1-2019

Person re-identification over encrypted outsourced surveillance videos

Hang CHENG

Huaxiong WANG

Ximeng LIU
Singapore Management University, xmliu@smu.edu.sg

Yan FANG

Meiqing WANG

See next page for additional authors

Published in IEEE Transactions on Dependable and Secure Computing, 2019
DOI 10.1109/TDSC.2019.2923653

Person Re-Identification over Encrypted Outsourced Surveillance Videos

Hang Cheng, Huaxiong Wang, Ximeng Liu, Yan Fang, Meiqing Wang, and Xiaojun Zhang

Abstract—Person re-identification (Re-ID) has attracted extensive attention due to its potential to identify a person of interest from different surveillance videos. With the increasing amount of the surveillance videos, high computation and storage costs have posed a great challenge for the resource-constrained users. In recent years, the cloud storage services have made a large volume of video data outsourcing become possible. However, person Re-ID over outsourced surveillance videos could lead to a security threat, i.e., the privacy leakage of the innocent person in these videos. Therefore, we propose an ef**F**icient priv**A**cy-prese**R**ving pe**R**son Re-**I**D **S**cheme (FARRIS) over outsourced surveillance videos, which can ensure the privacy of the detected person while providing the person Re-ID service. Specifically, FARRIS exploits the convolutional neural network (CNN) and kernels based supervised hashing (KSH) to extract the efficient person Re-ID feature. Then, we design a secret sharing based Hamming distance computation protocol to allow cloud servers to calculate similarities among obfuscated feature indexes. Furthermore, a dual Merkle hash trees based verification is proposed, which permits users to validate the correctness of the matching results. The extensive experimental results and security analysis demonstrate that FARRIS can work efficiently, without compromising the privacy of the involved person.

Index Terms—Privacy-Preserving, person re-identification, secret sharing, secure Hamming distance, Merkle hash tree.

1 Introduction

With the increasing popularity of smart city and digital home, surveillance cameras are significantly adopted in our daily life, which are often installed across highway, supermarket, university campus, and so on. Especially, the continuously improving infrastructures and increasing security issues caused by crime activities and terrorist attacks significantly facilitate the growth of surveillance camera market. In 2018, it was reported from BBC News that the over 170 million surveillance cameras have been deployed in China, and approximately 400 million more cameras are expected to be installed within the next three years [1]. As a world famous market research store, Research and Markets forecasted that the video surveillance market will grow at a compound average growth rate of 11.8%, which is expected to touch US$ 43.8 billion by 2025 from US$ 18.3 billion in 2017 [2].

At present, the video data from the surveillance cameras are used in a wide range of applications such as traffic monitoring, crime forensics, and activity detection. As an essential processing task over visual data, person re-identification (Re-ID) has attracted considerable research interests in recent years. The purpose of person Re-ID over video data is to match a person of interest from different cameras that are equipped at different locations [4] (as Fig. 1). It means that the target person in one camera can be identified whether he/she has appeared in other cameras at different time and places. This speciality of person Re-ID is widely employed in many security-related fields, especially public security. For example, it can readily identify the presence or absence of a suspect in different surveillance videos. This information assists the police to infer the next step from the suspect. With the large increasing number of cameras, the extensive video data are created to introduce enormous storage and computational costs. It was estimated that video data from the global surveillance cameras are over 560 petabytes per day [5], which is an extremely heavy amount for any resource-constrained individual or enterprise to store and process.

Fig. 1. The illustration of the person re-identification problem. **Left**: A query person image. **Right**: Different persons with different postures are taken by the two cameras in various lighting and viewpoints. **Notes**: All images are from the VIPeR [3] dataset.

- *H. Cheng is with the College of Mathematics and Computer Science, Fuzhou University, Fuzhou 350108, China; State Key Laboratory of Cryptology, P.O.Box 5159, Beijing 100878, China; the School of Physical and Mathematical Sciences, Nanyang Technological University, Singapore. E-mail: hcheng@fzu.edu.cn.*
- *H. Wang is with the School of Physical and Mathematical Sciences, Nanyang Technological University, Singapore. Email: hxwang@ntu.edu.sg.*
- *X. Liu and M. Wang are with the College of Mathematics and Computer Science, Fuzhou University, Fuzhou 350108, China. X. Liu is also with Key Lab of Information Security of Network Systems (Fuzhou University), Fujian Province, China. E-mail: snbnix@gmail.com, mqwang@fzu.edu.cn.*
- *Y. Fang is with the College of Computer and Information Sciences, Fujian Agriculture and Forestry University, Fuzhou 350002, China. Email: yfang@fafu.edu.cn.*
- *X. Zhang is with the School of Computer Science, Southwest Petroleum University, Chengdu 610500, China. Email: zhangxjdzkd2012@163.com.*

The massive storage and powerful computation of the cloud server render large-scale data processing possible at lower costs[6]. However, the problem is that users know nothing about where the data is stored and whether it is deliberately tampered due to losing the physical control of the outsourced data. In this case, users could only trust the cloud server completely. It is obviously not a wise option for users, since the data leakage events are frequently reported. Thus, it remains a challenging task to protect data security and privacy in the cloud [7]. Directly employing traditional encryption algorithms (e.g., AES or RSA) prior to outsourcing operation could be a good choice for users to ensure the confidentiality of the data. Nevertheless, it also hinders the further processing of these encrypted data, such as person Re-ID over encrypted surveillance videos. A naive solution for secure person Re-ID is to download all encrypted video data, and decrypt them to perform person Re-ID locally. Nevertheless, it incurs high computational and communication costs to users. Hence, how to perform efficient person Re-ID over encrypted outsourced surveillance videos without revealing the confidentiality of the video data, is significantly demanded.

In the plaintext domain, most person Re-ID research focus on person retrieval, namely, identifying correctly the same person across several non-overlapped surveillance videos [8]. At this point, person Re-ID can be largely perceived as the problem of image-to-video retrieval. The reason is, in both cases, the inputs are images and the outputs are the best matching videos associated with the input image. As discussed in [9], video retrieval using image query is an asymmetric problem, where videos contain spatiotemporal information, whereas only spatial data for query image. A direct approach to this problem is to transform it into the sophisticated image retrieval issue [10] by viewing each frame of video as an independent image. But, it is prone to inefficiency especially in a large video dataset because a video contains lots of frames. A more efficient alternative technique, i.e., the temporal aggregation, is widely adopted in the existing image-to-video schemes [11], [12], [13]. But, the retrieval accuracy of these schemes is limited by the hand-crafted feature representation, such as SIFT [14], and Fisher Vector [15]. It is difficult for the hand-crafted features to largely improve the retrieval performance due to the lack of the optimal compatibleness with the spatiotemporal information. To obtain better accuracy, the deep learning technique was introduced to extract more effective features in the schemes [9], [16]. However, the privacy of video data and query images are not guaranteed in these schemes [9], [11], [12], [13], [16].

To the best of our knowledge, few image-to-video research in the encrypted domain has been done. A rough framework about the privacy-preserving video retrieval has been proposed in [17]. Transferring into the privacy-preserving image retrieval is the core idea of this framework. However, it is not suitable for person Re-ID. This is because person Re-ID mainly targets identifying the existence of a given person over the different surveillance videos, whereas image retrieval focuses on returning the similar images with the query image. Therefore, the approaches of privacy-preserving image retrieval, such as [18], [19], [20], [21], [22], [23], cannot be applied directly to person Re-ID. Currently, there still lacks a feasible approach, which can support privacy-preserving person Re-ID over outsourced surveillance videos. Besides, another problem is that the cloud server may return a fraction of false retrieval results for its malicious purposes, i.e., saving storage costs or concealing the data corruption/loss accidents. To address this problem, many verifiable secure searchable encryption (SE) schemes [24], [25], [26], [27] have been developed to achieve keyword search over encrypted text files. However, the retrieval over encrypted outsourced videos or images does not take into consideration the correctness verification of the retrieval results.

In this paper, we devise an ef**F**icient priv**A**cy-prese**R**ving pe**R**son Re-**I**D **S**cheme (FARRIS) over outsourced surveillance videos, which allows the cloud server to perform the person Re-ID without knowing the plaintext content of the involved data, such as video data, and query information. To obtain better retrieval performance, we employ deep Convolutional Neural Network (CNN) to capture more effective features from key frames of videos. Also, the kernels based supervised hashing (KSH) is further exploited to accelerate person Re-ID. Moreover, we develop a verification mechanism to guarantee the exact matching results. The main contributions of our FARRIS are summarized as follows:

- ***Support secure person Re-ID***. To the best of our knowledge, this work is the first endeavor to develop privacy-preserving person Re-ID over outsourced surveillance videos. Our FARRIS allows users to store their data to the cloud server for secure Re-ID.
- ***Secure Hamming distance computation***. We construct a novel secure Hamming distance protocol. It allows the cloud server to compute the Hamming distance, without learning anything about the plaintext contents.
- ***Keyless feature index encryption***. Key Generation Center (KGC) is indispensable in general cryptosystem, who is responsible to manage and distribute users' private keys. However, over-reliance on KGC easily introduces the key escrow issue. Our scheme is designed to allow users to encrypt the feature indexes in keyless way.
- ***High accuracy and low costs***. An offline trained CNN based features are employed to capture invariant person characteristics for better matching performance. And also, KSH technique is exploited to transfer high-dimensional CNN features into short binary codes. It reduces storage and communication costs for users.
- ***Verifiable Re-ID***. Our FARRIS develops a dual Merkle hash trees to allow users to check the correctness of the matching results. Using the dual trees design, the users can identify whether the results have been tampered by the server or by the independent adversary.
- ***Privacy and efficiency***. Security analysis shows that our FARRIS can achieve the confidentiality of the plaintext surveillance videos and the query information. The extensive experiments demonstrate that our FARRIS is efficient and feasible.

The rest of the paper is arranged as follows. Section 2 provides some preliminaries. In section 3, we introduce the problem formulations including system model, problem statement, threat model, and design goals. The detail of the proposed FARRIS is shown in section 4. In section 5, the analysis of the correctness, security and performance is given. Section 6 reviews some related work. Conclusions are drawn in section 7.

2 Preliminaries

In this section, we review some feature-related techniques and cryptographic primitives, which serve as the basis of our FARRIS. The details are described as follows.

2.1 Image Feature Representation Based on CNN

Convolutional neural network (CNN) is a class of deep, feedforward artificial neural networks. Due to the great success of Krizhevsky et al. [28] in ILSVRC'12[1], CNN-based deep learning models have become increasingly common in the vision community, and have been used widely in image classification and image detection. In the literature of [28], [29], the deep CNN demonstrates the powerful capability of learning rich feature representation that is higher discriminative than traditional hand-engineered methods. Because of the low quality and high variety of person images, it is hard to succeed in performing person Re-ID without effective feature representation. In 2014, both Yi et al. [30] and Li et al. [31] first introduced a siamese neural network to determine whether a pair of person images observed by different cameras belongs to the same person of interest or not. Utilizing the powerful feature representation capability of CNN, these two methods obtain positive results in person Re-ID. Since then, deep CNN based person Re-ID has become popular.

2.2 Kernels Based Supervised Hashing (KSH)

In recent years, hashing has been widely used in large-scale vision applications. With the help of hashing techniques, high-dimensional data vectors can be encoded into short hash codes so as to reduce the storage costs and improve computation efficiency. Some classic hashing methods, such as product quantization (PQ) [32], iterative quantization (ITQ) [33], and KSH [34] have currently been proposed. The former two methods are unsupervised hashing and are hard to achieve better accuracy, especially in large-scale database. KSH is a kernel-based supervised hashing technique, where some techniques related to the kernel are already well studied in [35]. KSH employs the labeled data to learn compact hash codes that preserve the similarity between original data. The main idea of KSH is to minimize similar pairs' distances while maximizing the distances for dissimilar pairs in the Hamming space. As presented in [34], KSH's performance outperforms that of some state-of-the-art methods including LSH [36], PCAH [37], and LDAH [38].

1. namely, ImageNet Large Scale Visual Recognition Challenge 2012.

2.3 Secret Sharing

In this paper, we employ the secret sharing [39] mechanism to construct the privacy-preserving Hamming distance computation protocol. The main thought of secret sharing is to split any data (called *secret*) into n meaningless data (called *shares*), each of which does not reveal any information of the original secret data, but collecting the specific k shares can reconstruct the original secret data, known well as (k, n) threshold secret sharing. In [39], Shamir gives a practical solution to achieve (k, n) threshold secret sharing by using interpolation of polynomials over a finite field. In this paper, we adopt the Chinese Remainder Theorem (CRT) [40] for the convenience of integer operation. More specifically, let the secret be α, and denote its k distinct shares as $\{\alpha_1, \alpha_2, \cdots, \alpha_k\}$. Assume that each share α_i satisfies the following requirement,

$$\alpha_i = \alpha \bmod m_i, \tag{1}$$

where the set of positive integers $\{m_i\}_{(1\leq i\leq k)}$ is pairwise coprime, viz., $\gcd(m_i, m_j) = 1$ $(i \neq j$, and $1 \leq i, j \leq k)$. Based on the standard CRT, the secret α $(0 \leq \alpha < M)$ can be reconstructed by computing

$$\alpha = \left(\sum_{i=1}^{k} \alpha_i c_i\right) (\mathrm{mod}\ M), \tag{2}$$

where $M = \prod_{i=1}^{k} m_i$, and $c_i = M_i t_i$ $(M_i = \dfrac{M}{m_i}$, and $t_i M_i \equiv 1 \bmod m_i$, for all $1 \leq i \leq k)$. Certainly, if only k-1 distinct shares are available and $\alpha > m_{i_1} m_{i_2} \cdots m_{i_{k-1}}$, one does not exactly reconstruct the secret α.

