1007/s12083-021-01120-7

MASK: Efficient and privacy-preserving m-tree based biometric identification over cloud

Xiaopeng Yang[1,2] · Hui Zhu[1] · Fengwei Wang[1,2] · Songnian Zhang[2] · Rongxing Lu[2] · Hui Li[1]

Received: 16 August 2020 / Accepted: 8 March 2021 / Published online: 25 May 2021
© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2021

Abstract

In recent years, the extensive application of biometric identification has been witnessed in various fields, such as airport service, criminal investigation, counter-terrorism and so on. Due to the sensitivity of the biometric data, people's concern over the leakage of their biometric data is a critical obstacle to hinder the future adoption of biometric identification applications. To address this problem, many schemes focusing on the privacy protection during biometric identification process have been proposed. However, identifying an individual in a huge database still faces many challenges while considering privacy protection and efficiency at the same time. In this paper, an efficient and privacy-preserving cloud based biometric identification scheme (named MASK) is proposed based on the M-tree data structure and symmetric homomorphic encryption (SHE) scheme. With MASK, the privacy of the user's identification request and service provider's dataset is guaranteed, while the computational cost of the cloud servers in searching the biometric dataset is significantly reduced. Besides, the accuracy of the identification service is not lost. Detailed security analysis shows that MASK can resist various known security threats. In addition, MASK is implemented and evaluated with a synthetic dataset and a real face dataset, and extensive simulation results demonstrate that MASK is efficient in terms of computational and communication costs.

Keywords Biometric identification · Privacy-preserving · Efficiency · M-tree

1 Introduction

Biometric identification aims to identify a person among a group of individuals based on her/his biometric feature and has been applied in many areas, such as airport service, criminal investigation and counter-terrorism, etc [11, 17, 18, 22, 30]. For example, when the police find some biometric traits at a crime scene, they search these biometric traits in known offender databases to find possible culprits. In a typical biometric identification system, the user submits a biometric template as the identification request to the service provider who owns a biometric template dataset, then the service provider compares the identification request with all the biometric templates in the dataset to get the identification result. Since the searching process is very time consuming and the service provider may not be equipped with enough computer power, the service provider trends to outsource the dataset to the cloud to provide efficient identification service. However, in such a scenario, the user's identification request and service provider's dataset may be leaked to the cloud server. Since the biometric trait is unique and stable, which means a biometric trait can be linked to exact person and cannot naturally change a lot in a short time, the leaked biometric trait will lead to a more serious result. Furthermore, an increasing number of biometric data breach events in recent years [3, 5] greatly increase people's concern over the disclosure of their private data [20, 31], which hinders prosperity of biometric identification service.

To address this problem, many schemes have been proposed to protect the privacy of biometric data in the identification service [1, 6, 7, 12, 16, 23, 25, 28, 34]. However, these schemes merely design privacy-preserving and efficient methods for the similarity comparison of two biometric templates and few schemes focus on the efficiency of searching process. The searching process in these schemes directly traverses the whole dataset to get the identification result. Therefore, the computational costs in these schemes are linear to the size of the dataset, which will be inefficient while searching in a huge dataset.

✉ Hui Zhu
zhuhui@xidian.edu.cn

Extended author information available on the last page of the article.

In this paper, we propose an efficient and privacy-preserving cloud based biometric identification scheme, named MASK, based on the M-tree data structure and symmetric homomorphic encryption (SHE) algorithm. With MASK, the privacy of user's identification request and service provider's biometric dataset is guaranteed during the biometric identification process. Meanwhile, the computational cost of the cloud servers in searching the template dataset is far less than traverse the dataset directly. Specifically, the main contributions of this paper are threefold.

First, MASK can provide privacy-preserving biometric identification service. By introducing a SHE scheme, the similarity between two biometric templates can be securely computed with two cloud servers. Based on this, the privacy of the user's biometric template and the service provider's biometric dataset are guaranteed in the identification service.

Second, MASK can achieve efficient biometric identification. By introducing the M-tree data structure, an efficient index is built on the biometric dataset. With this index, the computational cost of the cloud servers while searching the template dataset is significantly reduced. Besides, the computational cost of building the M-tree and the communication cost of our proposed scheme also keep at a low level.

Third, to evaluate the performance of MASK, we implement our proposed scheme in Java and test the computational and communication cost on a synthetic dataset. The evaluation result shows that MASK is more efficient than other similar schemes. In addition, we also test the accuracy of our proposed scheme on a real face dataset with the face template extracted by the FaceNet algorithm. The results show that the accuracy rate is almost the same with original FaceNet algorithm.

The remainder of the paper is organized as follows. In Section 2, we first formalize our system model, security model, and identify our design goal. Then, we review some preliminaries, including an SHE scheme, FaceNet algorithm and the M-tree data structure in Section 3. After that, we present our proposed scheme in Section 4, followed by security analysis and performance evaluation in Section 5 and Section 6, respectively. Some related work is discussed in Section 7. Finally, we draw our conclusion in Section 8.

2 Models and design goal

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

2.1 System model

In our system model, we consider a cloud based biometric identification system which consists of three entities, namely service provider, cloud servers and the client, as shown in Fig. 1.

- **Service Provider:** The service provider (SP) has a set of biometric templates $\mathcal{T} = \{T_1, T_2, ..., T_n\}$ of size n and wants to provide biometric identification service to the users. Since SP may be not powerful in computing and storage, it tends to outsource the dataset to cloud for offering biometric identification service to query users. Each biometric template $T_i \in \mathcal{T}$ $(1 \leq i \leq n)$ can be denoted as an l-dimension vector $T_i = \{t_{i_1}, t_{i_2}, ..., t_{i_l}\}$. For the simplicity and clear description, we assume the value of each t_{i_j} $(1 \leq j \leq l)$ is a positive integer, since biometric template can be transformed into a positive integer vector under a given accuracy level. Figure 2 shows an example of the process of converting the template data into positive integers.

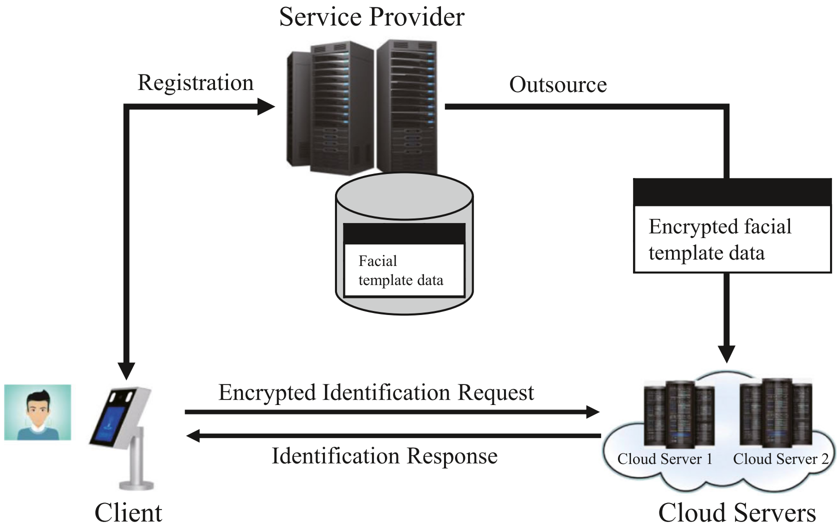
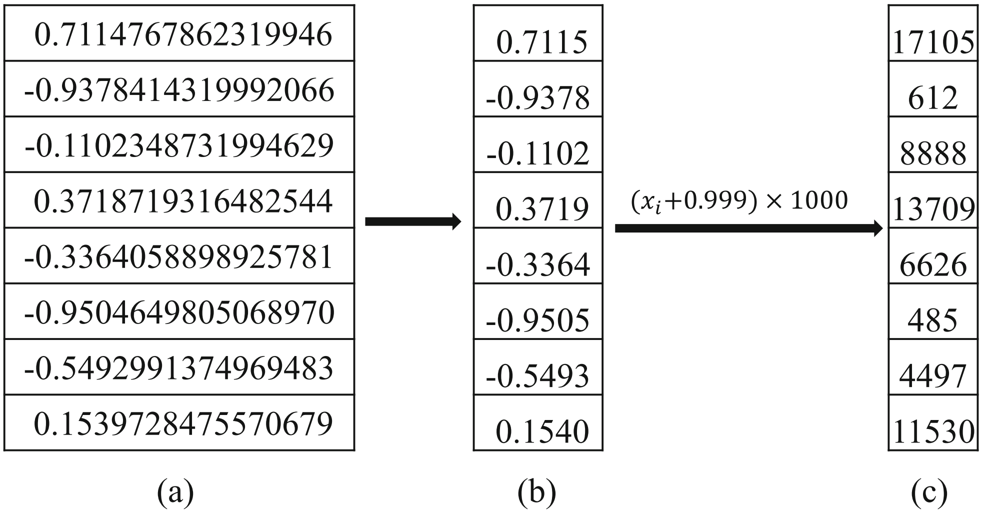


Fig. 1 System model under consideration



(a)	(b)	(c)
0.7114767862319946	0.7115	17105
-0.9378414319992066	-0.9378	612
-0.1102348731994629	-0.1102	8888
0.3718719316482544	0.3719	13709
-0.3364058898925781	-0.3364	6626
-0.9504649805068970	-0.9505	485
-0.5492991374969483	-0.5493	4497
0.1539728475570679	0.1540	11530

Fig. 2 An example of data conversion process. **a** shows a data slot from a face template extracted by FaceNet where each dimension lies in the range (-1,1). **b** shows the result of keeping four decimal places of the original data. **c** shows result of converting the data x_i into positive integer $\lfloor (x_i + 0.999) \times 1000 \rfloor$

- **Cloud Servers:** The cloud servers (CSs) are held by third parties, which are powerful in both computing and storage. In our system, there are two cloud servers, namely cloud server 1 (CS1) and cloud server 2 (CS2) working together to complete the biometric template searching. Note that, in our system model, whether two biometric templates are considered to be the same is measured by some similarity between them, e.g., the cosine similarity, Euclidean distance, etc.
- **Client:** The client in our system model is held by an application which aims to access biometric identification service. In the identification stage, the client submits an identification request to the cloud servers, which contains the biometric templates $T_r = \{t_{r_1}, t_{r_2}, ..., t_{r_l}\}$, and gets the response on whether or not $T_r \in \mathcal{T}$. Here "$T_r \in \mathcal{T}$" means that there exists a biometric template $T_j \in \mathcal{T}$ which makes the Euclidean distance $ED(T_r, T_j)$ less than or equal to a given threshold δ, i.e., $ED(T_r, T_j) = \sqrt{\sum_{i=1}^{l}(t_{r_i} - t_{j_i})^2} \leq \delta$. Note that, though our system model just considers one client, it is natural to extend one client with multiple query users.

