Enabling Secure Cross-Modal Retrieval Over Encrypted Heterogeneous IoT Databases With Collective Matrix Factorization

Cheng Guo, *Member, IEEE*, Jing Jia, Yingmo Jie, Charles Zhechao Liu, and Kim-Kwang Raymond Choo, *Senior Member, IEEE*

***Abstract*—Significant volume of information of a broad variety (or modalities, such as image, audio, video, and text) is sensed and collected [such as those by the Internet of Things (IoT) devices] regularly (e.g., hourly). Such information is then analyzed to inform decision making, such as clinical diagnosis and product recommendation. Data with different representations may have the same semantic information, and there have been considerable efforts devoted to designing efficient searching approaches on objects with different modalities. However, multimodal data carry sensitive information, and maintaining privacy is crucial in our privacy-aware and interconnected society. In this article, we combine both the collective matrix factorization (CMF) and homomorphic encryption (HE) to construct an efficient and accurate scheme to facilitate cross-modal retrieval, without the loss of any sensitive information. Our scheme identifies the unified feature vectors for every object in the training set with different modalities and obtains the mapping matrices for out-of-sample objects. After the encryption process, these matrices are stored on the remote cloud server (CS). Hence, the server can calculate the secure, unified features for any query. In this article, we also built a privacy-preserving index structure using locality-sensitive hashing (LSH), which provides both security and efficiency. Performance evaluations demonstrate the potential for our proposed scheme in the real-world IoT applications.**

***Index Terms*—Collective matrix factorization (CMF), homomorphic encryption (HE), locality-sensitive hashing (LSH), secure cross-modal retrieval (SCMR).**

Manuscript received October 4, 2019; revised November 23, 2019; accepted January 1, 2020. Date of publication January 8, 2020; date of current version April 14, 2020. This work was supported in part by the National Science Foundation of China under Grant 61501080, Grant 61871064, and Grant 61877007; in part by the Fundamental Research Funds for the Central Universities under Grant DUT19JC08; and in part by the Guangxi Key Laboratory of Trusted Software under Grant kx201903. The work of Kim-Kwang Raymond Choo was supported by the Cloud Technology Endowed Professorship and the National Science Foundation Centers of Research Excellence in Science and Technology (CREST) under Grant HRD-1736209. *(Corresponding author: Kim-Kwang Raymond Choo.)*

Cheng Guo, Jing Jia, and Yingmo Jie are with the School of Software Technology, Dalian University of Technology, Key Laboratory for Ubiquitous Network and Service Software of Liaoning Province, Dalian 116620, China, and also with the Guangxi Key Laboratory of Trusted Software, Guilin University of Electronic Technology, Guilin 541004, China (e-mail: guocheng@dlut.edu.cn; jiajing1995@163.com; jymsf2015@mail.dlut.edu.cn).

Charles Zhechao Liu and Kim-Kwang Raymond Choo are with the Department of Information Systems and Cyber Security, University of Texas at San Antonio, San Antonio, TX 78249 USA (e-mail: charles.liu@utsa.edu; raymond.choo@fulbrightmail.org).

Digital Object Identifier 10.1109/JIOT.2020.2964412

I. Introduction

INTERNET of Things (IoT) devices sense, collect, and transfer significant volume of data, and such data may exist in heterogeneous types (or modalities, such as image, audio, video, and text). Collectively, data from different sources can contain information of commercial and societal interests. Hence, there have been efforts to design cross-modal searches to facilitate the retrieval of heterogeneous data containing the same latent semantic meaning. For example, wearable smart healthy devices can monitor user physiological data, which can be used to inform medical diagnosis, and data collected by vision assistive devices can improve the quality of life for the visually impaired individual [1]–[4].

While a broad range and types of data collected by sensing devices can enrich the representation of things, the storage and computational capabilities of sensors are limited. The combination of IoT and cloud services can be utilized to process IoT data. However, data privacy is a potential concern. Therefore, in this article, we seek to determine how one can bridge two different modalities to facilitate searching for the same semantic information without affecting the privacy of the original data. Also, in this article, we focus on multimedia data.

A number of cross-modal retrieval methods have been developed in the literature, such as those designed for visual classification, searching of media, recognizing actions, and visual representations [5]–[8]. These methods are generally capable of filling the semantic gap among heterogeneous sources of data, but they are not designed to preserve user privacy. Thus, when sensitive, unencrypted data are uploaded to a search engine, privacy cannot be maintained, and an adversary may gain access to private information. However, after the original files are encrypted, the correlation between two similar files cannot be discerned. In other words, encryption complicates searching operations. Thus, we focus on achieving searchable, cross-modal encryption in this article.

The concept of searchable encryption (SE) was coined to facilitate searching and retrieval of encrypted contents that contained a concrete query keyword [9], and since then SE has been extended to support searching of multiple keywords, dynamic searching, ranked searching [10], [11], and other activities. For example, a number of SE methods have been proposed recently to provide secure image searching, such as label-based image searching and content-based image searching [12]–[14]. These methods were intended mainly

for single-modal rather than multimodal use. This is due to the semantic gap between heterogeneous modalities, which compounds the challenge in achieving secure cross-modal searching.

Existing cross-modal approaches in the plaintext domain are mainly based on subspace learning, which constructs a common subspace and transforms the data of different modalities to the same common subspace to compute the similarity [15]–[17]. However, the performance of these traditional approaches decreases as the volume of data increases. In other words, when processing significant volume of the data (becoming a norm in today's landscape), the time required, the cost, and the low accuracy of retrieval can be prohibitive.