2.4 Merkle Hash Tree (MHT)

An MHT is a complete binary tree [41], which can be often used to verify the integrity and validity of data so as to protect the data from tampering and forging, even deletion. In MHT, there are two types of tree nodes, leaf node and non-leaf node. For the former, the hash value is generated by hashing its corresponding data value. In addition, the hash value of the latter can be obtained through hashing the hash value of its all next child nodes. According to this working principle, any changes of the data value will affect the hash value of the root node.

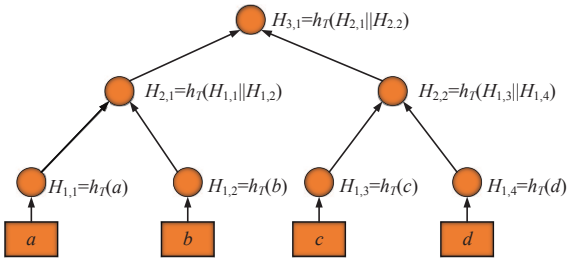


Fig. 2. The Merkle hash tree construction.

Here, we give a toy instance to further show the construction of MHT, shown in Fig. 2. Assume that a, b, c, d are four verified data values, where $a < b < c < d$. $h_T(\cdot)$ is a general collision-resistant hash function. "$\|$" indicates the concatenation of the hash values of two nodes. First, we

compute the hash values of four data: $H_{1,1} = h_T(a), H_{1,2} = h_T(b), H_{1,3} = h_T(c), H_{1,4} = h_T(d)$. These hash values are associated with four leaf nodes, respectively. Next, the hash values of the two non-leaf nodes from the intermediate level are obtained separately by hashing the hash values of their child nodes. As illustrated in Fig. 2, we get $H_{2,1} = h_T(H_{1,1}\|H_{1,2}), H_{2,2} = h_T(H_{1,3}\|H_{1,4})$. At last, we compute the root node's hash value $H_{3,1} = h_T(H_{2,1}\|H_{2,2})$. Under the assumption of a collision-resistant hash function with $h_T(\cdot)$, the adversaries cannot forge the data because of the uniqueness of $H_{3,1}$.

3 Problem Formulation

In this section, we present the system model, problem statement, threat model, and design goals, respectively.

3.1 System Model

In this paper, we propose a secure person Re-ID scheme in the cloud, as depicted in Fig. 3. It mainly contains four parties: Content Owner (**CO**), Cloud Storage Server (**CSS**), Cloud Data Server (**CDS**), and Authorized User (**AU**).

- The CO first extracts feature vectors from the plaintext surveillance videos, constructs corresponding feature indexes. Then, the CO employs the CRT-based secret sharing method to send the shares of the indexes to CDSs. Meanwhile, all encrypted surveillance videos and their corresponding identities are outsourced to CSS.
- The CSS provides the storage service to the COs, and returns the surveillance videos, which contain key frames similar to the query person image. Besides, CSS is responsible for responding to the AUs' challenges.
- The CDS has some data storage space to store shares submitted by users. Furthermore, CDS provides computational powder to perform modulus addition operation over feature indexes' shares, where the computational results are sent to the CSS.
- The AU authorized by CO splits the query feature index into shares, and randomly sends them to a certain number of CDSs, where no key is involved. Once obtaining the returned results, AUs can decrypt them privately with the help of the key. Here, we assume that the encryption key is shared through a secure channel. Also, AUs can submit the challenging information to CSS for verifying the correctness of the returned data. Note, CO can also be viewed as a specific AU.

3.2 Problem Statement

Considering that many COs have a large amount of surveillance videos, but they possess the limited computation and storage resources. To achieve privacy protection, video data should be encrypted before being outsourced into CSS. Given an image of a person of interest, AU can obtain the best matching encrypted surveillance videos from CSS, and also has an ability to verify the correctness of the returned Re-ID results. In this case, we need to overcome the following challenges because that all outsourced surveillance videos and query information are encrypted during the matching process.

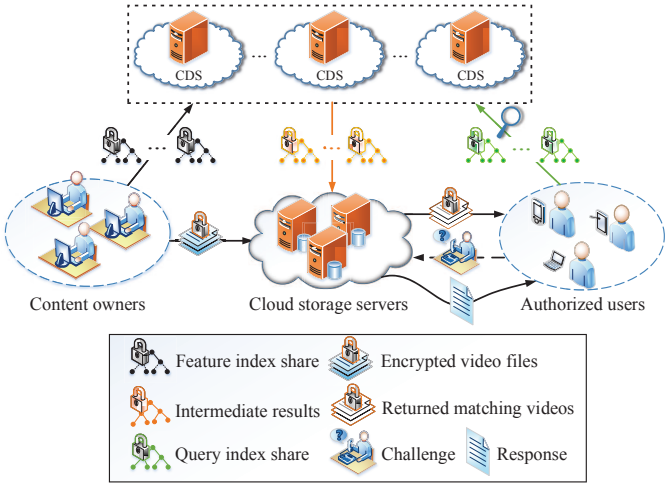


Fig. 3. The infrastructure of Re-ID.

- Since it takes more time to identify the existence of some person over large-scaled video data, an efficient feature representation in binary form is necessary to improve the matching speed. In addition, a secure Hamming distance computation protocol is demanding, without compromising the involved persons' privacy during person Re-ID.
- To reduce the costs of the key management, a keyless mechanism could be constructed to support the transformation of the uploaded feature vectors into the meaningless contents.
- In order to guarantee the correctness of the returned results, a verification mechanism is required to resist the tampering behavior from the CSS or the adversary.

3.3 Threat Model

In our scheme, CDS is considered as an *honest-but-curious* party, which is honest to carry out the pre-defined protocols, but curious with the private data related to COs and AUs, e.g., feature index shares and intermediate computational results. In addition, the CSS is assumed to be *semi-honest-but-curious*, which is also applied in [24], [25], [26]. Different from *honest-but-curious* model, this model not only takes into account the fact that CSS may return a false fraction of the person Re-ID results to AUs for the sake of concealing data loss accidents or its commercial interests , but also assumes that the partial malicious CDSs are allowed to collude with each other to share their information, in which the number of the involved CDSs mainly relies on k in the (k, n) secret sharing scheme. Only if the number of the involved CDSs is less than k, can our scheme keep the original feature vectors confidential to CDSs. It should be stressed that CDSs cannot collude with CSS in our scheme. Based on the information available to CDSs and CSS, we consider two threat models here, which are also used in these schemes [21], [22], [42].

Known Ciphertext Model. The CDS only learns about the shares sent by the COs and AUs. Meanwhile, the encrypted outsourced surveillance videos and the processed results obtained from CDSs are available to CSS.

Known Background Model. In this stronger model, the CDS and CSS can obtain more information than that in the known ciphertext model. They can leverage the statistical information to infer specific contents in a query image. The CDS and CSS can even gain some person images, but know nothing about the relations between their plaintext feature vectors and corresponding encrypted versions.

3.4 Design Goals

In order to address the above two threat models, we employ CRT-based secret sharing technique to develop a secure person Re-ID scheme, which achieves privacy-preserving query person matching over encrypted surveillance videos. In this case, our FARRIS should achieve the following goals:

- *Data Privacy*. FARRIS should guarantee the feature indexes and the outsourced video data not to be leaked. In addition, the query privacy should be protected during person Re-ID.
- *Query Unlinkability*. Considering that the queries are available to the CDSs during the person Re-ID, the queries should be unlinkable for privacy protection.
- *Secure Multi-user Support*. Due to the variation in the amount of users, the scalability and extensibility of the system should be preserved in FARRIS. Furthermore, users should not learn about the private information of feature indexes from each other.
- *Results Verification*. FARRIS should have a verification mechanism to allow AUs to check the correctness of the results returned by CSS who is assumed to follow the *semi-honest-but-curious* model.

4 Privacy-preserving person Re-ID scheme

In the plaintext domain, person Re-ID reveals the privacy of persons of interest in surveillance videos. Also, with an increasing amount of the videos, individuals and enterprises with limited resources suffer from high computation overheads and great storage costs. To solve these issues, we propose a novel privacy-preserving person Re-ID scheme over encrypted outsourced surveillance videos. More details are presented in this Section.

4.1 Notations

To facilitate understanding of FARRIS's concrete construction, we first define some notations, as shown in TABLE 1.

4.2 Proposed FARRIS

Prior to the construction of FARRIS, we first introduce how to extract the feature vectors from the surveillance videos. First, we extract the person images from the surveillance videos by using the existing person detection methods in [43]. Note that key frames are considered as candidates for the person images. The key frame, namely, I-frame, is a single video frame, which determines the beginning or end of a transition. Due to the information integrity of the key

TABLE 1
Notation descriptions in FARRIS scheme.

Notations	Descriptions
K	MAC[1] key of CO
$h_T(\cdot)$	Hash function for Merkle hash tree
(k, n)	Threhold for secret sharing
$M = \{m_1, \cdots, m_n\}$	Prime set for secret sharing
$\mathcal{D} = \{d_1, \cdots, d_n\}$	CDS set
$\mathcal{V} = \{v_1, \cdots, v_m\}$	Video file set
$\mathcal{C} = \{c_1, \cdots, c_m\}$	Ciphertext set for $\mathcal{V}$
$ID = \{id_1, \cdots, id_m\}$	Identity set of $\mathcal{V}$
$\mathcal{M} = \{\mathcal{M}_1, \cdots, \mathcal{M}_m\}$	MAC set for $\mathcal{C}$
$\mathcal{P} = \{p_1, \cdots, p_\ell\}$	Person image set
$\mathcal{F}_i = \{f_{i1}, \cdots, f_{id}\}_{1 \le i \le \ell}$	Feature vector for image i
I_i	Index for p_i
$I_{i,j}$	I_i's share for CDS d_j
$Q = \{q_1, \cdots, q_d\}$	Feature vector for query Q
T_Q	Index for Q
$T_{Q,j}$	Q's share for CDS d_j

[1] Message Authentication Code.

frame, it is often taken as the main reference by P-frame and B-frame to compress the video data. In this paper, the key frame mainly represents the I-frame which contains persons only. The rationale of this decision is to reduce temporal redundancy to improve the matching efficiency. Then, we employ the AlexNet CNN based HIPHOP feature [4] to represent person Re-ID features. Moreover, we introduce KSH technique to convert high-dimensional HIPHOP features into short binary hash codes, which can be considered as feature vectors in our FARRIS.

Now we present a **basic FARRIS**, as shown in Fig. 4, where we will show how to actually construct an efficient Re-ID over encrypted cloud video data. The **basic FARRIS** contains a tuple of five algorithms as follows.

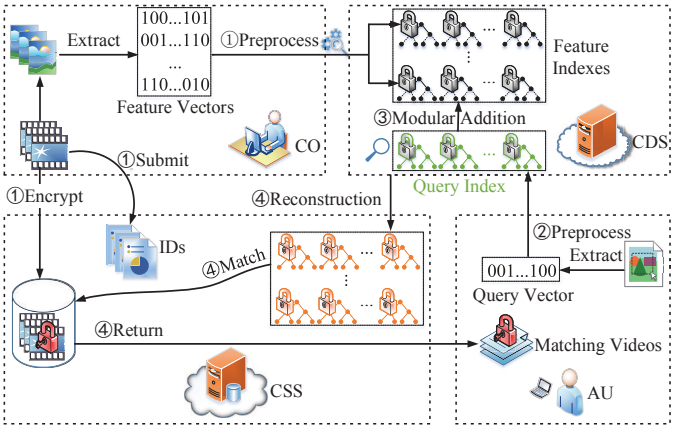


Fig. 4. Framework of **basic FARRIS** scheme.

KeyGen: In general, secure text/image retrieval schemes, such as [21], [22], [25], [26], [42], without concerning about the encryption details of the outsourced multimedia data, they focus on the feature encryption and how to match among encrypted features. Likewise, as for the outsourced surveillance videos, we encrypt them directly by using the traditional symmetric encryption (e.g., AES). Furthermore, different from the aforementioned retrieval schemes, the encryption for the indexes either from COs or AUs is keyless in FARRIS, whereas the key is necessary for the former. The

key-free characteristic will largely reduce the costs of the key management and storage. The fact behind this advantage is that we adopt the secret sharing technique to process the feature indexes. It can allow users to split their data into multiple obfuscated shares, where no encryption keys are involved, and the original data can be reconstructed as long as one aggregates a certain number of shares.

IndexBuild: As shown in the step ①, CO encrypts all video files $\mathcal{V} = \{v_1, \cdots, v_m\}$ into the ciphertext set $\mathcal{C} = \{c_1, \cdots, c_m\}$ one by one. Then, the feature vectors $\mathcal{F}_i = \{f_{i1}, \cdots, f_{id}\}_{1 \le i \le \ell}$ are extracted from person images, which are key frames obtained from the plaintext videos $\mathcal{V}$. Next, we carry out a preprocessing operation to generate feature indexes to avoid information leakage of feature vectors to CDSs and CSS. After that, CO splits each feature index into n shares by using the CRT method. Algorithm 1 provides the generation of feature indexes' shares.