2.2 Security model

In our security model, we consider the service provider and the client are fully trusted, while the cloud servers are honest-but-curious, which means they will faithfully execute the designed protocols, but will also try to get as much information as they can. Specifically, two cloud servers may try to obtain sensitive information from the identification request and the biometric template set $\mathcal{T}$. Meanwhile, we assume that the two cloud servers will not collude with each other. Note that, since we focus on the efficient and privacy-preserving biometric identification in this paper, active attacks on data integrity and source authentication from external adversaries are beyond the scope of our work. Those active attacks will be discussed in our future work, although it is not difficult to apply some mature digital signature and message authentication code techniques to tackle these attacks.

2.3 Design goal

Under the aforementioned system model and security model, our design goal is to propose an efficient and privacy-preserving biometric identification scheme. In particular, the following three objectives should be achieved.

- **Privacy:** The proposed biometric identification scheme should be privacy-preserving, which means the biometric data stored in the biometric template dataset and identification request should not be leaked. Specifically, the two cloud servers cannot obtain sensitive data from the identification request and the biometric templates in the database.
- **Efficiency:** The proposed biometric scheme should be efficient in terms of computational cost. There are two problems that can lead the biometric identification system impractical. On one hand, the cloud server needs to search for the target biometric template through the whole biometric template dataset at the identification stage, which will be quite time-consuming when the template dataset is large. On the other hand, in order to achieve the privacy-preserving requirements, some additional operations are introduced, which will significantly increase the computational cost. Therefore, some measures should be taken to ensure the efficiency of the proposed scheme.
- **Accuracy:** The accuracy of the proposed biometric scheme should be guaranteed. Although the privacy

and efficiency are the primary targets of our proposed scheme, they should not be achieved at the expense of accuracy, which will significantly reduce availability of the identification scheme.

3 Preliminaries

In this section, we briefly review the FaceNet face recognition system [24], symmetric homomorphic encryption (SHE) scheme [19] and the M-tree data structure [8], which will serve as the building blocks of our proposed scheme.

3.1 FaceNet face recognition system

FaceNet [24] is a face recognition system based on the deep convolutional network. In FaceNet, there are mainly two phases: the training phase and the matching phase. In the training phase, given a face image x, a mapping from a face image x to a compact Euclidean space $\mathbb{R}^d$ is built firstly. Then, based on the mapping, a Euclidean embedding $f(x) \in \mathbb{R}^d$ can be calculated for representing the face image x. In the matching phase, two face images x and y are given, which can be represented as two embeddings: $f(x) = (x_1, x_2, \ldots, x_d)$ and $f(y) = (y_1, y_2, \ldots, y_d)$. To evaluate the similarity of x and y, the squared L_2 distance of the two embeddings $f(x), f(y)$ can be computed as $D(f(x), f(y)) = \sum_{i=1}^{d}(x_i - y_i)^2$. Then, a threshold δ is used to determine whether the two faces x, y are the same (denoted as R_{same}) or different (denoted as R_{diff}). The decision process is performed as follows,

$$\begin{cases} (x, y) \in R_{same}, & \text{if } D(f(x), f(y)) \leq \delta; \\ (x, y) \in R_{diff}, & \text{if } D(f(x), f(y)) > \delta. \end{cases}$$

3.2 Description of SHE

As a symmetric homomorphic encryption scheme, SHE has been proved secure under the known-plaintext attack [19], and mainly consists of three algorithms, namely *key generation*, *encryption* and *decryption*.

- *Key Generation:* Select three security parameters (k_0, k_1, k_2) satisfying $k_1 \ll k_2 < \frac{k_0}{2}$, generate the secret key $SK = (p, q, \mathcal{L})$, where p, q are two large prime numbers with $|p| = |q| = k_0$, and $\mathcal{L}$ is a random number with the bit length $|\mathcal{L}| = k_2$. Then, compute $\mathcal{N} = pq$ and set the public parameter $PP = (k_0, k_1, k_2, \mathcal{N})$. At the same time, set the message space $\mathcal{M}$ as $\{0, 1\}^{k_1}$.
- *Encryption:* A message $m \in \mathcal{M}$ can be encrypted with the secret key $SK = (p, q, \mathcal{L})$ as

  $$c = E(m) = (r\mathcal{L} + m)(1 + r'p) \bmod \mathcal{N},$$

  where $r \in \{0, 1\}^{k_2}$ and $r' \in \{0, 1\}^{k_0}$ are two random numbers.
- *Decryption:* A ciphertext $c = E(m)$ can be decrypted with the secret key $SK = (p, q, \mathcal{L})$ as

  $$m = D(c) = (c \bmod p) \bmod \mathcal{L},$$

  The correctness of the decryption can be proven as follows.

  $$\begin{aligned} D(c) &= (c \bmod p) \bmod \mathcal{L} \\ &= (((r\mathcal{L} + m)(1 + r'p) \bmod \mathcal{N}) \bmod p) \bmod \mathcal{L} \\ &= (r\mathcal{L} + m) \bmod \mathcal{L} \qquad (\because 2k_2 < k_0) \\ &= m \qquad (\because k_1 \ll k_2) \end{aligned}$$

Given the public parameter PP, SHE will satisfy the following homomorphic properties:

- *Homomorphic Addition-I:* Given two ciphertexts $c_1 = E(m_1) = (r_1\mathcal{L} + m_1)(1 + r_1'p) \bmod \mathcal{N}, c_2 = E(m_2) = (r_2\mathcal{L} + m_2)(1 + r_2'p) \bmod \mathcal{N}$, we have $c_1 + c_2 \rightarrow E(m_1 + m_2)$.
- *Homomorphic Multiplication-I:* Given two ciphertexts c_1, c_2, we have $c_1 \cdot c_2 \rightarrow E(m_1 \cdot m_2)$.
- *Homomorphic Addition-II:* Given a ciphertext $c_1 = E(m_1) = (r_1\mathcal{L} + m_1)(1 + r_1'p) \bmod \mathcal{N}$, and a plaintext m_2, we have $c_1 + m_2 \rightarrow E(m_1 + m_2)$.
- *Homomorphic Multiplication-II:* Given a ciphertext c_1 and a plaintext m_2, we have $c_1 \cdot m_2 \rightarrow E(m_1 \cdot m_2)$.

3.3 M-tree

M-tree technique [8] is a data structure which can be used to achieve efficient similarity search in a metric space. A metric space [32] is denoted as $M = (\mathcal{D}, d)$, where $\mathcal{D}$ is a domain of feature values and d is a distance function with the following properties:

1. Symmetry. $\forall\, O_x, O_y \in \mathcal{D},\ d(O_x, O_y) = d(O_y, O_x))$.
2. Non-negativity. $\forall\, O_x, O_y \in \mathcal{D}$,

   $$\begin{cases} d(O_x, O_y) > 0, & \text{if } O_x \neq O_y; \\ d(O_x, O_y) = 0, & \text{if } O_x = O_y. \end{cases}$$
3. Triangle inequality. $\forall\, O_x, O_y, O_z \in \mathcal{D}, d(O_x, O_y) \leq d(O_x, O_z) + d(O_z, O_y)$.

Specifically, a M-tree is built on a given dataset from a metric space in a bottom-up way. There are two types of nodes in an M-tree, namely internal nodes and leaf nodes, and each node is constrained by sphere-like regions of the metric space. Both two types of nodes contain several entries and have a preset fixed capacity C. Each internal node entry can be seen as the root of a subtree and it can be denoted as $\langle O_r, d(O_r, P(O_r)), r(O_r), ptr(T(O_r))\rangle$, where O_r is the feature value of the pivot which is also an object of the dataset, $d(O_r, P(O_r))$ is the distance of O_r from the pivot of its parent, $r(O_r)$ is the covering radius of this subtree, and $ptr(T(O_r))$ is the pointer to the root of subtree $T(O_r)$. For simplicity, we call the feature value of the internal node entry's pivot as the feature value of the internal node entry. A leaf node entry contains a data object of the dataset and is denoted as $\langle O_j, d(O_j, P(O_j))\rangle$, where O_j is the feature value of the object, $d(O_j, P(O_j))$ is the distance of O_j from the pivot of its parent. Fig. 3 shows an example of a M-tree generation process based on a given dataset obtained from a metric space with Euclidean distance.

The operations of the M-tree mainly consist of insertion, tree-building and range query.

- *Insertion:* When a data point $x \in D$ needs to be inserted, it is recursively descended through the M-tree to locate the most suitable leaf node. The insertion process at each node is shown in Algorithm 1.
- *Tree-building*: When we start to build a M-tree on the given dataset D, we add the first data object $x \in D$ to the empty M-tree and built a M-tree with a root node and a leaf node. Then we run the insertion algorithm to add all the remaining data objects to the M-tree.
- *Range Query:* M-tree supports both KNN query and range query. The query range algorithm consists of two processes, namely the subtree pruning process and the data object verification process. The subtree pruning process aims to prune some redundant subtrees to improve the query efficiency, while the data object verification process concentrates on verifying whether an object satisfies the query requirements. In the query range algorithm, a range query (q, r) means to find out whether there exists an object p in the dataset satisfying $d(p, q) \leq r$, and the range query algorithm at a node is described in Algorithm 2. Since the distance to the parent pivot has no meaning for the root node, it will be treated as 0 in the pruning process.

Algorithm 1 The insertion process.

```
Input: An M-tree node $N$, a new data object $x$
Let $\mathcal{N}$ be the set of entries in node $N$.
if $N$ is a internal node then
    Let $N_{in}$ be the entries that satisfy
    $d(O_r, x) \leq r(O_r)$.
    if $N_{in} \neq \emptyset$ then
        Let $N_{next} \in N_{in}$, with minimum $d(O_r, x)$.
    else
        Let $N_{next} \in N_{in}$, with minimum
        $d(O_r, x) - r(O_r)$. Let $r(O_r) = d(O_r, x)$.
    end
    Insert($ptr(O_r), x$)
else
    if $N$ is not full then
        Store $x$ in leaf node $N$.
    else
        Add $x$ to node $N$, split $N$ into two new leaf
        nodes and add a new routing object in the
        parent node.
    end
end
```

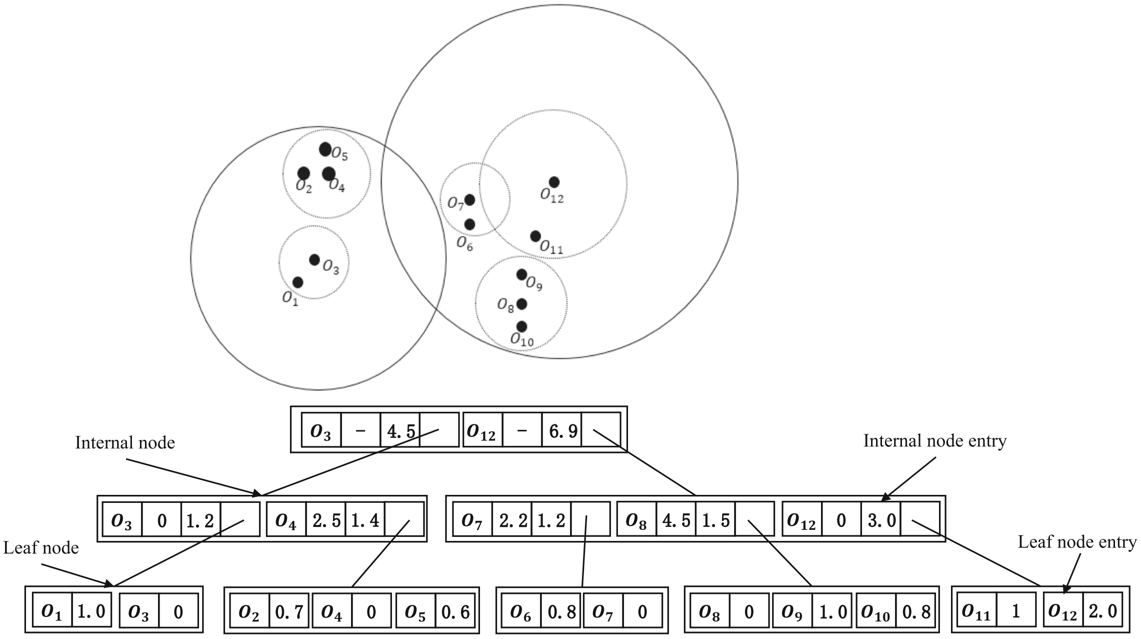


Fig. 3 An example of a M-tree built on a given dataset

```
Algorithm 2 The range query process at a node.
Input: An M-tree node $N$, a range query $(q, r)$
Let $O_p$ be the parent pivot of $N$.
if $N$ is a internal node then
    for each internal node entry $O_r$ in $N$ do
        if $|d(O_p, Q_r) - d(q, O_p)| \le r(O_r) + r$ then
            Compute $d(O_r, q)$.
            if $d(Q_r, q) \le r(O_r) + r$ then
                $RangeQuery(ptr(O_r), (q, r))$.
            end
        end
    end
else
    for each leaf node entry $O_j$ in $N$ do
        if $|d(O_p, q) - d(O_j, O_p)| \le r$ then
            Compute $d(O_j, q)$. if $d(O_j, q) \le r$ then
                Add $O_j$ to the result.
            end
        end
    end
end
```

4 Our proposed scheme

In this section, we will present our privacy-preserving M-tree based biometric identification scheme (MASK), which mainly consists of four phases: 1) *System initialization*; 2) *M-tree Building and Encryption*; 3) *Encrypted Identification Request Generation*; 4) *Biometric Identification*. For a clear description, the overview of our proposed scheme is first shown in Fig. 4. At first, SP generates system parameters and distributes them to the client and cloud servers. Then, SP builds a M-tree based on the biometric template dataset, encrypts it with the SHE algorithm and outsources it to CS1. In the identification stage, the client sends the encrypted identification request to CS1, and two cloud servers work together to get the identification result and return it to the client.

4.1 System initialization

In the system initialization phase, SP generates the system parameters, publishes the public parameters and distributes cryptographic parameters to the client and CSs. Specifically, SP chooses the security parameters (k_0, k_1, k_2), generates the secret key $SK = (p, q, \mathcal{L})$ and sets the public parameters $PP = (k_0, k_1, k_2, \mathcal{N})$ for the SHE algorithm, where $\mathcal{N} = p \cdot q$. Then, SP calculates the ciphertexts for numbers 0 and 1 with the secret key SK, and we denote them as $E(0)$ and $E(1)$. Next, SP sets the identification threshold δ, which indicates two biometric templates will be considered the same if the distance between two templates is under δ. After all parameters are generated, SP publishes $\{PP, \delta\}$, sends $\{E(0), E(1)\}$ to the client and transmits the secret key SK to CS2.

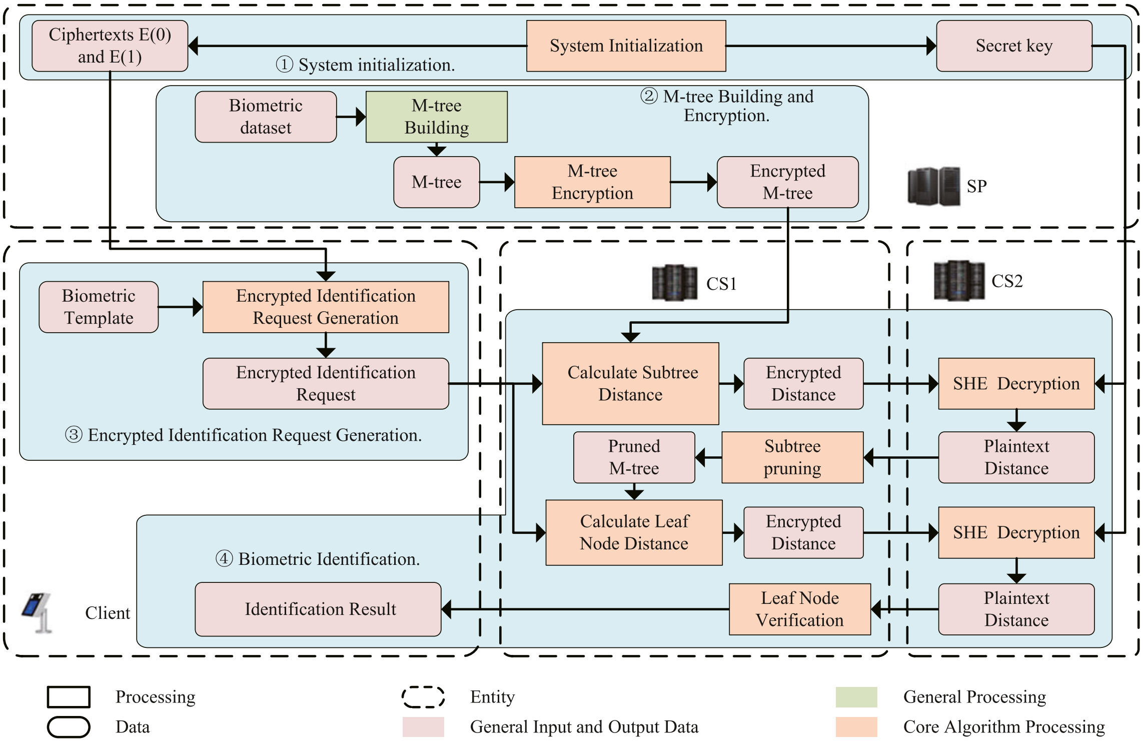


Fig. 4 Overview of our proposed scheme

4.2 M-tree building and encryption

In this phase, SP builds a M-tree based on the biometric template dataset $\mathcal{T}$ at first. Then, SP encrypts the M-tree with the secret key SK. Eventually, SP transmits the encrypted M-tree to CS1.