Algorithm 1: Generation of feature indexes' shares

Input: Person image set $\mathcal{P} = \{p_1, \cdots, p_\ell\}$, prime set $M = \{m_1, \cdots, m_n\}$
Output: Feature index's shares $I_{i,j}$ ($i \in [1, \ell]$, $j \in [1, n]$)

Build the feature vector set $\mathcal{F} = \{\mathcal{F}_1, \cdots, \mathcal{F}_\ell\}$;
for $i = 1$ *to* ℓ **do**
  Padding zeros $\rightarrow \mathcal{R}_i$;
  Random permutation $\rightarrow \mathcal{X}_i$;
  Substitution in odd/even number manner $\rightarrow \mathcal{Y}_i$;
  Scale transformation $\rightarrow \mathcal{Z}_i$;
  Set feature index $I_i = \mathcal{Z}_i$;
  while *not at the end of set* M **do**
    Compute share $I_{i,j} = I_i \ mod \ m_j$;
  end
end
return All shares of all feature indexes.

More specifically, some modifications should be done for feature vectors in advance for achieving secure Hamming distance computation. The details are presented as follows:

- CO pads z zeros into $\mathcal{F}_i$. Thus, the dimension of $\mathcal{F}_i$ would be extended to $(d + z)$, denoted as
$$\mathcal{R}_i = \{f_{i1}, \cdots, f_{id}, \underbrace{0, \cdots, 0}_{z}\}.$$
- After padding operation, CO randomly permutates the positions of all elements of any feature vector $\mathcal{R}_i$. The permutation operation converts $\mathcal{R}_i$ into
$$\mathcal{X}_i = \{f_{i_{g1}}, \cdots, f_{i_{g(d+z)}}\}.$$
- When getting $\mathcal{X}_i$, CO randomly chooses a positive odd number to replace 1, and a positive even number instead of 0. Here, the range of selected odd or even is set $[1, \Gamma]$. Note that $1/0$ at different positions of $\mathcal{X}_i$ may correspond to different odd/even number. According to the above replacement rule, CO modifies all feature vectors $\mathcal{X}_i$ as
$$\mathcal{Y}_i = \{f'_{i_{g1}}, \cdots, f'_{i_{g(d+z)}}\}.$$
- Following the third modification, CO continues to modify the values of the elements of $\mathcal{Y}_i$. Here, we leverage the scale technique [44] to further protect the information of $\mathcal{F}_i$ from being disclosed. The scale modification can be done using
$$u' = u \cdot s + \varepsilon,$$
where u is the value of an element in $\mathcal{Y}_i$, s is a scale factor as well as positive constant, and ε is a random noise whose value is uniformly distributed, namely $\varepsilon \sim \mathrm{U}(0, \gamma)$, $(\gamma \le s)$. Here, s is public and is allowed to vary for different elements in $\mathcal{Y}_i$, and ε is private to CO. Thus, the feature vector for each person image p_i is finally modified as
$$\mathcal{Z}_i = \{f''_{i_{g1}}, \cdots, f''_{i_{g(d+z)}}\}.$$
Then, CO takes $\mathcal{Z}_i$ as feature index I_i of person image p_i, namely $I_i = \mathcal{Z}_i$.

After the feature preprocessing with the above four modifications, CO splits the I_i into n shares by using
$$\pi(I_i) = \{I_{i,1}, I_{i,2}, \cdots, I_{i,n}\} \tag{3}$$
$$I_{i,j} = I_i \bmod m_j, \tag{4}$$
where $\pi(\cdot)$ is denoted as a splitting function, which is mainly based on the modulo prime as present in Eq. 1. The prime set $\{m_j\}_{1 \le j \le n}$ is available for CDSs and CSS.

In fact, all components of I_i execute separately π operation in FARRIS. Let the w-th component of I_i be $I_i(w)$, its corresponding shares are
$$\pi(I_i(w)) = \{I_{i,1}(w), I_{i,2}(w), \cdots, I_{i,n}(w)\} \tag{5}$$
and
$$I_{i,j}(w) = I_i(w) \bmod m_j. \tag{6}$$
Finally, CO uploads all encrypted video files $\mathcal{C}$ together with $ID = \{id_1, \cdots, id_m\}$ to the CSS. Besides, n shares of each I_i $(1 \le i \le \ell)$ are sent to the corresponding CDS, respectively. The mapping relation between these shares and CDS set $\mathcal{D}$ is represented below:
$$I_{i,j} \longleftrightarrow d_j, (1 \le i \le \ell, 1 \le j \le n, d_j \in \mathcal{D}).$$
QueryGen: As shown in the step ②, the query index T_Q would be generated by using the above four modifications before outsourcing. Similar to the splitting process of the CO's index, a query user AU splits T_Q into n shares, which meet the requirement below.
$$\pi(T_Q) = \{T_{Q,1}, T_{Q,2}, \cdots, T_{Q,n}\} \tag{7}$$
and
$$\pi(T_Q(w)) = \{T_{Q,1}(w), T_{Q,2}(w), \cdots, T_{Q,n}(w)\} \tag{8}$$
$$T_{Q,j}(w) = T_Q(w) \bmod m_j, \tag{9}$$
where $T_{Q,j}(w)$ is the j-th share of the w-th element of T_Q.

Then, AU randomly selects k out of shares $\{T_{Q,j}\}_{1 \le j \le n}$, and randomly sends to k out of n CDSs, respectively. In this case, CSS does not send the requests to k CDSs for reconstructing the secret from k shares because that k CDSs activated by AU would automatically send their intermediate results to CSS. It would avoid the interaction between CDSs and CSS, reducing the communication costs.

SumComp: As shown in the step ③, once obtaining the shares of the query T_Q from the query user AU, CDSs start to calculate the sum between query index share and any index share stored in CDSs. Given a query index T_Q and the index I_i of any person image $p_i (1 \leq i \leq \ell)$, the CDS d_j that is activated by AU carries out the addition operation under modulo prime m_j, denoted as Sum_j^{Q,p_i}, i.e.,

$$\begin{aligned} Sum_j^{Q,p_i} &= (T_{Q,j} + I_{i,j}) \bmod m_j \\ &= \{(T_{Q,j}(1) + I_{i,j}(1)) \bmod m_j, \\ &(T_{Q,j}(2) + I_{i,j}(2)) \bmod m_j, \\ &\cdots, \\ &(T_{Q,j}(g) + I_{i,j}(g)) \bmod m_j\}, \end{aligned} \tag{10}$$

where g denotes the dimension of feature index and is equal to $d + z$.

After computing Sum_j^{Q,p_i} for any i ($i \in [1, \ell]$), the CDS d_j submits all intermediate values $\{Sum_j^{Q,p_i}\}_{1 \leq i \leq \ell}$ to CSS. Since the reconstruction of the final sum Sum^{Q,p_i} is done by CSS in FARRIS, the Sum^{Q,p_i} is a secret to any CDS. Even if CDSs may collude with each other, this sum is not revealed if and only if the amount of involved CDSs is less than the threshold k according to the CRT method.

SimComp: As shown in the step ④, when getting k intermediate results Sum_j^{Q,p_i}, the CSS first employs the CRT method to reconstruct the original sum vector Sum^{Q,p_i} between T_Q and each I_i. Then, CSS scales down the sum vector Sum^{Q,p_i} by the scale factor s used in the feature preprocessing stage. Let the Sum^{Q,p_i} be $\{t_1, t_2, \cdots, t_g\}$, the scaled version $\widetilde{Sum}^{Q,p_i}$ can be computed by

$$\widetilde{Sum}^{Q,p_i} = \{\lfloor t_1/s \rfloor, \lfloor t_2/s \rfloor, \cdots, \lfloor t_g/s \rfloor\}.$$

Although the CSS has no idea of the original feature vectors Q and $\mathcal{F}_i$, it can still calculate the Hamming distance between them based on the $\widetilde{Sum}^{Q,p_i}$. In the following, we briefly introduce the calculation process.

As described in the **IndexBuild** algorithm, the $\mathcal{Y}_i$ is generated from $\mathcal{F}_i$ by the former three modifications during the feature vector preprocessing. Assume that Q' is the counterpart of $\mathcal{Y}_i$ for the query Q. Apparently, the $\widetilde{Sum}^{Q,p_i}$ is exactly the sum of Q' and $\mathcal{Y}_i$ only if the random noise $\varepsilon \leq s$. As we all know that the number of 1s in the sum of any two binary vectors with same dimension is equivalent to their Hamming distance. Furthermore, we also find that all but the odd number 1 are even in the two binary vectors' sum. It means that the Hamming distance can also be calculated by counting the number of the sum vector's elements with odd values. Although Q' and $\mathcal{Y}_i$ are dramatically different from the corresponding original binary feature vectors after the feature vector preprocessing, the quantity of odd numbers in the sum $\widetilde{Sum}^{Q,p_i}$ between them remains unchanged. The main reason is that the replacement rule (odd$\leftrightarrow$ 1, even$\leftrightarrow$ 0) in the **IndexBuild** algorithm does not change the parity of the elements of the original binary vectors' sum. Moreover, the quantity invariance of odd numbers is also not affected by the zero-padding process because that the sum of even numbers is still even. Based on the above analyses, CSS can obtain the Hamming distance between Q and $\mathcal{F}_i$ directly from their preprocessed feature vectors' sum $\widetilde{Sum}^{Q,p_i}$ by counting the number of its elements with odd values.

Finally, CSS sorts all Hamming distances, and then returns the related encrypted video set $\mathcal{C}' = \{c'_1, \cdots, c'_q\}$ as well as its corresponding identity set $ID' = \{id'_1, \cdots, id'_q\}$ in the ascend order.

Remark: Now we explain why we need to do the four preprocessing operations for feature vectors. The *padding zero* and *random permutation* operations aim at solving the following issue. If the dimensionality of $\mathcal{F}_i$ ($i \in [1, \ell]$) is small, the CSS can readily employ the exhaustive way to infer these two original binary vectors from their sum. The strategy of the *odd/even number substitution* aims to change the fact that the number 1 or 0 always remains unchanged in any secret share. Assume that $(1, 0, 0, 1)$ is modified as $(133, 14, 68, 77)$ via this step processing. Its share would be $(9, 14, 6, 15)$ under the prime 31, which would conceal the original binary vector $(1, 0, 0, 1)$. The *scale transformation* is to avoid a universal phenomenon, the parity of a positive number may stay the same with a certain probability under modulo prime. As the above example, the odd number $133/77 \rightarrow 9/15$. The even number draws a similar conclusion. Through this phenomenon, CDSs may infer the distribution of 1s and 0s in the original feature vector $\mathcal{F}_i$.

The above **basic FARRIS** can achieve privacy-preserving person Re-ID over encrypted outsourced surveillance videos under the assumption that the cloud server is *honest-but-curious* model. However, in real-world applications, the cloud server is more likely to follow *semi-honest-but-curious* model. It means that the cloud server can forge or tamper person Re-ID matching results. Since the surveillance videos are outsourced to CSS, we only assume that CSS is *semi-honest-but-curious*. To address this problem, we equip **basic FARRIS** with a verifiable mechanism to enhance it for resisting attacks emerged from the *semi-honest-but-curious* model. This new version is denoted as **enhanced FARRIS**. The critical parts of the **enhanced FARRIS** are presented as follows.

Firstly, we need to appropriately modify **BuildIndex** algorithm in **basic FARRIS**. That is because that the MAC technique is used in our verification mechanism. MAC is a cryptographic primitive, which is used to achieve both the integrity and authentication of a message. Due to the existential unforgeability of MAC, it is considered to be added into our **enhanced FARRIS**. Specifically, we employ the MAC method to generate an MAC value $\mathcal{M}_i = \mathbf{MAC}_K(c_i \| id_i)$ for any encrypted video file c_i associated with identity id_i ($i \in [1, m]$), and MAC key K, where $\mathbf{MAC}(\cdot)$ is a secure MAC scheme (e.g. HMAC[2]). Taking $\mathcal{M}_i$ as element, an MAC set $\mathcal{M}$ is finally produced as follows,

$$\begin{aligned} \mathcal{M} &= \{\mathcal{M}_1, \cdots, \mathcal{M}_m\} \\ &= \{\mathbf{MAC}_K(c_1 \| id_1), \cdots, \mathbf{MAC}_K(c_m \| id_m)\}. \end{aligned} \tag{11}$$

After that, MACs $\mathcal{M}$ along with all encrypted video files are outsourced to the CSS.

Secondly, we give the details of **verification** algorithm construction. To be specific, after receiving the Re-ID results $\mathcal{C}'$, AU first randomly selects a subset $\mathcal{C}'' = \{c'_{l_i}\}_{1 \leq i \leq t}$

2. A specific type of message authentication code based on hash function and encryption key.

from the set $\mathcal{C}'$, where $\{l_i\}_{1\leq i\leq t}$ denotes the positions of the selected encrypted videos in $\mathcal{C}'$, and $t \leq q$. Then, AU sends the challenging information $\{l_i\}_{1\leq i\leq t}$ to CSS. When obtaining the verification request from AU, CSS constructs an MHT (shown in Algorithm 2) based on the challenging information, where assume that the MAC set $\{\mathcal{M}'_{l_i}\}_{1\leq i\leq t}$ is the counterpart of $\{c'_{l_i}\}_{1\leq i\leq t}$ in the whole MAC set $\mathcal{M}$. In this MHT, the hash values of all leaf nodes are calculated by calling the hash function $h_T : \{0,1\}^* \rightarrow \{0,1\}^\kappa$, which takes as input the corresponding ciphertext video's MAC. For any intermediate node, CSS can obtain its hash value by hashing the concatenation of its two direct child nodes' hash values, where if only one child node exists, father node's hash value can be calculated just by hashing this child's hash value. In the similar way, CSS computes the hash value H_{root} of the root node. At last, CSS sends H_{root} to AU.