- **Stage 1: M-tree building**

At first, SP builds a M-tree over the biometric dataset $\mathcal{T} = \{T_1, T_2, ..., T_n\}$ following the M-tree building algorithm. In the M-tree, all biometric templates are stored in the internal node entry and leaf node entry, and we denote the feature value of the leaf node entry and internal node entry as $T_i = \{t_{i_1}, t_{i_2}, ..., t_{i_l}\}$ and $O_i = (o_{i1}, o_{i2}, \cdots, o_{il})$, respectively. Then, SP calculates the square of modulus-length of the feature value of each node entry, where $|O_i|^2 = \sum_{j=1}^{l} O_{ij}^2$ and $|T_i|^2 = \sum_{j=1}^{l} t_{ij}^2$. Next, SP expands each internal node entry and leaf node entry by mounting $|O_i|^2$ and $|T_i|^2$ to corresponding entry. The expanded internal node entry and leaf node entry are represented as $\langle O_i, d(O_i, P(O_i)), r(O_i), ptr(T(O_i)), |O_i|^2\rangle$ and $\langle T_i, d(T_i, P(T_i)), |T_i|^2\rangle$, respectively.

- **Stage 2: M-tree encryption**

After the M-tree is built, SP encrypts the M-tree using SHE. Specifically, each expanded internal node entry and leaf node entry are encrypted with SK. The encrypted internal node entry and leaf node entry are denoted as $\langle E(O_i), d(O_i, P(O_i)), r(O_i), ptr(T(O_i)), E(|O_i|^2)\rangle$ and $\langle E(T_i), d(T_i, P(T_i)), E(|T_i|^2)\rangle$, respectively. After encrypting the M-tree, SP outsources the encrypted M-tree to CS1.

4.3 Encrypted identification request generation

The client has a face template and wants to verify whether the template T_r exists in the biometric dataset $\mathcal{T}$. To get the identification result, the client needs to send T_r to the cloud server as an identification request. At first, the client computes the square of modulus-length of the face template $|T_r|^2 = \sum_{i=1}^{l} t_{ri}$. Then the client encrypts $|T_r|^2$ and T_r with SHE. Since the client does not have the secret key SK, the encryption is completed with $E(0)$ and $E(1)$ according to the homomorphic properties of SHE. For the template $T_r = \{t_{r1}, t_{r2}, \cdots, t_{rl}\}$, the ciphertext of T_r is calculated as

$$E(T_r) = (E(t_{r1}), E(t_{r2}), \cdots, E(t_{rl})),$$

where

$$E(t_{ri}) = E(1 \cdot t_{r_i} + 0 \cdot r_i) = E(1) \cdot t_{ri} + E(0) \cdot r_i,$$

and $r_i \in \{0, 1\}^{k_2}$ is a random number. Then, the client encrypts $|T_r|^2$ by

$$E(|T_r|^2) = E(1 \cdot |T_r|^2) + E(0 \cdot r_{t_r}) = E(1) \cdot |T_r|^2 + E(0) \cdot r_{t_r},$$

where $r_{t_r} \in \{0, 1\}^{k_2}$ is a random number. When the encryption is completed, the client sends $\langle E(T_r), E(|T_r|^2)\rangle$ to CS1.

4.4 Biometric identification

In this section, CS1 and CS2 work together to search T_r in the M-tree to get the identification result. The search process consists of two stages, i.e., the subtree pruning stage and the leaf node verification stage.

- **Stage 1: Subtree Pruning Process**

Upon receiving the identification request, CS1 and CS2 collaboratively prune the subtrees which do not have an intersection with the query range at first. The subtree pruning process follows the pruning algorithm of M-tree and starts from the root node of the M-tree. For an encrypted internal node entry $\langle E(O_i), d(O_i, P(O_i)), r(O_i), ptr(T(O_i)), E(|O_i|^2)\rangle$, CS1 calculates

$$E(d(O_i, T_r)^2) = E(|O_i|^2) + E(|T_r|^2) - 2\sum_{j=0}^{l} E(o_{ij})E(t_{rj})$$

to confirm whether $O_i's$ subtree O_j has an intersection with the query range. Since $E(|O_i|^2)$ and $E(o_{ij})$ are stored in the encrypted M-tree and $E(|T_r|^2)$ and $E(t_{ri})$ can be obtained from the identification request, it is easy for CS1 to get the result. After that, CS1 sends $E(d(O_i, T_r)^2)$ to CS2 to get the plaintext. Since CS2 has obtained the SK from SP in the *System initialization* stage, CS2 can decrypt the ciphertext. After getting the plaintext from CS2, CS1 checks whether

$$|d(O_j, O_i) - d(O_i, T_r)| > r(O_j) + \delta$$

holds, where $d(o_j, o_i) = d(o_j, P(o_j))$. If it does, it means that this subtree does not intersect with the query range and will be pruned from the M-tree. Since the root node does not have a parent node, the judgment condition of an entry in the root node is $d(O_i, T_r) > r(O_i) + \delta$.

After finishing the pruning process, all the subtrees that do not intersect with the query range have been pruned from the M-tree.

- **Stage 2: Leaf Node Verification Stage**

To verify whether T_r exists in the pruned M-tree, two cloud servers work together to traverse all remained leaf node entries. We take the verification process of one leaf node entry as an example. Assume there is an encrypted leaf node entry $\langle E(T_i), d(T_i, P(T_i)), E(|T_i|^2)\rangle$, CS1 computes the ciphertext of the square of the distance between T_i and T_r by

$$E(d(T_i, T_r)^2) = E(T_i)^2 + E(T_r)^2 - 2\sum_{j=0}^{l} t_{ij} \cdot t_{rj}.$$

Then CS1 sends $E(d(T_i, T_r)^2)$ to CS1 to get the plaintext. While receiving $E(d(T_i, T_r)^2)$, CS2 decrypts it with the secret key SK, and returns $d(T_i, T_r)^2$ to CS1. After getting $d(T_i, T_r)^2$, CS1 calculates the positive square root of $d(T_i, T_r)^2$ to get $d(T_i, T_r)$. Eventually, CS1 checks whether this leaf node entry satisfies the query requirements by judging whether $d(T_i, T_r) < \delta$ holds. If it does, it means this leaf node entry satisfies the requirements, and CS1 returns the result that template T_r exists in dataset $\mathcal{T}$ to the client. Inversely, if all the remained leaf node entries are checked but none of them satisfies the requirements, CS1 returns the result that T_r does not exist in the dataset.

5 Security analysis

In this section, we will analyze the security of our proposed privacy-preserving biometric identification scheme, which mainly focuses on the privacy-preserving properties. Specifically, the biometric templates and user's identification request should be privacy-preserving.

5.1 The privacy of the biometric templates

According to our security model in Section 2, keeping the privacy preservation in biometric templates means to prevent the cloud servers from obtaining the plaintext of the biometric template. In MASK, only the *M-tree Building and Encryption* and *Biometric Identification* stages are related to the biometric template.

In the *M-tree Building and Encryption* stage, SP builds a M-tree based on the biometric template dataset $\mathcal{T}$, encrypts the M-tree using the SHE algorithm and outsources the encrypted M-Tree to CS1. Since SP is trusted, the privacy of the biometric templates in the M-tree building and encrypting process is guaranteed. The M-tree sent to CS1 is stored in the encrypted form. Specifically, each internal node entry and leaf node entry is encrypted and stored in the form $\langle E(O_i), d(O_i, P(O_i)), r(O_i), ptr(T(O_i)), E(|O_i|^2)\rangle$ and $\langle E(T_i), d(T_i, P(T_i)), E(|T_i|^2)\rangle$, respectively. Note that all ciphertexts are generated by the SHE cryptosystem which has been proven to be secure against the known-plaintext attack in [19]. The security of the SHE cryptosystem guarantees that the adversary has no idea on the plaintext without knowing the secret key $SK = (p, q, \mathcal{L})$. Since CS1 does not have the private key SK and CS2 can not get the ciphertext of the encrypted node entry, both two cloud servers can not decrypt the feature value based on the assumption that two cloud servers will not collude. Therefore, the privacy of the biometric templates is guaranteed in the *M-tree Building and Encryption* stage.

In *Biometric Identification* stage, after receiving a identification request from a client, CS1 and CS2 work together to search in the M-tree. Specifically, CS1 computes the encrypted distance between the encrypted T_r and some entries in the M-tree. Then CS1 sends the encrypted distance to CS2. CS2 decrypts the distance and returns the plaintext to CS1. While getting the plaintext of the distance, CS1 continues the subtree pruning or the leaf node verification to complete the identification process. In this stage, CS1 can obtain the encrypted biometric template, the encrypted identification request and the distance between them, while CS2 can only get the distance between the biometric template and identification request. According to the system model, only CS2 knows the secret key and two cloud servers do not collude. Since the biometric template and identification request are two high dimension vectors, CS1 can obtain neither plaintext of the biometric template nor T_r only through the distance between them. As CS2 only gets distance between T_r and some entries, she/he can infer nothing about the specific data of them based on this.

Besides, since the M-tree is built based on the Euclidean distance between the biometric templates, CS1 can know which templates are closer to each other. However, all the template stored in the M-tree are encrypted by the SHE algorithm, and CS1 can not infer the specific data of the templates. To sum up, the biometric dataset is privacy-preserving in our proposed scheme.

5.2 The privacy of the identification request

According to our security model in Section 2, protecting the privacy of the identification request means to prevent the cloud servers from getting the plaintext of the T_r. In MASK, the biometric identification request is only processed in the *Encrypted Identification Request Generation* and *Biometric Identification* stages.

In the *Encrypted Identification Request Generation* stage, the client encrypts T_r based on the homomorphic property of the SHE algorithm, then the client sends $\langle E(T_r), E(|T_r|^2)\rangle$ to CS1. Since the secret key SK is only known by CS2, the encrypted identification request $\langle E(T_r), E(|T_r|^2)\rangle$ is only sent to CS1 and the two cloud servers will not collude with each other, both the two cloud servers can not get the plaintext of the identification request in this process. Therefore, the privacy of the identification request is guaranteed in the *Encrypted Identification Request Generation* stage.

In the *Biometric Identification* stage, both CS1 and CS2 can obtain the distance between T_r with some biometric templates in the dataset $\mathcal{T}$. CS1 knows the ciphertexts of the identification request and these biometric templates, but it can neither decrypt the ciphertext nor conclude the specific data of the identification request. CS2 has the secret key SK, but it can get neither the ciphertext nor the plaintext of these templates, hence CS2 can not get the identification request. Therefore, the privacy of the identification request is guaranteed in the *Biometric Identification* stage.

In addition, when the client submits two identification requests from the same person in two identification service, CS1 may find these two identification requests are from the same person. However, CS1 still can not get the specific data of the identification request or the biometric template. From the above, the biometric identification is privacy-preserving in MASK.

6 Performance evaluation

In this section, we evaluate the performance of MASK in terms of computational cost, communication cost and the identification accuracy and make the comparison with [29]. In order to have a better evaluation of the scheme, we consider two face datasets: a real face dataset and a synthetic dataset. The real dataset [10] is provided by University of Essex and it is used to test the accuracy of our proposed scheme on a real dataset. The synthetic dataset is a randomly generated dataset which is used to test the performance.

6.1 Evaluation Environment

In order to measure the integrated performance, we implement both schemes with Java and run our experiments on an Intel Core i7-8750H CPU@2.1 GHz Windows Platform with 16GB RAM. Specifically, we set the parameters of the SHE scheme as $k_0 = 4096$, $k_1 = 70$ and $k_2 = 2000$ and the capacity of the M-tree node as $C = 50$. And two datasets are prepared as follows.

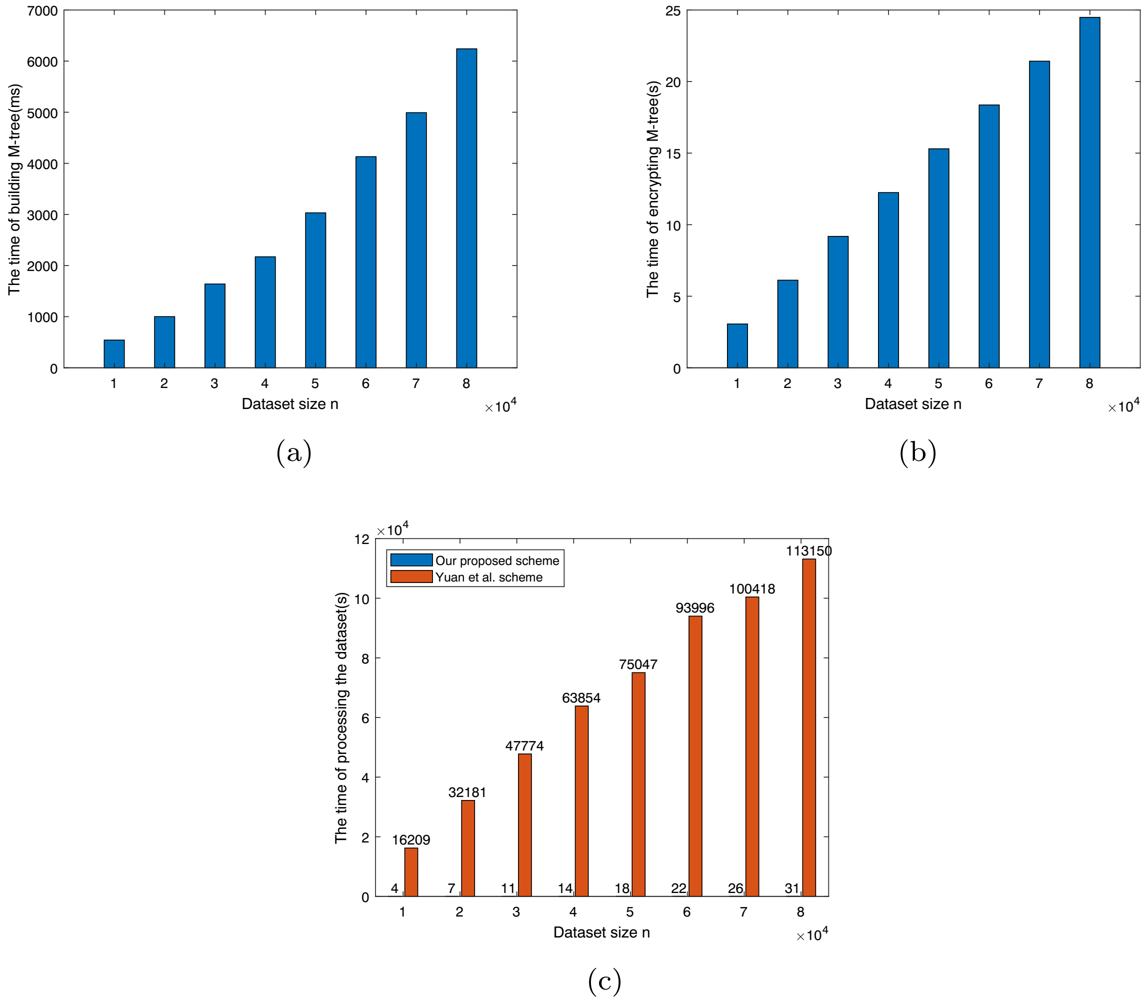
- **Real Dataset**. A dataset contains face images collected from 153 individuals (including 113 male students, 20 male students and 20 stuffs) and there are 20 images per individual. In this paper, we use the FaceNet algorithm to extract face features from these face images at first. Each face feature is a 512-dimension vector and all the face features live on the same hypersphere which means each of the dimensions of the vector is in the range [-1,1] and the sum the squared of each dimension is equal to 1.
- **Synthetic Dataset**. We randomly generate a synthetic dataset which contains 8×10^4 face features. Each face feature is a 512-dimension vector and all face features lie in the same range $(-1, 1)$ as the face feature extracted by the FaceNet.

6.2 Computational cost

In this section, we will evaluate the computational cost of MASK while processing the biometric template dataset, encrypting the identification request and searching the biometric templates which are corresponding to the computational cost in *M-tree Building and Encryption*, *Encrypted Identification Request Generation* and *Biometric Identification*, receptively. We analyze each computational cost at first and test it over the synthetic dataset. For the sake of simplicity, we denote the computational cost of the modular addition and the big integer multiplication as C_{m-add} and C_{m-mul}, respectively. Since [29] is designed based on the transformation of matrix, we denote the computational cost of the integer multiplication and integer addition as C_{i-add} and C_{i-mul}, respectively. In addition, we assume the size of the dataset is n and the length of the each face template is l. The computational cost of each phase of the two schemes is evaluated as follows.

- **Computational cost of processing the biometric dataset.** In our proposed scheme, the service provider takes charge of the dataset processing. In the M-tree building and encryption stage, SP builds a M-tree on the plaintext of the biometric template dataset $\mathcal{T}$. Since the construction of the M-tree is executed by inserting each template in the $\mathcal{T}$ into the M-tree, the computational cost of SP while building the M-tree is $\mathcal{O}(n)$. In the M-tree encryption, SP needs to encrypt all the internal node entries and leaf node entries in the M-tree. According to the encryption algorithm of SHE, 3 big integer multiplication and 2 modular addition are required while encrypting a plaintext. There are n leaf node entries and $\sum(\lceil\frac{n}{C^w}\rceil)$, where $C^{w+1} > n$ and $C^w < n$ internal node entries in the M-tree. The feature value of the leaf node entry and the pivot of the internal node entry are both l dimension vectors. The computational cost of encrypting the M-tree is less than $(n+\sum(\lceil\frac{n}{C^w}\rceil))(3C_{m-mul}+2C_{m-add})l$, where $C^{w+1} > n$ and $C^w < n$. Figure 5a and b show the computational cost of building and encrypting the M-tree varies with dataset size n, respectively.