After receiving the H_{root}, AU first generates the MAC set $\{\mathcal{M}''_{l_i}\}_{1\leq i\leq t}$ for $\{c'_{l_i}\}_{1\leq i\leq t}$ with the MAC key K shared by the corresponding CO. Then, AU builds an MHT by the method as presented above. Finally, AU checks the correctness of $\mathcal{C}'$ by identifying whether its own generated root node's hash value H'_{root} is equal to the H_{root} from the CSS. A formal description of the verification mechanism is presented in Algorithm 3.

Algorithm 2: Merkle hash tree construction

Input: MACs $\{\mathcal{M}_{l_i}\}_{1\leq i\leq t}$
Output: A Merkle hash tree

$T \leftarrow \emptyset$;
Leaf_Node_set$\leftarrow \{h_T(\mathcal{M}_{l_i})\}_{1\leq i\leq t}$;
$Insert(T$, Leaf_Node_set);
New_Node_set$\leftarrow$ Leaf_Node_set;
while *the size of New_Node_set* ≥ 2 **do**
  Leaf_Node_set$\leftarrow$ New_Node_set;
  New_Node_set$\leftarrow \emptyset$;
  while *not at the end of Leaf_Node_set* **do**
    Left_Node$\leftarrow$ Leaf_Node_set(1);
    Right_Node$\leftarrow$ Leaf_Node_set(2);
    Inter_Node$\leftarrow h_T$(Left_Node||Right_Node);
    $Insert(T$, Inter_Node);
    New_Node_set$\leftarrow$ Inter_Node;
    Delete Leaf_Node_set(1), Leaf_Node_set(2) from Leaf_Node_set ;
  end
end
return T

Remark: As discussed above, AU can verify whether the Re-ID results $\mathcal{C}'$ are correct or not. However, the verification failure does not necessarily mean that the returned results must have been tampered by CSS. A third-party adversary may modify the results during the transmission. To avoid the ambiguity, we design an advanced verification mechanism, where a dual MHTs based construction is employed to check the correctness of Re-ID results. The new verification mechanism merits the unique determination of the true counterfeiter.

As shown in Fig.5, with the challenging information $\{l_i\}_{1\leq i\leq t}$, CSS builds another MHT CT for $\{c'_{l_i}\}_{1\leq i\leq t}$ as it builds the tree MT for MAC set $\{\mathcal{M}'_{l_i}\}_{1\leq i\leq t}$. Overall,

Algorithm 3: Verification mechanism for CSS returned results

Input: Returned results $\mathcal{C}'$, $\mathcal{M}'$, the corresponding identity set ID', MAC key K
Output: "True" or "False"

$\mathcal{C}' = \{c'_1, \cdots, c'_q\}$,$\mathcal{M}' = \{\mathcal{M}'_1, \cdots, \mathcal{M}'_q\}$, $ID' = \{id'_1, \cdots, id'_q\}$;
AU sends the challenging information $\{l_i\}_{1\leq i\leq t}$ to CSS;
Build a Merkle hash tree $MT \leftarrow MerkleTreeGen(\{\mathcal{M}'_{l_i}\}_{1\leq i\leq t})$;
Obtain the tree root's hash value H_{root};
CSS publishes the H_{root};
AU computes MAC codes for $\{\mathcal{M}''_{l_i} \leftarrow \mathbf{MAC}_K(c'_{l_i}||id'_{l_i})\}_{1\leq i\leq t}$;
Build a Merkle hash tree $MT' \leftarrow MerkleTreeGen(\{\mathcal{M}''_{l_i}\}_{1\leq i\leq t})$;
Obtain the tree root's hash value H'_{root};
Check $H_{root} \stackrel{?}{=} H'_{root}$;
If the above equation holds, output "True"; otherwise, output "False".

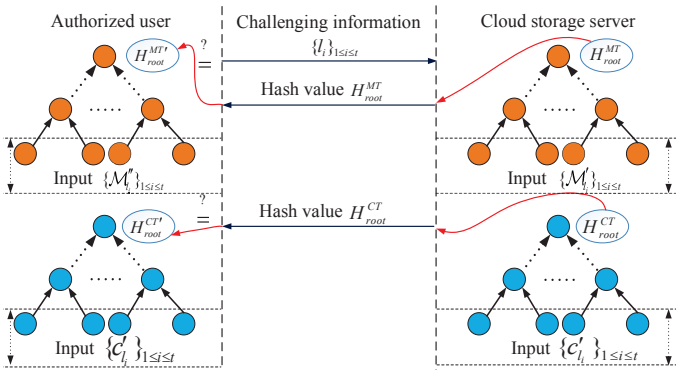


Fig. 5. The dual Merkle hash trees based verification mechanism.

these two trees are constructed in the same way. The only distinction is that the inputs to the hash function h_T are different at leaf nodes. The tree MT takes as input $\{\mathcal{M}'_{l_i}\}_{1\leq i\leq t}$, whereas $\{c'_{l_i}\}_{1\leq i\leq t}$ for CT. After the tree construction, CSS publishes their root nodes' hash values. Likewise, AU also needs to build the two trees CT' and MT' to echo the trees CT and MT, respectively. Assume that the hash values of the root nodes of CT' and MT' are represented as $H^{CT'}_{root}$ and $H^{MT'}_{root}$, respectively. Also, their corresponding hash values in CSS, denoted as H^{CT}_{root} and H^{MT}_{root}. If $H^{MT}_{root} = H^{MT'}_{root}$ holds, AU confirms the Re-ID results are correct; otherwise, AU justifies that the results are forged. If the verification fails, AU can further identify whether the true faker is CSS or the adversary based on the two following cases. If $H^{CT}_{root} \neq H^{CT'}_{root}$ happens, it shows that the results are tampered by the adversary. When $H^{CT}_{root} = H^{CT'}_{root}$ and $H^{MT}_{root} \neq H^{MT'}_{root}$ occur, AU can determine the results have been tampered by CSS.

5 Analysis of our FARRIS

In this section, we first give the correctness analysis of the proposed FARRIS, and then demonstrate its security and performance over encrypted outsourced surveillance videos.

5.1 Correctness Analysis

The exact Hamming distance computation depends on whether the reconstruction of the intermediate results from CDSs is done correctly or not. In our FARRIS scheme, the CSS can be guaranteed to correctly reconstruct the original sum vector Sum^{Q,p_i} of T_Q and I_i if and only if it obtains the k intermediate values Sum_j^{Q,p_i} from the k CDSs $d_j (j \in [1,n])$.

Theorem 1. Given a $j \in [1,n]$, the Sum_j^{Q,p_i} satisfies:

$$\begin{aligned} Sum_j^{Q,p_i} = \{&(T_Q(1) + I_i(1)) \bmod m_j, \\ &(T_Q(2) + I_i(2)) \bmod m_j, \\ &\cdots, \\ &(T_Q(g) + I_i(g)) \bmod m_j\}. \end{aligned} \tag{12}$$

Proof: Based on Eq. 6 and 9, we rewrite Eq. 10 as

$$\begin{aligned} Sum_j^{Q,p_i} &= (T_{Q,j} + I_{i,j}) \bmod m_j \\ &= \{(T_Q(1) \bmod m_j + I_i(1) \bmod m_j) \bmod m_j, \\ &(T_Q(2) \bmod m_j + I_i(2) \bmod m_j) \bmod m_j, \\ &\cdots, \\ &(T_Q(g) \bmod m_j + I_i(g) \bmod m_j) \bmod m_j\}. \end{aligned}$$

For the sake of presentation, we denote $T_Q(w)$, $I_i(w)$ as a and b, respectively, where w $(w \in [1,g])$. Based on the knowledge of Subsection 4.2, all components of the preprocessed feature vectors are integers, which means that both a and b are integers. According to the quotient remainder theorem, also known as the division algorithm, there exist unique integers k_a/k_b and r_a/r_b such that $a = k_a \cdot m_j + r_a$, and $b = k_b \cdot m_j + r_b$, where $0 \le r_a, r_b < m_j$. Then, we can obtain

$$\begin{aligned} (a+b) \bmod m_j &= (k_a \cdot m_j + r_a + k_b \cdot m_j + r_b) \bmod m_j \\ &= ((k_a + k_b) \cdot m_j + (r_a + r_b)) \bmod m_j \\ &= (r_a + r_b) \bmod m_j. \end{aligned}$$

Besides,

$$\begin{aligned} &(a \bmod m_j + b \bmod m_j) \bmod m_j \\ &= ((k_a \cdot m_j + r_a) \bmod m_j + \\ &(k_b \cdot m_j + r_b) \bmod m_j) \bmod m_j \\ &= (r_a \bmod m_j + r_b \bmod m_j) \bmod m_j \\ &= (r_a + r_b) \bmod m_j. \end{aligned}$$

So, the following conclusion holds true,

$$(a \bmod m_j + b \bmod m_j) \bmod m_j = (a+b) \bmod m_j,$$

namely,

$$\begin{aligned} &(T_Q(w) \bmod m_j + I_i(w) \bmod m_j) \bmod m_j \\ &= (T_Q(w) + I_i(w)\ ) \bmod m_j. \end{aligned}$$

Based on this conclusion, we can check that the **Theorem** 1 is correct.

The **Theorem** 1 shows that the share of the two original vectors' sum is equal to the sum of their corresponding shares under the same modulo prime. When getting the shares sent by CDSs, CSS can exactly reconstruct the original sum by integrating CRT's conclusions shown in Eq. 1 and Eq. 2. □

Moreover, in order to identify whether the matching results have been tampered, the **enhanced FARRIS** can verify the correctness of the returned results with the following theorem.

Theorem 2. The returned results are correct under *semi-honest-but-curious* model, provided that $H_{root}^{MT} = H_{root}^{MT'}$.

Proof: As introduced in Subsection 2.4, a Merkle hash tree (MHT) is constructed by using a number of leaf nodes and non-leaf nodes. Every leaf node is labeled with a hash value, which is calculated by hashing a specific data block. And every non-leaf node is labeled with the hash value of the concatenation of its two child nodes' hash values. Based on the construction process of MHT, the root node's hash value of an MHT depends on its hash function and the input data block. Since the hash function used in MHT is collision resistant, the input data block is the only factor that affects the value of the MHT root node. This is why MHT can be used to verify the correctness of the input data block. H_{root}^{MT} is the root nodes' hash value of the MHT MT, in which the message authentication codes (MAC) $\{\mathcal{M}'_{l_i}\}_{1 \le i \le t}$ are taken as the input data block. According to the Eq. 11, for any $i \in [1,t]$, $\{\mathcal{M}'_{l_i}\}_{1 \le i \le t}$ can be obtained by

$$\mathcal{M}'_{l_i} = \mathbf{MAC}_K(c'_{l_i} || id'_{l_i})$$

It indicates that $\mathcal{M}'_{l_i}$ depends on the encrypted surveillance video c'_{l_i} with ID id'_{l_i} and MAC key K. In our scheme, Only CO or AU has the MAC key K, which means that the CSS does not calculate the MAC values $\{\mathcal{M}'_{l_i}\}_{1 \le i \le t}$ by itself. Therefore, the H_{root}^{MT} by CSS is as the same as that of the unaltered $\{c'_{l_i}\}_{1 \le i \le t}$ equivalents. When obtaining the matching results $\{c'_{l_i}\}_{1 \le i \le t}$, AU itself can calculate the corresponding MAC values $\{\mathcal{M}''_{l_i}\}_{1 \le i \le t}$ with the help of MAC key K. Due to the unforgeable property of MAC, $H_{root}^{MT} = H_{root}^{MT'}$ holds true when the CSS correctly returns the corresponding matching videos $\{c'_{l_i}\}_{1 \le i \le t}$. However, the CSS or the third party may tamper the returned results $\{c'_{l_i}\}_{1 \le i \le t}$ under the *semi-honest-but-curious model*. If the tampering occurs, the MAC $\{\mathcal{M}''_{l_i}\}_{1 \le i \le t}$ values will be altered and differ from the MAC $\{\mathcal{M}'_{l_i}\}_{1 \le i \le t}$ stored in CSS, which leads to $H_{root}^{MT} \ne H_{root}^{MT'}$. Therefore, we can prove that the returned results are correct via $H_{root}^{MT} = H_{root}^{MT'}$. □

5.2 Security Analysis

In this subsection, we give the security analysis of our FARRIS in the form of the following theorems.

Theorem 3. In our scheme, it is infeasible for any polynomial-time CSS to recover the plaintext surveillance video v_i from its ciphertext version c_i or its corresponding MAC value $\mathcal{M}_i$.

Proof: In FARRIS, the general symmetric encryption is used to encrypt all surveillance video files. Without the corresponding encryption keys, it is impossible for the CSS

to learn about the details of the original videos. Only the authorized users AUs or the key holders COs can access the encrypted videos. In addition, **MAC**$(\cdot)$ used to generate MAC value is a secure MAC scheme with an negligible probability for breaking such **MAC**$(\cdot)$. Based on the above arguments, our scheme achieves the privacy of the outsourced surveillance videos and MAC values. □

Theorem 4. The proposed FARRIS can prevent the CDSs and CSS from recovering feature indexes under **Known Ciphertext Model** or **Known Background Model**.