  In [29], the encryption of the dataset consists of two steps. In the first step, the data owner hides each biometric template using random matrix. To hide a biometric template, the data owner needs to compute $(l + 1)^2$ integer multiplication. Then, the data owner encrypts each biometric template using the matrix transformation. The data owner needs $2(l + 1)^2(l + 3)$ integer multiplication and $2l(l + 1)(l + 3)$ integer addition while encrypting each biometric template. Hence, the computational cost of the data owner while encrypting the dataset in [29] is $2n(l + 1)^3C_{r-mul} + 2nl(l + 1)^2C_{r-add}$. Figure 5c shows the integrated



(a)

(b)

(c)

Fig. 5 The computational cost of processing the biometric dataset

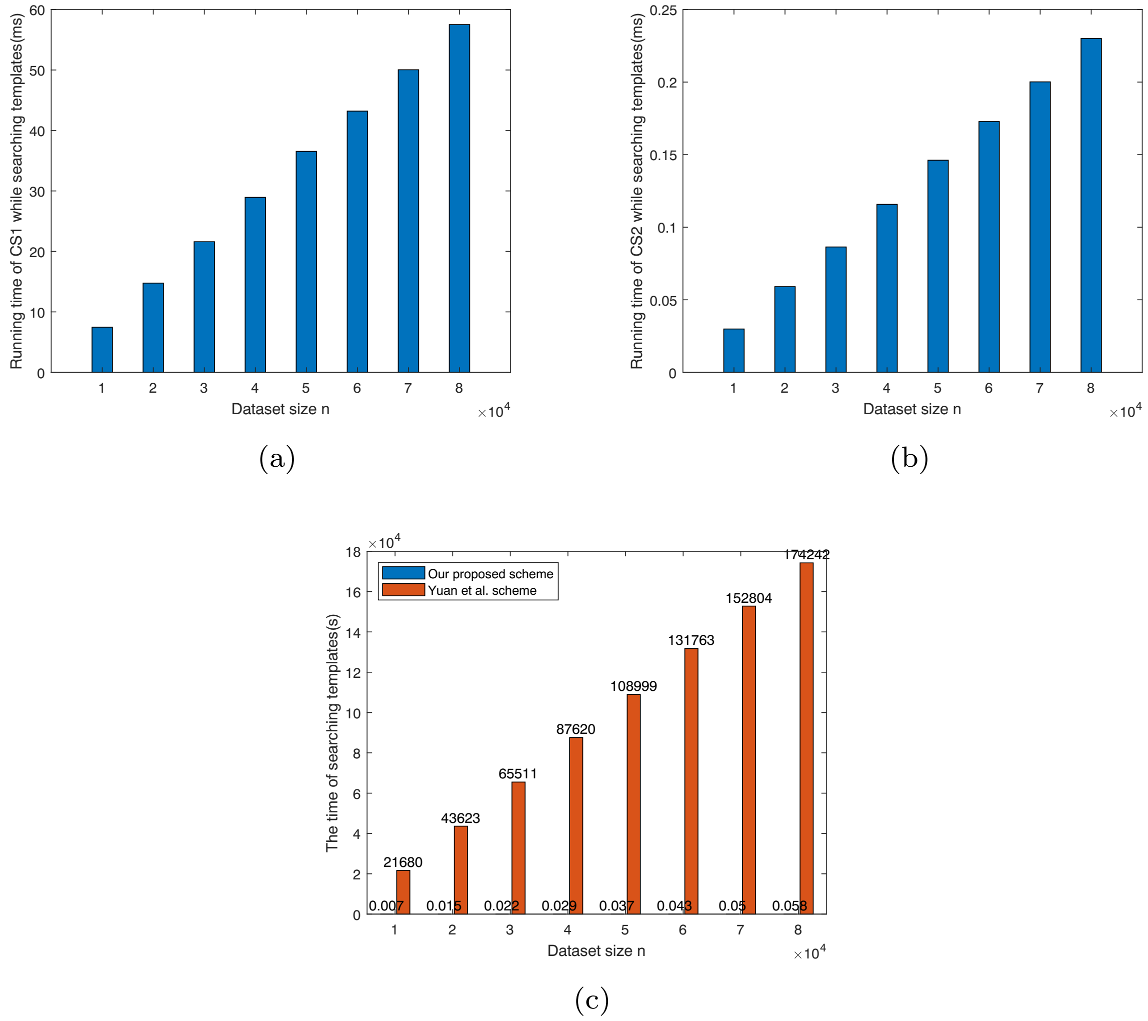
running time of process the biometric dataset in both two schemes. It can be seen that the computational cost in MASK is much lower than that in [29].

- **Computational cost of encrypting the identification request** In our proposed scheme, the client encrypts the identification request before sending it to the cloud servers. In the encrypted identification request generation stage, the client encrypts each dimension of the face feature using two ciphertexts $E(0)$ and $E(1)$. 2 big integer multiplication and 1 modular addition is needed while encrypting each dimension of the face feature and there are $l + 1$ data to be encrypted. Therefore, the computational cost of generating the encrypted identification request is $(l+1)(2C_{m-mul} + C_{m-add})$.

  In [29], the data owner encrypts the identification request and sends it to the cloud server. The data owner blinds the identification request at first, and then encrypts it with matrix. $(l+1)^2$ integer multiplication is needed in the blinding stage and $(l+1)^3$ integer multiplication and $(l(l+1)^2)$ integer addition are required in the encrypting stage. Therefore, the computational cost of encrypting the identification request is $(l+1)^2(l+2)C_{i-mul} + l(l+1)^2C_{i-add}$.

- **Computational cost of biometric identification**

  In our proposed scheme, two cloud servers work together to find the closest match biometric template with the identification request in the M-tree in the biometric identification stage. The computational cost of MASK in this stage depends on distribution of the templates on the M-tree, where the computational cost consists of $(\frac{n}{C} + \log_C)$ and $n + \sum(\lceil \frac{n}{C^w} \rceil)$, where $(C^{w+1} > n$ and $C^w < n)$, distance calculation operation under the best and worst case respectively. Figure 6a and b show the computational cost of searching for the template of CS1 and CS2 varying with dataset size n under the synthetic dataset respectively.



(a)

(b)

(c)

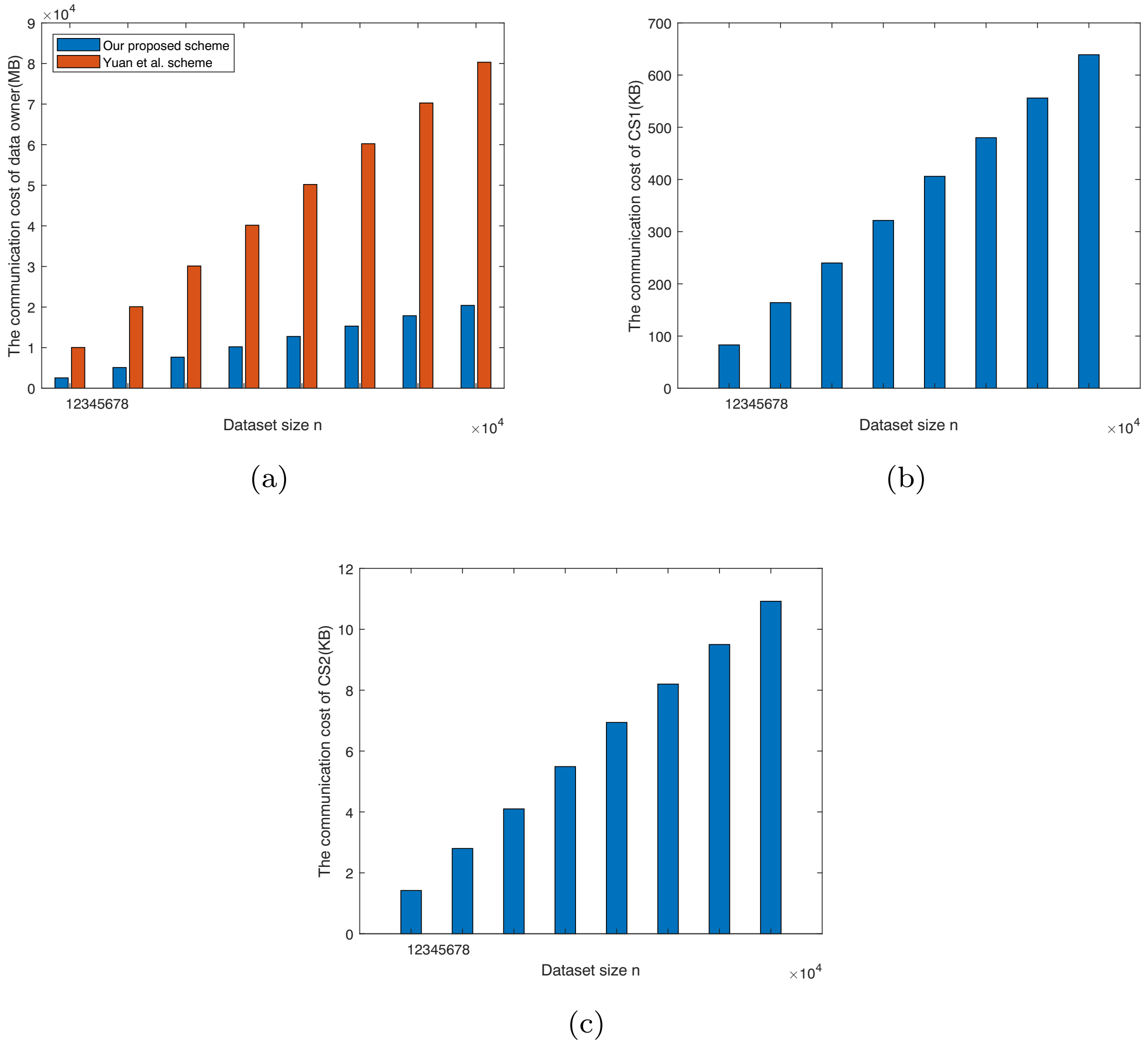
Fig. 6 The computational cost of cloud servers

In [29], the cloud server traverses the encrypted dataset to find the closest matching with the identification request. $(l+1)^2(l+3)$ integer multiplication and $l(l+1)(l+3)$ integer addition are required while calculating the distance between the identification request and a biometric template in the dataset. The computational cost of the cloud server while traversing the biometric dataset is $n(l+1)^2(l+3)C_{i-mul} + nl(l+1)(l+3)C_{i-add}$. The comparison of the integrated computational cost of two schemes while searching the biometric templates is shown in Fig. 6c.