Proof: Each feature index I_i by COs and query index T_Q are encrypted by using the CRT-based (k,n) secret sharing method. Different from the traditional encryption algorithms, no key is required in this encryption technique. And also, the secret sharing technique itself can guarantee that each share divided from any feature index does not leak any useful information. Even though the adversary is allowed to obtain more shares, it could not know the original feature index if and only if the number of the obtained shares is less than k under **Known Ciphertext Model**.

Furthermore, our FARRIS can support the stronger security model, **Known Background Model**. Under this model, more statistical information can be obtained by CDS. However, CDS still knows nothing about the feature indexes in the plaintext content. Two reasons can be explained as follows. *One* is that the secret sharing technique used to process feature indexes is information theoretically secure, which is well known and makes each share hidden from its related independent CDS. The preprocessing operation before generating the feature index is *another reason* for resisting the known background attack. Because, the random replacement rule (odd$\leftrightarrow$ 1, even$\leftrightarrow$ 0) makes it impossible for CDS to deduce the relation between the feature index and the received share. To achieve this high level of security in our scheme, the primes and scale factor s are chosen according to the following requirement, which is discussed in [44].

$$s < \left(\frac{\prod_{i=1}^{k} m_i - \varepsilon}{\Gamma} \right), \tag{13}$$

where $\{m_1, \cdots, m_k\}$ are selected primes out of n primes, and Γ is the maximum intermediate value before the scale processing. Generally, s should be larger than any prime out of n primes for obscuring each share. If the above inequality is satisfied, the individual CDSs know nothing about the feature indexes either from COs or AUs.

Furthermore, Only COs know how to perform the permutation and padding zeros, as well as the trained parameters of the feature extraction. This related information keeps secret from the CDS. Therefore, even if some plaintext person images and corresponding shares are obtained, CDS is still unable to build the relationship between the feature indexes and the corresponding shares. In short, CDS does not infer the specific contents from the index shares. In addition, the CSS is less likely to get any information about the feature indexes. The reason is that all it can get is the sum between indexes I_i and T_Q, and nothing else. Consequently, CSS learns nothing about the current content of the feature index I_i and query index T_Q. □

Theorem 5. Our FARRIS achieves query unlinkability and multi-user security.

Proof: As discussed before, the preprocessing operation for feature vectors differentiate the generated query index T_Q from the original feature vector. In particular, the replacement rule (odd$\leftrightarrow$ 1, even$\leftrightarrow$ 0) is done randomly for every time. Additionally, the noise ε in the scale process is also selected at random. Based on the two random operations, the distinct feature indexes T_Q are allowed to correspond to the same query Q. So, this non-deterministic index generation can provide query unlinkability for our scheme so that CDS could not decide whether the two different feature indexes come from the same query Q. Besides, the non-deterministic generation for the query Q also leads to the changes of the final reconstructed sum vector Sum^{Q,p_i}, thereby limiting CSS to distinguish the query index T_Q.

Additionally, the secret sharing technique is used to securely calculate the Hamming distance, where no key is required. The keyless characteristic naturally enables our scheme to support multi-user. Nobody can obtain useful information from any share. Even if partial CDSs collude with each other, it is difficult for CDSs to infer the information of the related feature indexes as long as the number of involved CDSs is less than k. □

5.3 Performance Analysis

In this subsection, we evaluate the performance of our scheme in terms of precision, computational overheads, and communication and storage costs.

Precision: The precision of our scheme is tested on person Re-ID benchmark: VIPeP [3] dataset, which is one of the most widely adopted datasets for Re-ID task. The dataset contains 1264 images taken from 632 persons, each of which is observed from two different surveillance cameras with arbitrary viewpoints and various lighting conditions. In the experiments, we randomly divide 632 people into $582/50$ and $532/100$ for the training/test set, where this random partition is repeated 10 times while taking the averaged performance as the final matching results. As for feature extraction, we combine the HIPHOP feature [4] and KSH technique to represent person Re-ID feature. Concretely, each person image is first resized into a uniform size of 227×227, which is required by AlexNet CNN. Following up, the feature maps from the first and second layer can be obtained in forward propagation manner. Then, one can calculate the HIPHOP feature descriptor by the combination of the fusion and rank technique. Finally, we employ KSH technique to convert the 84096-dim HIPHOP feature into a short hash code, which is taken as the final person Re-ID feature.

The evaluations are performed on a Lenovo laptop running windows 7-64bit with an I5-7200 2.5GHz processor, and 8GB Memory. The cumulative matching characteristic (CMC) curve is used to evaluate the matching precision of our scheme. CMC indicates that the probability of a query person in returned results with the different sizes. That is, it can provide the matching precision for each rank.

TABLE 2 and TABLE 3 show the matching precision of our scheme on different sized test sets, containing 50 persons and 100 persons, respectively. It is clear that our

scheme is effective and feasible. From either TABLE 2 or TABLE 3, the longer hash codes always achieve better matching rate than the shorter codes. The fact behind the result is that the hash code with the longer bit lengths can capture more distinguishing identity information in a space with large variations regarding viewpoints, lighting, and postures. It also indicates that the bit length can significantly affect the matching precision in Re-ID. However, the longer bit lengths may compromise the Re-ID efficiency because of high computation and storage costs, which will be discussed in the analysis of the computation, communication and storage costs.

In general, the discrimination of the hash codes could be improved with the increase of the trained samples. As shown in these two tables, the hash codes in the former test have the stronger capability to match truth identity than that in the latter test under the same rank and bit length. For example, for the rank 1 with 64 bits, the former with 1164 training samples can obtain the matching rate of 34.6%, a precision value about 12.6% higher than the latter with 1064 training samples. It is worth noting that the rank 1 matching rate with 64 bits in TABLE 2 is higher than that with 128 bits in TABLE 3. To some extent, more samples involved in the training process contribute to the generation of shorter hash codes, leading to the improvement of Re-ID efficiency.

TABLE 2
Matching rate (%) with different bit length of hash code on the test set with 50 persons.

Bit length	Rank 1	Rank 5	Rank 10	Rank 20	Rank 30
16 bits	17.2	45.0	62.0	83.8	93.2
32 bits	24.4	57.8	72.8	87.0	95.0
64 bits	34.6	64.8	80.2	91.2	95.4
128 bits	39.6	67.4	82.0	93.0	96.6
256 bits	47.4	74.8	87.4	94.2	97.2

TABLE 3
Matching rate (%) with different bit length of hash code on the test set with 100 persons.

Bit length	Rank 1	Rank 5	Rank 10	Rank 20	Rank 30
16 bits	11.5	33.2	47.4	66.2	76.4
32 bits	17.4	43.9	57.9	73.3	82.8
64 bits	22.0	48.7	62.6	74.8	83.1
128 bits	26.9	53.5	66.5	78.1	85.8
256 bits	31.2	59.2	71.8	84.1	90.0

Computation costs: In this subsection, we discuss the computation costs of our scheme in terms of theoretical analysis and actual computation efficiency analysis. To better analyze the theoretical costs, we first define some related notations as shown in TABLE 4.

As the core part of our FARRIS, the privacy-preserving Hamming distance computation is composed of five algorithms: **KeyGen**, **IndexBuild**, **QueryGen**, **SumComp**, and **SimComp**. For the computation costs analysis, we only analyze the latter four algorithms. There are two reasons for it. First, how to encrypt the surveillance videos and generate their corresponding MAC values is not an issue to be solved in our FARRIS, which is already explained in the previous **KeyGen** algorithm. Second, no encryption key is involved to compute the Hamming distance. Hence, we do not discuss the computation costs associated with the **KeyGen** algorithm here. TABLE 5 lists that the computation costs for these four algorithms. Especially, **IndexBuild** may has greater computation costs because that COs also encrypt the video files and generate the corresponding MACs. Since the two operations are not related to the secure Hamming distance computation, their corresponding costs are not listed in TABLE 5. Although **IndexBuild** algorithm has high computation costs, only one-time operation is required to carry out offline, and hence the Re-ID efficiency and the users' experience are not greatly affected. Except for **IndexBuild** algorithm, it is also found in the TABLE 5 that the **SimComp** algorithm is computationally expensive, where the involved modular multiplication inversion operation is a major factor. This result can be verified in the following actual efficiency analysis.

TABLE 4
Notation descriptions used in computation costs.

Notations	Descriptions
T_{mo}	Modulus operation
T_{mm}	Modular multiplication inversion operation
T_{po}	Permutation operation
T_{pz}	Padding zero operation
T_{mu}	Multiplication operation
T_{ao}	Addition operation

TABLE 5
Computation costs in different algorithms.

Algorithms	T_{mo}	T_{mm}	T_{po}	T_{pz}	T_{mu}	T_{ao}
IndexBuild	$n\ell g$	**N/A**	ℓ	ℓz	ℓg	ℓg
QueryGen	kg	**N/A**	1	z	g	g
SumComp	$k\ell g$	**N/A**	**N/A**	**N/A**	**N/A**	$k\ell g$
SimComp	ℓg	$k\ell g$	**N/A**	**N/A**	$(k^2-k)\ell g$	$(k-1)\ell g$

"ℓ": The number of person images by CO;
"z": The number of padding zeros;
"g": The dimension of feature index;
"(k,n)": The threshold of secret sharing;
"**N/A**": No related operations.

To guarantee the correctness of Re-ID results, we propose a new verifiable mechanism, which is an integral part of our FARRIS. So, it is necessary to analyze the computation costs with respect to the **verification** algorithm. In our FARRIS, the MHT is considered as an authentication technique to achieve the verification of the Re-ID results. As discussed in subsection 2.4, we know that the hash function h_T needs to be executed once per node, including root node, intermediate node, and leaf node. Based on the above analysis, the computation overhead in the **verification** algorithm typically relies on the amount of nodes in the MHT. According to the challenging information $\{l_i\}_{1\le i\le t}$, t identified data blocks (MACs or encrypted video data) are required to construct an MHT. It is not difficult to derive that the height of the corresponding Merkle hash tree is equal to $\lceil \log_2 t \rceil + 1$. By combining the generation characteristic of the MHT, the total of the nodes of the MHT can be computed as

$$N = \sum_{i=1}^{\lceil \log_2 t \rceil + 1} \left\lceil \frac{t}{2^{i-1}} \right\rceil.$$

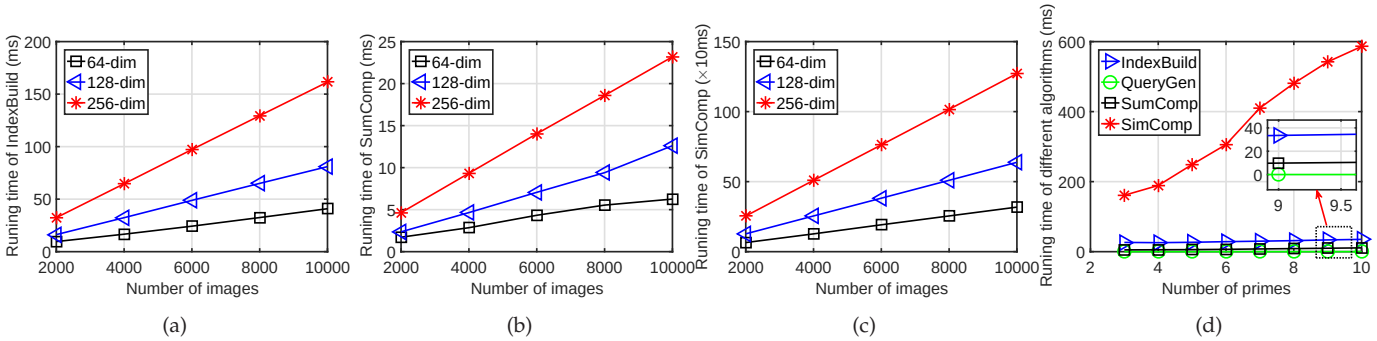


Fig. 6. Performance analysis in different algorithms: (a) Computational costs in **IndexBuild** algorithm; (b) Computational costs in **SumComp** algorithm; (c) Computational costs in **SimComp** algorithm; (d) Impact of k in different algorithms ($n = k$, $s = 80$, $\ell = 1000$, $g = 256$).

Assume that the computation costs of h_T and MAC are H and A, respectively. The verification algorithm will cost $4NH + tA$, in which $4NH$ is spent in 4 MHTs (AU and CSS have two trees each), and tA is used in computing the leaf nodes' MAC values of the tree MT' in AU.

With regard to the actual efficiency analysis, some experimental simulations written in C++ are conducted to show the computation costs on different algorithms, where the $(3, 5)$ secret sharing method with the prime set $\{29, 31, 37, 41, 43\}$ is adopted. In order to satisfy Eq. 13, we set the scale factor s as 47. Besides, Γ is set as 2^8 for the high level of feature security and efficient computations.

In Fig. 6(a), we show the computation costs of **IndexBuild** algorithm over different scale person image set. Obviously, the dataset with the larger size will consume more time at the same feature dimension. Besides, the dimension of the feature vector is also an important factor to affect the costs. If the dimension is greater, more costs will be spent to carry out the modulus operations, which are based on each element of the feature vector. The same trend is observed for **SumComp** and **SimComp** algorithms, shown in Fig. 6(b), and Fig. 6(c), respectively. Relatively speaking, the former does not yield too much computation costs. Specifically, when the number of key frames is 10000 and the feature dimension is 256, the entire costs of the 3 CDSs are 23.19 milliseconds only. The reason is that the **SumComp** algorithm only involves the modulus addition operation, which is suitable for CDSs with limited computation resources to handle. Moreover, we can also notice from Fig. 6(c) that CSS takes more time than the above algorithms. When the number of key frames is 10000 and the feature dimension is 256, CSS needs 1.27 seconds to reconstruct the 10000 sum vectors. It is obvious that the computation costs are greater than that in the former with millisecond level. The modular multiplication inverse used in **SimComp** algorithm is the major factor because its efficiency is far below than modular addition and multiplication. Although the higher computation costs of **SimComp** algorithm could straggle the whole Re-ID efficiency of our scheme, especially for large-scale dataset, it is easy to solve this problem by integrating the existing mature parallel techniques. The feasibility of this solution is based on the fact that each element of the sum vector Sum^{Q,p_i} can be reconstructed separately in FARRIS, and thus not interact with each other.

In Fig. 6(d), we also demonstrate the relation between the number of primes (or individual CDSs) and the computation costs. It is seen that, as the number of primes grows, it takes more much time to finish the privacy-preserving Hamming distance computation because of the increasing amount of the involved modulus operations. Especially, the computation costs of **SimComp** algorithm are far more than any others. It means that the user experience mainly depends on the execution speed of **SimComp** algorithm under the same conditions. To demonstrate the efficiency of different algorithms on larger dataset and more CDSs, we conduct a simulation experiment for 10000 feature vectors and up to 50 primes selected from the range between 37 and 283. The experimental result is shown in TABLE 6, where the dimension of feature vector is set to 256, and the scale factor s is equal to 300 for complying with Eq. 13. As we can see from TABLE 6, the computational costs of all algorithms show an upward tendency when the number of primes increases. It is remarkably obvious that the **SimComp** algorithm is the performance bottleneck because of lots of modular multiplication inversion operations. Under the same condition, **IndexBuild**, as well as **SumComp**, is faster than **SimComp** by two orders of magnitude, which is mainly due to the absence of the modular multiplication inversion. For the same computational level, more elementary operations are involved to make **IndexBuild** cost nearly as twice times as **SumComp** does on average. In comparison, the computational costs of **QueryGen** are much lower than those of the other algorithms under the same number of primes, since it only targets a frame image for a Re-ID request. In Table 6, we also list the total Re-ID time at the different numbers of primes. Clearly, the timing performance of Re-ID that is made up of **SumComp** and **SimComp** is mainly influenced by **SimComp** process.

TABLE 6
A comparison of different algorithms at computational costs (seconds) with different k where $\ell = 10000$, $s = 300$, and $g = 256$.

Algorithms	$k = 30$	$k = 40$	$k = 50$
QueryGen	2.32×10^{-4}	2.63×10^{-4}	3.61×10^{-4}
IndexBuild	0.623	0.726	0.822
SumComp	0.276	0.377	0.435
SimComp	20.4	27.5	34.7
Total Re-ID Time	20.7	27.9	35.2

Besides, we further test the complexity of the different algorithms with the same number of primes for our FARRIS,

where larger key frames collection size $\ell = 50000$, 80000, and 100000 than that of TABLE 6 are tested. The corresponding results are demonstrated in TABLE 7. It is not hard to find a homogeneous relationship in timing consumption among different algorithms if compared to TABLE 6. Specifically, it indicates that the largest computational costs occur in **SimComp**, following by **IndexBuild**, **SumComp**, and **QueryGen** in descending order. Also, a characteristic shared by the above algorithms except for **QueryGen** is that the time complexity rises as the dataset size increases. Based on the construction of **QueryGen**, it has nothing to do with the number of key frames. Therefore, the timing consumption of **QueryGen** keeps unchanged at different dataset sizes. It takes about 59.28 seconds to carry out person Re-ID for a given query request when the number of key frames is set to 100000. The matching costs can be sharply reduced by using parallel techniques. The reason is that each dimension of any feature vector does not communicate with others during the course of **SumComp** and feature reconstruction in **SimComp**. If the given g servers share the equivalent configuration, the total Re-ID time is reduced to $1/g$ of the original time theoretically, without considering task allocation and communication costs during the parallel process. Besides, the Re-ID is typically performed on the cloud servers, which occupy sufficient computational resources to ensure the efficiency performance. It means that our FARRIS is efficient in practical applications.

TABLE 7
A comparison of different algorithms at computational costs (seconds) with different ℓ where $k = 10$, $s = 180$, and $g = 256$.

Algorithms	$\ell = 50000$	$\ell = 80000$	$\ell = 100000$
QueryGen	1.72×10^{-4}	1.72×10^{-4}	1.72×10^{-4}
IndexBuild	1.98	3.09	3.79
SumComp	0.349	0.563	0.705
SimComp	29.7	47.8	58.5
Total Re-ID Time	30.1	48.4	59.3

Note that the scale modification $u \cdot s + \varepsilon$ (hereafter referred to as SM) in the feature preprocessing is a critical step before executing the above algorithms. To some extent, it increases the computational costs of the whole Re-ID system by introducing ample multiplication and addition operations. However, it can further improve the security of the feature vector. Here, we add some experiments to show its necessity to the establishment of our FARRIS. For better illustration purpose, we view an image as a feature vector. Examples of image shares with/without SM are shown in Fig. 7, where shares are generated by using modulo prime 151 operations based on CRT-based secret sharing technique. The results from the second row of Fig. 7 show that the partial content of each image is leaked when directly carrying out mod 151 operations for all pixels of the image. In this case, when obtaining the image share without SM, CDS can immediately know the rough content of the original image. It means that the privacy of the feature vector cannot be guaranteed. On the contrary, the image shares with the SM look like random noise images, with the scale factor $s = 160$, and random noise $\varepsilon \in (0, 160)$. And it suggests that images after SM do not reveal any visual information.

Fig. 7. Examples of images with/without the scale modification. First row: ten original person images randomly selected from VIPeR [3]. Second row: the corresponding shares of the first row without the scale modification; Third row: the corresponding shares of the first row with the scale modification.

Furthermore, the SM is robust against the histogram analysis. For simplicity, we select the first two columns of Fig. 7 to count their respective histograms. It can be observed from Fig. 8 that the distributions without SM still resemble those of the corresponding original images. It is easy for the attacker to infer useful information by analyzing the statistical histogram. However, with the SM, the generated histograms follow a uniform distribution, which are clearly displayed in the third column of Fig. 8. It is very difficult for the attackers to extract useful information from this type of histograms.

TABLE 8
Comparison of SSIM and information entropy with/without SM.

Image	SSIM		Entropy	
	without *SM*	with *SM*	without *SM*	with *SM*
1	0.6452	**0.0467**	7.0583	**7.2330**
2	0.2950	**0.0273**	6.9136	**7.2333**
3	0.2679	**0.0461**	7.2125	**7.2324**
4	0.2796	**0.0162**	7.1399	**7.2322**
5	0.1684	**0.0378**	7.1607	**7.2337**
6	0.5536	**0.0407**	7.0171	**7.2345**
7	0.5221	**0.0440**	7.0500	**7.2329**
8	0.5690	**0.0580**	6.9193	**7.2326**
9	0.5016	**0.0411**	6.9718	**7.2327**
10	0.5723	**0.0336**	6.9945	**7.2340**

In addition, we also adopt the structural similarity (SSIM) index [45] and information entropy to measure the difference between the original feature and the feature share with/without SM. The results are shown in TABLE 8, in which images correspond to those in the first row of Fig. 7 one by one. For example, the image labeled 1 in TABLE 8 refer to the first image in the first row of Fig. 7. Obviously, the SSIM values after SM are far below those before SM operation. It implies that the internal structure of the feature is more unrecognizable than the latter. As for the information entropy, the higher it is, the more uniform the feature element value distribution is, which means that the feature privacy can be better protected. According to the figures in

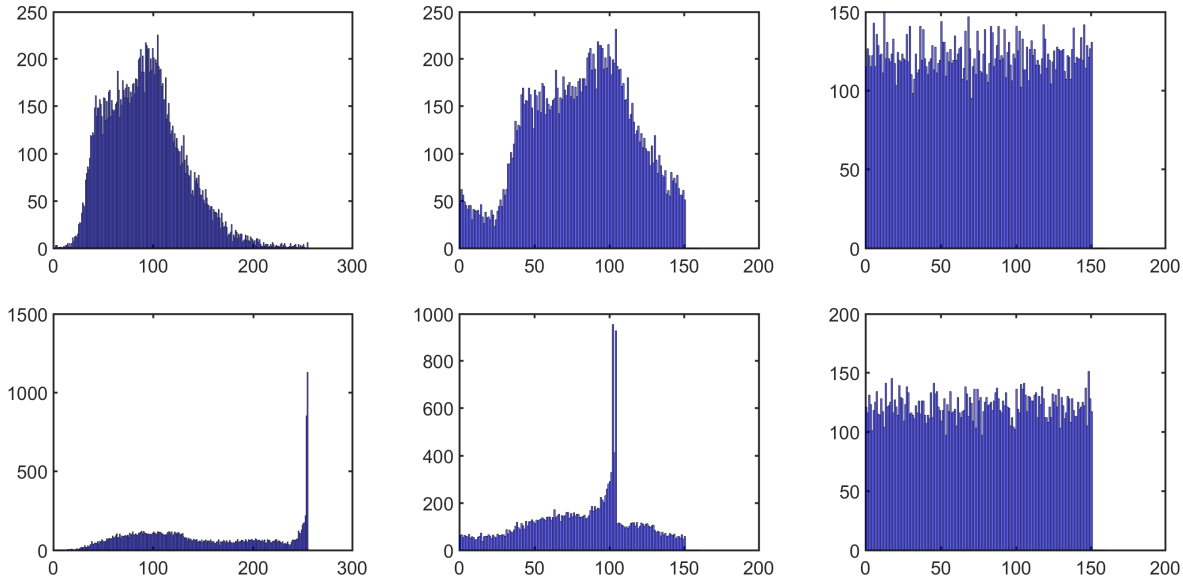


Fig. 8. The histograms for the first two columns of Fig. 7. Left: the original images. Middle: the image shares without SM. Right: the image shares with SM.

TABLE 8, the average information entropy for the images with SM is 7.2331, while 7.0438 for images without SM operation. Theoretically, for an image under modulo prime 151, the upper limit of its information entropy is about 7.2384. Therefore, we can conclude that the SM indeed introduces more randomness to the feature elements and conceal the feature information better.

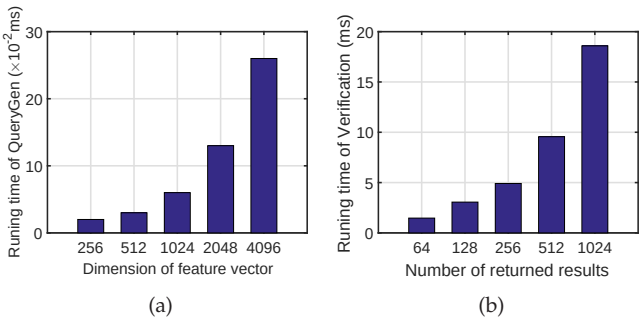


Fig. 9. Performance analysis in different algorithms. (a) Computational costs in **QueryGen** algorithm; (b) Computational costs in **Verification** algorithm.

In Fig. 9(a), we show the computation burden of query index with the different feature dimensions during **QueryGen** process. As the dimension size escalates, the corresponding costs also increase. However, the gain of the computation costs is relatively trivial. For example, when the dimension size is 4096, it takes around 0.26 milliseconds, and almost negligible in the total timing cost of FARRIS. What's more, the feature vector in FARRIS is encoded into a compact hash code, the dimension of which is usually much lower than 4096 in practical applications. Except for the costs taken for **QueryGen** process, AU also needs some computation costs to verify whether the returned results have been tampered by building the dual MHTs. Fig. 9(b) shows that the computation costs in the **Verification** process grow with the increase of the Re-ID results, again it does not take much time. Specifically, about 18.59 milliseconds are required to verify the correctness of 1024 results. In general, these costs on the millisecond level are relatively low and do not affect the user Re-ID experience, which is suitable for resource-limited AUs, such as mobile users. It should be noted that, given a fixed t, the construction time of the MHT in AU is slightly longer than that in CSS due to the need for generating the MAC set $\{\mathcal{M}''_{l_i}\}_{1\leq i\leq t}$ for AU itself. In the experimental simulation of **Verification** algorithm, we adopt the SHA256 hash function to compute the MHT nodes' hash values and MACs.

TABLE 9
Communication and storage costs in different algorithms.