6.3 Communication cost

In this section, we will evaluate the communication cost of our proposed scheme in sending the encrypted biometric template dataset, transmitting the encrypted identification request and searching the biometric templates which are corresponding to the communication in *M-tree Building and Encryption*, *Encrypted Identification Request Generation* and *Biometric Identification* receptively. We analyze the communication cost at first and test it over the synthetic dataset. For the sake of simplicity, we denote the bit length of a integer as L_i. Later, we test the communication cost on the synthetic dataset. Specifically, we set the integer bit length $L_i = 32$ in the experiment.

- **The communication cost of sending the encrypted biometric dataset** In the M-tree build and encryption stage, SP generates a M-tree on the plaintext of the biometric dataset $\mathcal{T}$ and encrypts all the node entries in the M-tree. According to the SHE, the size of the ciphertext is k_0 bits. Since there are n biometric templates and at most $\lceil \frac{n}{C} \rceil w$, where $C^{w+1} > n$ and $C^w < n$ and the feature value of the leaf node entry and the pivot of the internal node entry are both l dimension vectors, the size of the encrypted M-tree is $k_0(n + \lceil \frac{n}{C} \rceil w)l$ bits.



(a)

(b)

(c)

Fig. 7 The communication cost of each stage

In [29], after the encryption process, each biometric template is represented by a $(l + 1) \times (l + 1)$ matrix and there are n biometric templates in the biometric dataset. Hence, the communication cost of sending the encrypted biometric dataset is $n(l + 1)^2 L_i$. Figure 7a shows the communication cost of sending the encrypted biometric dataset changes with dataset size n in both two schemes.

– **The communication cost of sending the encrypted identification request**

In the encrypted identification request generation stage, the client encrypts identification request and sends it to the cloud server. Since the face feature in the identification request is l dimension, there are $l + 1$ ciphertext to be sent. The ciphertext of SHE is k_0 bits and ciphertext sent to the cloud servers is $(l + 1)k_0$ bits.

In [29], the encrypted identification request is denoted as a $(l + 1)^2$ matrix and communication cost of sending the encrypted identification request is $(l + 1)^2 L_i$.

– **The communication cost of searching the biometric template**

In the biometric identification stage, two cloud servers work together to searching on the M-tree to find closest match of the identification request. In the subtree pruning process and leaf node verification stage, CS1 sends ciphertexts to CS2 to get the corresponding plaintexts. The communication cost of our proposed scheme in this stage depends on distribution of the templates on the M-tree, where the communication cost consists of $(\frac{n}{C} + \log_C)$ and $n + \sum(\lceil \frac{n}{C^w} \rceil)$, where $(C^{w+1} > n$ and $C^w < n)$, interaction between CS1 and CS2 under the best and worst case respectively. Figure 7b and c show the communication cost of CS1 and CS2 varying with dataset size n respectively. Since there is only one cloud server in [29], no communication cost are generated in this stage. It can be seen that the communication cost of sending the encrypted dataset in MASK is much lower than that in [29].

6.4 Accuracy

In our proposed scheme, No error is introduced by the encryption scheme or the M-tree data structure. The only reason that may lead to the similarity between the identification request and biometric templates in the dataset being changed is the process of converting the face template data into positive integers, but this error is quite trivial. Therefore, the accuracy of our proposed scheme stays almost the same as the FaceNet algorithm. We test the accuracy of our proposed scheme and the original FaceNet algorithm over the real dataset [10], the result shows that the accuracy rate is almost the same as original FaceNet algorithm.

7 Related work

In this section, we will briefly review some related work on the privacy-preserving biometric identification.

In the early, privacy-preserving biometric identification schemes are designed under a two-party model, where the matching process between the candidate template and the template in the data set is directly executed by the data owner. Huang et al. [15], Blanton et al. [4] and Barni et al. [2] proposed three party privacy-preserving biometric identification schemes based on the secure computation protocol. Hirnao et al. [13], Higo [12] and [21] proposed three privacy-preserving biometric identification schemes based on the homomorphic encryption scheme. These two-party schemes mainly focus on how to achieve privacy-preserving biometric template matching, and the efficiency is not well guaranteed. Specifically, Since the matching process is completed on the data owner, it requires that the data owner should be equipped with strong computing power which can not be satisfied in many cases.

In order to release the data owner from heavy computational cost burden, some schemes are presented in an outsourced environment, where the data owner outsources the encrypted biometric data set to the cloud server and the matching process is completed on the cloud. Yuan et al. [29] proposed the first cloud based privacy-preserving biometric identification scheme using a matrix encryption scheme, where the biometric data set and identification query are both encrypted and sent to the cloud server by the data owner. However, Wang et al. [26] and Zhu et al. [36] pointed out that [29] is not secure under the known-plaintext attack model [9]. In addition, [26] presented a privacy-preserving biometric identification scheme with a new biometric data encryption based on the similarity matrix under the same system model in [29] and the security analysis showed [26] had a higher security level than [29]. Zhang et al. [33] proposed an efficient privacy-preserving biometric identification scheme based the matrix and perturbed terms with lower time cost and bandwidth consuming than [29] and [26]. Wang et al. [27] proposed an inference-based framework for privacy-preserving similarity search in Hamming space and achieved privacy-preserving biometric identification based on it. Hu et al. [14] proposed a privacy-preserving biometric identification scheme in outsourcing environment with two non-colluded servers based on the homomorphic encryption and batched protocols. With the help of the cloud, the computing cost of data owner during the biometric matching is significantly reduced in the above schemes. However, in [26, 29, 33], the data owner has to keep online to encrypt the user's query data and decrypt the identification result, which whittles some advantages of the cloud computing away and leads heavy load to the data owner if it serves too many users at the same time. What's more, in all the cloud based schemes above, the searching process is not optimized which means the searching cost of the cloud server is linear with the size of the data set. Despite the cloud server is equipped with strong computing power, it may still run into bottleneck while simultaneously servering too many users.

To address this issue, some researchers begin to focusing on how to reducing the researching time to sub-linear which will significantly ease the pressure of cloud server. Zhu et al. [35] proposed an cloud-assisted privacy-preserving biometric identification scheme. With the help of an asymmetric scalar-product preserving encryption scheme and R-tree, sub-linear search efficiency is achieved in [35]. Nevertheless, the data owner also needs to be keep online in [35]. Since R-tree is not constructed based on the metric relation between the data objects, the cloud server needs to traverse the tree for twice to find the closest biometric template in the data set, which reduces the efficiency of the searching process.

In this paper, to protect the security of the biometric data and reduce the time cost in the biometric searching process, we introduce SHE and M-tree to construct a privacy-preserving biometric identification scheme based on two no-colluded cloud servers. Compared with previous works, the service provider in our proposed scheme does not need to keep online in the identification scheme and the efficient identification service is achieved.

8 Conclusion

In this paper, we proposed an efficient and privacy-preserving M-tree based biometric identification scheme, named MASK. By introducing the SHE scheme, the privacy of the user's identification request and the service provider's dataset is guaranteed. Based on the M-tree data structure, the computational cost of the cloud servers is significantly reduced. Detailed security analysis showed the security of our proposed scheme, and extensive experiments were conducted to demonstrate its efficiency in terms of computational and communication costs.

Acknowledgment This work was supported in part by National Key Research and Development Program of China (2017YFB0802200), National Natural Science Foundation of China (61972304,61932015), Natural Science Foundation of Shaanxi Province (2019ZDLGY12-02),Technical Research program of the Public Security Ministry (2019JSYJA01).

References

1. Abidin A (2016) On privacy-preserving biometric authentication. In: Chen K, Lin D, Yung M (eds) Information Security and Cryptology - 12th International Conference, Inscrypt 2016, Beijing, China, November 4-6, 2016, Revised Selected Papers, *Lecture Notes in Computer Science*, vol 10143. Springer, pp 169–186
2. Barni M, Droandi G, Lazzeretti R (2015) Privacy protection in biometric-based recognition systems: A marriage between cryptography and signal processing. IEEE Signal Process Mag 32(5):66–76
3. BBC News India aadhaar id cards: Collecting biometric data from 1bn people. [EB/OL]. https://www.bbc.com/news/world-asia-40371523
4. Blanton M, Gasti P (2011) Secure and efficient protocols for iris and fingerprint identification. In: Atluri V, Díaz C (eds) Computer Security - ESORICS 2011 - 16th European Symposium on Research in Computer Security, Leuven, Belgium, September 12-14, 2011. Proceedings, *Lecture Notes in Computer Science*, vol 6879. Springer, pp 190–209
5. Baraniuk C Biostar security software 'leaked a million fingerprints'. [EB/OL]. https://bbc.com/news/technology-49343774
6. Chen L, Zhang K (2021) Privacy-aware smart card based biometric authentication scheme for e-health. Peer Peer Netw. Appl. 14(3):1353–1365
7. Chun H, Elmehdwi Y, Li F, Bhattacharya P, Jiang W (2014) Outsourceable two-party privacy-preserving biometric authentication. In: Moriai S, Jaeger T, Sakurai K (eds) 9th ACM Symposium on Information, Computer and Communications Security, ASIA CCS '14, Kyoto, Japan - June 03 - 06, 2014, pp 401–412. ACM
8. Ciaccia P, Patella M, Zezula P (1997) M-tree: An efficient access method for similarity search in metric spaces. In: Jarke M, Carey MJ, Dittrich KR, Lochovsky FH, Loucopoulos P, Jeusfeld MA (eds) VLDB'97, Proceedings of 23rd International Conference on Very Large Data Bases, August 25-29, 1997, Athens, Greece, pp 426–435. Morgan Kaufmann
9. Delfs H, Knebl H (2015) Introduction to Cryptography - Principles and Applications, Third Edition. Information Security and Cryptography Springer
10. of Essex U Description of the collection of facial images. [EB/OL]. https://cswww.essex.ac.uk/mv/allfaces/index.html/
11. Fianyi I, Zia TA (2016) Biometric technology solutions to countering today's terrorism. Int J Cyber Warf Terror 6(4):28–40
12. Higo H, Isshiki T, Mori K, Obana S (2015) Privacy-preserving fingerprint authentication resistant to hill-climbing attacks. In: Dunkelman O, Keliher L (eds) Selected Areas in Cryptography - SAC 2015 - 22nd International Conference, Sackville, NB, Canada, August 12-14, 2015, Revised Selected Papers, *Lecture Notes in Computer Science*, vol 9566. Springer, pp 44–64
13. Hirano T, Hattori M, Ito T, Matsuda N (2013) Cryptographically-secure and efficient remote cancelable biometrics based on public-key homomorphic encryption. In: Sakiyama K, Terada M (eds) Advances in Information and Computer Security - 8th International Workshop on Security, IWSEC 2013, Okinawa, Japan, November 18-20, 2013, Proceedings, *Lecture Notes in Computer Science*, vol 8231. Springer, pp 183–200
14. Hu S, Li M, Wang Q, Chow SSM, Du M (2018) Outsourced biometric identification with privacy. IEEE Trans Inform Forensics Sec 13(10):2448–2463
15. Huang Y, Malka L, Evans D, Katz J (2011) Efficient privacy-preserving biometric identification. In: Proceedings of the Network and Distributed System Security Symposium, NDSS 2011, San Diego, California, USA, 6th February - 9th February 2011. The Internet Society
16. Kikuchi H, Nagai K, Ogata W, Nishigaki M. (2010) Privacy-preserving similarity evaluation and application to remote biometrics authentication. Soft Comput 14(5):529–536
17. Klontz JC, Jain AK (2013) A case study of automated face recognition: The boston marathon bombings suspects. IEEE Computer 46(11):91–94
18. Kelion L Gatwick airport commits to facial recognition tech at boarding. [EB/OL]. https://www.bbc.com/news/technology-49728301
19. Mahdikhani H, Lu R, Zheng Y, Shao J, Ghorbani A (2020) Achieving o(log3n) communication-efficient privacy-preserving range query in fog-based iot. IEEE Internet Things J 7(6):5220–5232
20. Mahdikhani H, Shahsavarifar R, Lu R, Bremner D (2020) Achieve privacy-preserving simplicial depth query over collaborative cloud servers. Peer-to-Peer Netw Appl 13(1):412–423
21. Mandal A, Roy A, Yasuda M (2015) Comprehensive and improved secure biometric system using homomorphic encryption. In: García-Alfaro J, Navarro-Arribas G, Aldini A, Martinelli F, Suri N (eds) Data Privacy Management, and Security Assurance - 10th International Workshop, DPM 2015, and 4th International Workshop, QASA 2015, Vienna, Austria, September 21-22, 2015. Revised Selected Papers, *Lecture Notes in Computer Science*, vol 9481. Springer, pp 183–198
22. Nguyen N-T, Chang C-C (2018) Untraceable biometric-based three-party authenticated key exchange for dynamic systems. Peer-to-Peer Netw Appl 11(3):644–663
23. Patsakis C, van Rest J, Choras M, Bouroche M (2015) Data Privacy Management, and Security Assurance - 10th International Workshop, DPM 2015, and 4th International Workshop, QASA 2015, Vienna, Austria, September 21-22, 2015. Revised Selected Papers, *Lecture Notes in Computer Science*. In: García-Alfaro J, Navarro-Arribas G, Aldini A, Martinelli F, Suri N (eds), vol 9481. Springer, pp 169–182
24. Schroff F, Kalenichenko D, Philbin J (2015) Facenet: A unified embedding for face recognition and clustering. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, pp 815–823
25. Toli C, Preneel B (2018) Privacy-preserving biometric authentication model for e-finance applications. In: Mori P, Furnell S, Camp O (eds) Proceedings of the 4th International Conference on Information Systems Security and Privacy, ICISSP 2018, Funchal, Madeira - Portugal, January 22-24, 2018. SciTePress, pp 353–360
26. Wang Q, Hu S, Ren K, He M, Du M, Wang Z (2015) Cloudbi: Practical privacy-preserving outsourcing of biometric identification in the cloud. In: Pernul G, Ryan PYA, Weippl ER (eds) Computer Security - ESORICS 2015 - 20th European Symposium on Research in Computer Security, Vienna, Austria, September 21-25, 2015, Proceedings, Part II, *Lecture Notes in Computer Science*, vol 9327. Springer, pp 186–205

27. Wang Y, Wan J, Guo J, Cheung Y, Yuen PC (2018) Inference-based similarity search in randomized montgomery domains for privacy-preserving biometric identification. IEEE Trans Pattern Anal Mach Intell 40(7):1611–1624
28. Yang X, Zhu H, Lu R, Liu X, Li H (2018) Efficient and privacy-preserving online face recognition over encrypted outsourced data. In: IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), iThings/GreenCom/CPSCom/SmartData 2018, Halifax, NS, Canada, July 30 - August 3, 2018, pp 366–373. IEEE
29. Yuan J, Yu S (2013) Efficient privacy-preserving biometric identification in cloud computing. In: Proceedings of the IEEE INFOCOM 2013, Turin, Italy, April 14-19, 2013, pp 2652–2660. IEEE
30. Yu S, Park K, Park Y, Kim H, YoungHo P (2020) A lightweight three-factor authentication protocol for digital rights management system. Peer-to-Peer Netw Appl 13(5):1340–1356
31. Zheng Y, Lu R, Beibei L, Shao J, Yang H, Choo K-KR (2019) Efficient privacy-preserving data merging and skyline computation over multi-source encrypted data. Inf. Sci. 498:91–105
32. Zezula P, Amato G, Dohnal V, Batko M (2006) Similarity search - the metric space approach. Adv Database Syst 32, Kluwer
33. Zhang C, Zhu L, Xu C (2017) PTBI: an efficient privacy-preserving biometric identification based on perturbed term in the cloud. Inf Sci 409:56–67
34. Zhu H, Wei Q, Yang X, Lu R, Li H (2018) Efficient and privacy-preserving online fingerprint authentication scheme over outsourced data
35. Zhu Y, Li X, Wang J, Li J (2020) Cloud-assisted secure biometric identification with sub-linear search efficiency. Soft Comput 24(8):5885–5896
36. Zhu Y, Takagi T, Hu R (2014) Security analysis of collusion-resistant nearest neighbor query scheme on encrypted cloud data. IEICE Trans Inf Syst 97-D(2):326–330

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Xiaopeng Yang received the B.Sc. from Xidian University in 2014, M.Sc from Xidian University in 2017. He is current working toward his ph.D. degree with the school of Cyber Engineering, Xidian University, China. His interests are in the areas of applied cryptography, data security and privacy.

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.

Fengwei Wang received the B.Sc. degree from Xidian University, Xi'an, China, in 2016. He is current working toward the ph.D. degree with the School of Cyber Engineering, Xidian University, Xi'an, China. His research interests include the areas of applied cryptography, cyber security, and privacy.

Songnian Zhang received his M.S. degree from Xidian University, China, in 2016 and he is currently pursuing his Ph.D. degree in the Faculty of Computer Science, University of New Brunswick, Canada. His research interest includes cloud computing security, big data query and query privacy.

Rongxing Lu(S'09-M'11-SM'15) is currently an associate professor at the Faculty of Computer Science (FCS), University of New Brunswick (UNB), Canada. 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 Electrical & Computer Engineering, University of Waterloo, Canada, in 2012; and won the 8th IEEE Communications Society (ComSoc) Asia Pacic (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. He has published extensively in his areas of expertise, and was the recipient of 9 best (student) paper awards from some reputable journals and conferences. Currently, Dr. Lu currently serves as the Vice-Chair (Conferences) of IEEE ComSoc CIS-TC (Communications and Information Security Technical Committee). Dr. Lu is the Winner of 2016-17 Excellence in Teaching Award, FCS, UNB.

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.

Affiliations

Xiaopeng Yang[1,2] · **Hui Zhu**[1] · **Fengwei Wang**[1,2] · **Songnian Zhang**[2] · **Rongxing Lu**[2] · **Hui Li**[1]

Xiaopeng Yang
xiaopengyang2015@gmail.com

Fengwei Wang
xdwangfengwei@gmail.com

Songnian Zhang
szhang17@unb.ca

Rongxing Lu
rlu1@unb.ca

Hui Li
lihui@mail.xidian.edu.cn

[1] State Key Laboratory of Integrated Services Networks,
Xidian University, Xi'an, 710126, China

[2] Faculty of Computer Science, University of New Brunswick,
New Brunswick, E3B 5A3 Canada