Algorithms	Communication costs (bits)	Storage costs (bits)
IndexBuild	$n\ell g\|P\|$ $+m(\|E\|+\|D\|+\|A\|)$	$n\ell g\|P\|$ $+m(\|E\|+\|D\|+\|A\|)$
QueryGen	$kg\|P\|$	$kg\|P\|$
SumComp	$k\ell g\|P\|$	$k\ell g\|P\|$
SimComp	$q(\|E\|+\|D\|)$	$k\ell g\|\Gamma\|+q(\|E\|+\|D\|)$
Verification	$tlog_2q+2\|H\|$	$4N\|H\|+t\|A\|$

Communication and storage costs: To express these two types of costs more clearly, we assume that $|P|$ is the bit-length of the maximum prime selected by the (k, n) secret sharing method. Denote $|E|$, $|D|$, $|A|$, $|H|$, and $|\Gamma|$ as the bit-length of the encrypted video file with the maximum file size, ID, MAC, h_T value, and Γ, respectively. TABLE 9 gives the theoretic analysis of the communication and storage costs caused by different algorithms in FARRIS. In **IndexBuild** algorithm, each preprocessed feature vector in ℓ person images is divided into n shares, and sent into the corresponding CDS. After that, it generates $n\ell g|P|$ communication and storage costs, respectively. Besides, these two types of the costs will further increase by $m(|E|+|D|+|A|)$ due to the generation of m encrypted video files, IDs, and MACs. Given a preprocessed feature index T_Q, AU splits

it into k shares, and randomly submits them to k out of n CDSs. It is easy to compute the communication and storage costs presented in **QueryGen** algorithm in this table. Different from **QueryGen** algorithm, ℓ person image shares in **SumComp** process are involved to compute the sum Sum_j^{Q,p_i} between $T_{Q,j}$ and each $I_{i,j}$ ($i \in [1,\ell]$) in k CDSs d_j, which produces ℓ times as much as the costs in **QueryGen** process. After the reconstruction of all ℓ sum vectors, CSS will scale them down by the scale factor s. As a result, $k\ell g|\Gamma|$ bits are used to store the final reconstruction data in **SimComp** process. AU requests for returning the q most related videos and corresponding IDs, and thus CSS generates $q(|E|+|D|)$ bits of data in response to the request, and followed by storing the data in AU. In **verification** process, AU sends the challenging information $\{l_i\}_{1\leq i\leq t}$ with $tlog_2q$ bits to CSS. When obtaining the challenging request, CSS builds the dual MHTs to respond, which requires $2|H|$ bits for storing the hash values of two root nodes, and in total uses $2N|H|$ bits for hashing N nodes. Accordingly, the AU also generates the dual MHTs to justify the correctness of the Re-ID results, in which an extra storage costs with $t|A|$ bits are introduced to store t MACs computed by itself, except the costs as the same as that of constructing the dual MHTs in CSS.

6 Related Work

In this paper, person Re-ID mainly refers to person retrieval. To some extent, person Re-ID can also be regarded as the issue of image-to-video retrieval for two reasons. First, a static image is taken as a query image in these two retrieval issues. Second, the goal of both tasks focuses on identifying which videos within the video dataset are the best matches for a given visual input. Since query images and database videos contain different types of spatiotemporal information, searching videos using images is an asymmetric issue [9], which poses a challenge to the image-to-video retrieval. In general, a video with one second duration comprises 24 up to 30 frames. It means that even a few videos can cause tremendous data storage. For example, 12 videos, each of which is recorded for an hour, contain over one million images. Based on the result, it becomes infeasible to directly employ the image retrieval technique to solve the asymmetric issue. An alternative technique, namely temporal aggregation, is to remove the inter- or intra-frame redundancies in video for acquiring more compact representations [16]. Most of the temporal aggregation algorithms fall into two categories: local features based reduction [11], [12] and holistic aggregation [9], [13]. The former aims at reducing the number of local features extracted from each frame of the video such that the matching speed is improved. It is widely accepted that the local features based reduction can achieve better matching due to its ability to capture more-refined feature information. By contrast, the latter compresses multiple frames information of video into a compact global feature vector, thus the video retrieval using query image can be dramatically accelerated. However, resource-constrained users cannot afford high computation overheads and storage costs as the amount of video data increases.

Due to the popularity of cloud computing technique, users tend to store their multimedia data to the cloud server in consideration of costs saving and convenience. However, the privacy of the outsourced data may be leaked by the cloud [46], [47]. In cloud computing environment, the problem of privacy-preserving retrieval over encrypted data has been studied for many years. In early days, most of the research schemes, such as [48], [49], [50], [51], mainly focus on SE technique, which takes text files as the objective. As far as we know, Song et al. [48] developed the first SE scheme to allow the users to search over encrypted text files. Using this scheme, the large computation costs will be generated because of the word-wise operation for both encryption and search. Worse still, the lack of index is also a disadvantage. To mitigate the above issues, Bloom filter technique was introduced by Goh [49] to construct a secure index for improving the search speed. Furthermore, Curtmola et al. [50] leveraged the inverted index to obtain better search efficiency. The above SE schemes are based on symmetric encryption, in which the private key for encryption and decryption is the same, making key distribution inconvenient. An alternative direction in SE is developed to perform secure search on encrypted data by utilizing public key mechanism. The first public key SE was presented by Boneh et al. [51] in 2004. With this scheme, the search efficiency decreases as the encrypted keywords in each text file increase. The above SE schemes can achieve searching over encrypted text files under the assumption of *honest-but-curious* threat model. However, in a practical application, the cloud server may return a fraction of false matching results to users to reduce computational overhead or conceal data loss. In other words, the cloud server should be *semi-honest-but-curious*. To address this problem, Sun et al. [25] proposed a scalable search authorization based on the file level. Using this scheme, a larger number of attributes readily lead to high computation costs, which limits the practical deployment. Miao et al. [26] constructed a verifiable search scheme based on bilinear map. With this scheme, an independent private audit server is employed to verify the correctness of the returned search results. Also, some dynamic operations (e.g., document addition, modification and deletion) are not allowed. In [27], Liu et al. proposed a novel verifiable SE scheme, which not only supports dynamic operations but also allows multiple users to participate. Recently, some SE approaches with various functionalities have been put forward, such as secure multi-keyword ranked search [52], [53], [54], semantic search [55], and other functionalities [56], [57], [58].

From the perspective of computer vision, person Re-ID can be treated as image retrieval problem to some extent [59]. Therefore, we can take encrypted image retrieval as a powerful reference to investigate secure person Re-Id over cloud video data. With the rapid development of the imaging devices (e.g., digital cameras, smart phones), the amount of image data is increasing dramatically. Therefore, the approaches related to secure image retrieval have also been studied extensively in recent years. The work in [18] presented the three distance-preserving methods for feature protection, ensuring the distance between features remains approximately invariant before and after encryption. Two secure efficient search indexes based on order-preserving

encryption [60] and min-hash algorithm were introduced to perform privacy-preserving retrieval for large scale image databases in [19]. But the retrieval performance is not sufficient to serve the practical application. Another method [61] has been introduced for secure image retrieval by using homomorphic encryption (HE). The high computation complexity and communication costs, however, are beyond the reach of users. In [21], Xia et al. employed the asymmetric scalar-product preserving encryption technique [62] to encrypt image features, while a watermark-based protocol was designed to perform the privacy-preserving copy-deterrence. Yuan et al. [42] discussed the problem of secure content-based large-scale image search. The lightweight multi-level HE is used for image retrieval process in this literature. A very recent work [22] was proposed to achieve high accuracy and efficiency of privacy-wise image retrieval. Its main idea is to use transformed convolutional neural network to extract high accuracy feature while devising a hierarchical index tree to boost retrieval speed. Huang et al. [23] developed a secure relevance feedback mechanism to improve retrieval performance while maintaining the user privacy. However, the above secure image retrieval schemes need to perform the feature extraction/encryption independently, which brings inconvenience to users. Considering the universality of JPEG image format, Cheng et al. [63], [64] proposed secure image retrieval schemes for JPEG images, where the feature extraction/encryption is needless, but it is indispensable in the schemes [18], [19], [21], [22], [23], [42], [61]. However, the characteristic of the feature statistical invariance before and after encryption is vulnerable to feature guessing attacks. The same problem also exists in [65].

Up to date, much less work has been done for privacy-preserving video retrieval. In [17], Lu et al. provided a feasible solution, where visual features extracted from a set of frames of the video can be protected by using distance-preserving encryption. During video retrieval, the cloud server can calculate the similarity between encrypted videos through accumulating the distances of their corresponding features, without learning about the query and database videos in the plaintext. Although the retrieval approaches over encrypted text files/images have been developed for many years, no such scheme that can support privacy-preserving person Re-ID over encrypted outsourced videos as well as simultaneous verifiable retrieval exists. Therefore, we propose FARRIS to achieve the above goals by combining the keyless secret sharing method and MHT technique. Furthermore, FARRIS is compatible with any secure retrieval scheme that takes Hamming distance as matching measure. TABLE 10 gives the comparison of our scheme with other related schemes under different functionalities.

TABLE 10
Functional comparison in various schemes.

Schemes	Func 1	Func 2	Func 3	Func 4	Func 5
[4], [8]	✓				
[17], [21], [23], [42]		✓			
[25], [26], [27]		✓		✓	
[22]		✓			✓
Basic FARRIS	✓	✓	✓		✓
Enhanced FARRIS	✓	✓	✓	✓	✓

"Func 1": Person Re-ID;
"Func 2": Privacy preserving;
"Func 3": Keyless feature encryption;
"Func 4": Verifiable search;
"Func 5": Compatibility.

7 Conclusions

In this paper, we propose a novel privacy-preserving person Re-ID scheme over outsourced surveillance videos, which provides high accurate feature in the binary form by synthesizing the CNN model and KSH technique. In order to securely measure the similarities among feature vectors, we construct a privacy-preserving Hamming distance computation protocol based on CRT-based secret sharing. Besides, a dual MHTs based verification mechanism is proposed to identify the correctness of the matching results. The security analysis shows that our FARRIS can guarantee person Re-ID efficiently, without revealing the privacy of related persons. Furthermore, the theoretic analysis and practical simulations demonstrate that FARRIS is effective and feasible.

In the future, we plan to investigate the optimal relationship between the dimension of feature index and the number of cloud servers, which aims at achieving better user experience as well as preserving the privacy of users. Moreover, we will further improve the efficiency of secure person Re-ID in a real-world environment.

Acknowledgment

The authors thank the Associate Editor and reviewers for their constructive and generous feedback. We thank Dr. Jingwei Hu for his linguistic assistance during the preparation of this manuscript. This work was supported by the National Natural Science Foundation of China (No. U1804263, No. 61702105), the Natural Science Foundation of Fujian Province (No. 2015J01013, No. 2016J01016, No. 2017J01502, No. 2017J01555, and No. 2017J01751), Education Research Project for Young and Middle-aged Teachers of the Education Department of Fujian Province (No. JAT170185), the Scientific Research Foundation of Fuzhou University (No. 510483, No. 50010885), the National Research Foundation, Prime Minister's Office, Singapore under its Strategic Capability Research Centres Funding Initiative, and Singapore Ministry of Education under Research Grant MOE2016-T2-2-014(S).

References

[1] "Smart cameras catch man in 60,000 crowd," *BBC News*, April 13, 2018, https://www.bbc.com/news/world-asia-china-43751276.
[2] "Video surveillance market to 2025 - global analysis and forecasts by platforms (hardware and software)," https://www.researchandmarkets.com/research/zn5s9z/global_video?w=5.
[3] D. Gray, S. Brennan, and H. Tao, "Evaluating appearance models for recognition, reacquisition, and tracking," in *Proc. IEEE International Workshop on Performance Evaluation for Tracking and Surveillance (PETS)*, vol. 3, no. 5. Citeseer, 2007, pp. 1–7.
[4] Y.-C. Chen, X. Zhu, W.-S. Zheng, and J.-H. Lai, "Person re-identification by camera correlation aware feature augmentation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 2, pp. 392–408, 2018.
[5] L. Tian, H. Wang, Y. Zhou, and C. Peng, "Video big data in smart city: Background construction and optimization for surveillance video processing," *Future Generation Computer Systems*, 2018.

[6] X. Liu, R. Deng, K.-K. R. Choo, Y. Yang, and H. Pang, "Privacy-preserving outsourced calculation toolkit in the cloud," *IEEE Transactions on Dependable and Secure Computing*, 2018.

[7] X. Liu, K.-K. R. Choo, R. H. Deng, R. Lu, and J. Weng, "Efficient and privacy-preserving outsourced calculation of rational numbers," *IEEE Transactions on Dependable and Secure Computing*, vol. 15, no. 1, pp. 27–39, 2018.

[8] L. Zheng, Y. Yang, and A. G. Hauptmann, "Person re-identification: Past, present and future," *arXiv preprint arXiv:1610.02984*, 2016.

[9] A. Araujo and B. Girod, "Large-scale video retrieval using image queries," *IEEE transactions on circuits and systems for video technology*, vol. 28, no. 6, pp. 1406–1420, 2018.

[10] L. Zheng, Y. Yang, and Q. Tian, "Sift meets cnn: A decade survey of instance retrieval," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 5, pp. 1224–1244, 2018.

[11] A. Anjulan and N. Canagarajah, "Object based video retrieval with local region tracking," *Signal Processing: Image Communication*, vol. 22, no. 7-8, pp. 607–621, 2007.

[12] A. Araujo, M. Makar, V. Chandrasekhar, D. Chen, S. Tsai, H. Chen, R. Angst, and B. Girod, "Efficient video search using image queries," in *Image Processing (ICIP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 3082–3086.

[13] C.-Z. Zhu and S. Satoh, "Large vocabulary quantization for searching instances from videos," in *Proceedings of the 2nd ACM International Conference on Multimedia Retrieval*. ACM, 2012, p. 52.

[14] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.

[15] J. Sánchez, F. Perronnin, T. Mensink, and J. Verbeek, "Image classification with the fisher vector: Theory and practice," *International journal of computer vision*, vol. 105, no. 3, pp. 222–245, 2013.

[16] N. Garcia, "Temporal aggregation of visual features for large-scale image-to-video retrieval," in *Proceedings of the 2018 ACM on International Conference on Multimedia Retrieval*. ACM, 2018, pp. 489–492.

[17] W. Lu, A. Varna, and M. Wu, "Secure video processing: Problems and challenges," in *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*. IEEE, 2011, pp. 5856–5859.

[18] W. Lu, A. L. Varna, A. Swaminathan, and M. Wu, "Secure image retrieval through feature protection," in *Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on*. IEEE, 2009, pp. 1533–1536.

[19] W. Lu, A. Swaminathan, A. L. Varna, and M. Wu, "Enabling search over encrypted multimedia databases," in *Media Forensics and Security*, vol. 7254. International Society for Optics and Photonics, 2009, p. 725418.

[20] L. Weng, L. Amsaleg, A. Morton, and S. Marchand-Maillet, "A privacy-preserving framework for large-scale content-based information retrieval." *IEEE Trans. Information Forensics and Security*, vol. 10, no. 1, pp. 152–167, 2015.

[21] Z. Xia, X. Wang, L. Zhang, Z. Qin, X. Sun, and K. Ren, "A privacy-preserving and copy-deterrence content-based image retrieval scheme in cloud computing," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 11, pp. 2594–2608, 2016.

[22] X. Li, Q. Xue, and M. C. Chuah, "Casheirs: Cloud assisted scalable hierarchical encrypted based image retrieval system," in *INFOCOM 2017-IEEE Conference on Computer Communications, IEEE*. IEEE, 2017, pp. 1–9.

[23] Y. Huang, J. Zhang, L. Pan, and Y. Xiang, "Privacy protection in interactive content based image retrieval," *IEEE Transactions on Dependable and Secure Computing*, 2018.

[24] Q. Chai and G. Gong, "Verifiable symmetric searchable encryption for semi-honest-but-curious cloud servers," in *Communications (ICC), 2012 IEEE International Conference on*. IEEE, 2012, pp. 917–922.

[25] W. Sun, S. Yu, W. Lou, Y. T. Hou, and H. Li, "Protecting your right: verifiable attribute-based keyword search with fine-grained owner-enforced search authorization in the cloud," *IEEE Transactions on Parallel and Distributed Systems*, vol. 27, no. 4, pp. 1187–1198, 2016.

[26] Y. Miao, J. Weng, X. Liu, K.-K. R. Choo, Z. Liu, and H. Li, "Enabling verifiable multiple keywords search over encrypted cloud data," *Information Sciences*, vol. 465, pp. 21–37, 2018.

[27] X. Liu, G. Yang, Y. Mu, and R. Deng, "Multi-user verifiable searchable symmetric encryption for cloud storage," *IEEE Transactions on Dependable and Secure Computing*, 2018.

[28] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[29] A. Sharma, O. Tuzel, and D. W. Jacobs, "Deep hierarchical parsing for semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 530–538.

[30] D. Yi, Z. Lei, S. Liao, and S. Z. Li, "Deep metric learning for person re-identification," in *Pattern Recognition (ICPR), 2014 22nd International Conference on*. IEEE, 2014, pp. 34–39.

[31] W. Li, R. Zhao, T. Xiao, and X. Wang, "Deepreid: Deep filter pairing neural network for person re-identification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 152–159.

[32] H. Jégou, M. Douze, C. Schmid, and P. Pérez, "Aggregating local descriptors into a compact image representation," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE, 2010, pp. 3304–3311.

[33] Y. Gong, S. Lazebnik, A. Gordo, and F. Perronnin, "Iterative quantization: A procrustean approach to learning binary codes for large-scale image retrieval," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 12, pp. 2916–2929, 2013.

[34] W. Liu, J. Wang, R. Ji, Y.-G. Jiang, and S.-F. Chang, "Supervised hashing with kernels," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE, 2012, pp. 2074–2081.

[35] W. Liu, J. C. Principe, and S. Haykin, *Kernel adaptive filtering: a comprehensive introduction*. John Wiley & Sons, 2011, vol. 57.

[36] M. Datar, N. Immorlica, P. Indyk, and V. S. Mirrokni, "Locality-sensitive hashing scheme based on p-stable distributions," in *Proceedings of the twentieth annual symposium on Computational geometry*. ACM, 2004, pp. 253–262.

[37] J. Wang, S. Kumar, and S.-F. Chang, "Semi-supervised hashing for large-scale search," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, no. 12, pp. 2393–2406, 2012.

[38] C. Strecha, A. Bronstein, M. Bronstein, and P. Fua, "Ldahash: Improved matching with smaller descriptors," *IEEE transactions on pattern analysis and machine intelligence*, vol. 34, no. 1, pp. 66–78, 2012.

[39] A. Shamir, "How to share a secret," *Communications of the ACM*, vol. 22, no. 11, pp. 612–613, 1979.

[40] M. Mignotte, "How to share a secret," in *Workshop on Cryptography*. Springer, 1982, pp. 371–375.

[41] P. Devanbu, M. Gertz, C. Martel, and S. G. Stubblebine, "Authentic data publication over the internet 1," *Journal of Computer Security*, vol. 11, no. 3, pp. 291–314, 2003.

[42] L. Zhang, T. Jung, K. Liu, X.-Y. Li, X. Ding, J. Gu, and Y. Liu, "Pic: Enable large-scale privacy preserving content-based image search on cloud," *IEEE Transactions on Parallel and Distributed Systems*, vol. 28, no. 11, pp. 3258–3271, 2017.

[43] J. Li, X. Liang, S. Shen, T. Xu, J. Feng, and S. Yan, "Scale-aware fast r-cnn for pedestrian detection," *IEEE Transactions on Multimedia*, vol. 20, no. 4, pp. 985–996, 2018.

[44] M. Upmanyu, A. M. Namboodiri, K. Srinathan, and C. Jawahar, "Efficient privacy preserving video surveillance," in *Computer Vision, 2009 IEEE 12th International Conference on*. IEEE, 2009, pp. 1639–1646.

[45] Z. Wang, A. C. Bovik, H. R. Sheikh, E. P. Simoncelli *et al.*, "Image quality assessment: from error visibility to structural similarity," *IEEE transactions on image processing*, vol. 13, no. 4, pp. 600–612, 2004.

[46] Y. Zhang, C. Xu, X. Liang, H. Li, Y. Mu, and X. Zhang, "Efficient public verification of data integrity for cloud storage systems from indistinguishability obfuscation," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 3, pp. 676–688, 2017.

[47] Y. Zhang, C. Xu, H. Li, K. Yang, J. Zhou, and X. Lin, "Healthdep: An efficient and secure deduplication scheme for cloud-assisted ehealth systems," *IEEE Transactions on Industrial Informatics*, 2018.

[48] D. X. Song, D. Wagner, and A. Perrig, "Practical techniques for searches on encrypted data," in *Proc. IEEE Symposium on Security and Privacy (S&P'00)*. IEEE, 2000, pp. 44–55.

[49] E.-J. Goh *et al.*, "Secure indexes." *IACR Cryptology ePrint Archive*, vol. 2003, p. 216, 2003.

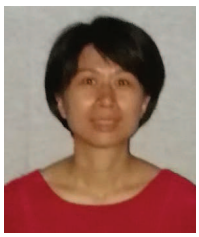
[50] R. Curtmola, J. Garay, S. Kamara, and R. Ostrovsky, "Searchable symmetric encryption: improved definitions and efficient con-

structions," *Journal of Computer Security*, vol. 19, no. 5, pp. 895–934, 2011.
[51] D. Boneh, G. Di Crescenzo, R. Ostrovsky, and G. Persiano, "Public key encryption with keyword search," in *Advances in Cryptology-Eurocrypt 2004*. Springer, 2004, pp. 506–522.
[52] 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.
[53] Y. Yang, X. Liu, and R. Deng, "Multi-user multi-keyword rank search over encrypted data in arbitrary language," *IEEE Transactions on Dependable and Secure Computing*, no. 1, 2017.
[54] C. Chen, X. Zhu, P. Shen, J. Hu, S. Guo, Z. Tari, and A. Y. Zomaya, "An efficient privacy-preserving ranked keyword search method," *IEEE Transactions on Parallel and Distributed Systems*, vol. 27, no. 4, pp. 951–963, 2016.
[55] Z. Fu, F. Huang, K. Ren, J. Weng, and C. Wang, "Privacy-preserving smart semantic search based on conceptual graphs over encrypted outsourced data," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 8, pp. 1874–1884, Aug 2017.
[56] D. He, M. Ma, S. Zeadally, N. Kumar, and K. Liang, "Certificateless public key authenticated encryption with keyword search for industrial internet of things," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 8, pp. 3618–3627, 2018.
[57] J. Li, Y. Huang, Y. Wei, S. Lv, Z. Liu, C. Dong, and W. Lou, "Searchable symmetric encryption with forward search privacy," *IEEE Transactions on Dependable and Secure Computing*, 2019.
[58] X. Zhang, H. Wang, and C. Xu, "Identity-based key-exposure resilient cloud storage public auditing scheme from lattices," *Information Sciences*, vol. 472, pp. 223–234, 2019.
[59] L. Zheng, L. Shen, L. Tian, S. Wang, J. Bu, and Q. Tian, "Person re-identification meets image search," *arXiv preprint arXiv:1502.02171*, 2015.
[60] R. Agrawal, J. Kiernan, R. Srikant, and Y. Xu, "Order preserving encryption for numeric data," in *Proceedings of the 2004 ACM SIGMOD international conference on Management of data*. ACM, 2004, pp. 563–574.
[61] W. Lu, A. L. Varna, and M. Wu, "Confidentiality-preserving image search: a comparative study between homomorphic encryption and distance-preserving randomization," *IEEE Access*, vol. 2, pp. 125–141, 2014.
[62] W. K. Wong, D. W.-l. Cheung, B. Kao, and N. Mamoulis, "Secure knn computation on encrypted databases," in *Proceedings of the 2009 ACM SIGMOD International Conference on Management of data*. ACM, 2009, pp. 139–152.
[63] H. Cheng, X. Zhang, J. Yu, and Y. Zhang, "Encrypted jpeg image retrieval using block-wise feature comparison," *Journal of Visual Communication and Image Representation*, vol. 40, pp. 111–117, 2016.
[64] H. Cheng, X. Zhang, J. Yu, and F. Li, "Markov process-based retrieval for encrypted jpeg images," *EURASIP Journal on Information Security*, vol. 2016, no. 1, p. 1, 2016.
[65] B. Ferreira, J. Rodrigues, J. Leitao, and H. Domingos, "Practical privacy-preserving content-based retrieval in cloud image repositories," *IEEE Transactions on Cloud Computing*, 2017.

Hang Cheng received his BS and MS degrees in applied mathematics from Fuzhou University, Fuzhou, China, in 2002 and 2005, respectively, and PhD in signal and information processing with Shanghai University, Shanghai, China, in 2016. He is an associate professor in the Department of Information and Computational Science, College of Mathematics and Computer Science, Fuzhou University, Fuzhou, China. He is a research scholar in the School of Physical and Mathematical Sciences, Nanyang Technological University, Singapore. His current research interests include multimedia security, image processing, cryptography, and information hiding.

Huaxiong Wang received the PhD degree in mathematics from the University of Haifa, Israel, in 1996 and the PhD degree in computer science from the University of Wollongong, Australia, in 2001. He joined Nanyang Technological University in 2006 and is currently an associate professor in the Division of Mathematical Sciences. He is also an honorary fellow at Macquarie University, Australia. His research interests include cryptography, information security, coding theory, combinatorics, and theoretical computer science. He has been on the editorial board of three international journals: Designs, Codes and Cryptography (2006-2011), the Journal of Communications (JCM), and Journal of Communications and Networks. He was the program cochair of Ninth Australasian Conference on Information Security and Privacy (ACISP 04) in 2004 and Fourth International Conference on Cryptology and Network Security (CANS 05) in 2005, and has served in the program committee for more than 70 international conferences. He received the inaugural Award of Best Research Contribution from the Computer Science Association of Australasia in 2004.

Ximeng Liu (M'16) received the B.E. degree with the Department of Electronic Engineering from Xidian University, Xi'an, China, in 2010 and Ph.D. degree with the Department of Telecommunication Engineering from Xidian University, Xi'an, China in 2015. He is currently a postdoctoral fellow with the Department of Information System, Singapore Management University, Singapore. And he is also a professor in the College of Mathematics and Computer Science, Fuzhou University, Fuzhou, China. His research interests include applied cryptography and big data security. He is a member of the IEEE.



Yan Fang received her BS and MS degrees in information and computation science from Fuzhou University, Fuzhou, China, in 2003 and 2006, respectively. She is a lecturer in the College of Computer and Information Sciences, Fujian Agriculture and Forestry University, Fuzhou, China. Her current research interests include multimedia security, and image processing.

Meiqing Wang received her BS and MS degrees in applied mathematics from Tsinghua University, Beijing, China, in 1987 and 1989, respectively, and PhD in Department of Computing, Xi'an Jiaotong University, China, in 2002. Now, she is a professor in the Department of Information and Computational Science, College of Mathematics and Computer Science, Fuzhou University, Fuzhou, China. Her current research interests include computing science, image processing, and computational finance.

Xiaojun Zhang received the Ph.D. degree in information security from the University of Electronic Science Technology of China (UESTC), Chengdu, China, in 2015. He is a lecturer in the School of Computer Science, Southwest Petroleum University, Chengdu, China. He also works as a Postdoctoral Fellow in University of Electronic Science Technology of China from 2016. He is a research scholar in the School of Physical and Mathematical Sciences, Nanyang Technological University, Singapore. His research interests include cryptography, network security, and cloud computing security. He is a member of the China Association for Cryptologic Research.