A number of hashing-based methods have also been proposed in recent years [18]–[21]. For example, such hashing-based methods can embed data from different modalities into compact binary hash codes; consequently, reducing storage requirement and increasing search speed. More relevant results are also easier to obtain when the database contains (a large amount of) data in different modalities. In other words, the hash-based methods are viable for large-scale databases. However, these schemes simply combine the information from heterogeneous sources of an instance and ignore the correlations between different modalities.

More recently, deep-learning techniques have been utilized in cross-modal research, for example, to discover the latent semantic information among multiple modalities. Such deep-learning-based schemes can be practical in the plaintext domain. However, the training process and forward propagation phases usually involve multiple layers of network and complex operations, which are too prohibitive to be executed over encrypted data.

Collective matrix factorization (CMF) is an efficient approach to learn the semantic similarity and make relational prediction. It is more capable than hashing-based schemes in determining the implicit information among different modalities. In addition, CMF has more concise operations than deep-learning-based schemes. Thus, CMF can be used on encrypted data.

In this article, we explore the problem of constructing a secure, cross-modal searching method based on CMF to calculate a unified hash value for different modalities. We train the private data sets of heterogeneous data sets to obtain the permutation function, which can take the user's trapdoor as input and output the secure unified feature vector. Then, we utilize local sensitive hashing (LSH) to convert the unified feature vectors to irreversible hash values and locate the corresponding hash buckets to determine correlative candidates. Our main contributions are summarized as follows.

1) We design a practical, secure, cross-modal searching scheme, hereafter, referred to as secure cross-modal retrieval (SCMR). SCMR leverages the latent semantic correlation between heterogeneous data, without leaking any sensitive information.
2) This is the first attempt to introduce CMF to the encrypted domain to address the cross-modal problem. Our proposed approach protects the confidentiality of data sets and the index, unlike existing cross-modal retrieval methods (since they are not designed to do so).
3) Our scheme is efficient because only homomorphic addition and plain-text multiplication are required of the cloud server (CS), and our scheme avoids multiround interactions among entities.

In the next two sections, we will briefly review the related literature and introduce the problem formulations. In Section IV, we present our proposed approach. We then evaluate its performance and security in Section V. We also compare its performance with several plaintext methods. Then, we conclude this article in Section VI.

II. Related Literature

There are a number of challenges associated with preserving privacy during cross-modal searching. For example, how can we determine how to narrow down and ultimately eliminate the semantic gap between heterogeneous data sources? Also, how can we determine how to execute conventional searching operations over encrypted multimodal data?

In recent years, researchers have proposed a number of different schemes that facilitate searching of single modal data. The schemes can be broadly categorized into unsupervised schemes and supervised schemes. The former is less reliant on the number of instances in the data sets; thus, it is relatively easier to adapt to databases of varying volume and is practical for multiple scenarios. However, the performance of unsupervised schemes is dependent on the data distribution, and they have no resistance to malicious attacks. In addition, the semantic gaps between low-level features and high-level semantics can result in poor performance for unsupervised schemes.

Supervised schemes, on the other hand, reflect the semantic features into a common space and have better capabilities to dig out the potential relations of data. In [20], for example, Bronstein *et al.* introduced a cross-view hashing (CVH) model, designed to solve the generalized eigenvalue problem and minimize the multiview Euclidean distance between heterogeneous pairs of data. Another scheme proposed in [22], intermedia hashing (IMH) also reflects multiview data into a common hamming space to protect both intermedia similarity and intramedia similarity. Song *et al.* [18] proposed an approach called cross-modality similarity search hashing (CMSSH), which embeds heterogeneous data into a common subspace and trains the hash functions with using Eigen decomposition. These hashing-based schemes outperform other traditional methods because they speed up the process and have low time cost even for large-scale databases. In other words, the hashing-based schemes are more useful for real-world applications. However, such schemes ignore the latent correlations between different modalities.

The deep-learning techniques have also been used extensively to solve cross-modal searching problems [23]–[28]. They can extract the latent information among multiple modalities and learn the semantic feature of data points. However, it remains a challenging problem to determine how one can achieve the same performance in protecting consistency among heterogeneous encrypted data.

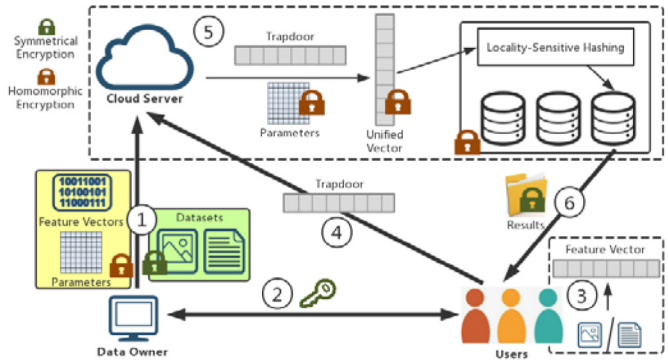


Fig. 1. System model.

Similarity searching protocols over simple-view encrypted data is another emerging research topic. The development of simple-view, secure, similarity searching, such as multikeyword searching, secure high-dimensional data searching, and secure image searching, has also contributed to advances relating to secure, cross-modal searching.

There have also been advances in approaches designed for secure, content-based searching of images. These schemes that extract features of images take into account the semantic gap between images and texts, which perfectly matches the aim of cross-modal retrieval. CMF was first proposed to predict unknown values for relational learning, which is an efficient approach for determining the semantic similarity between multimodalities. Compared with other supervised schemes, such as deep-learning and convolutional neural networks, CMF has a more concise training process with less computational complexity; thus, it is a viable candidate for applications involving encryption.

In this article, we also focus on protecting the consistency of multimodal data resources, without potentially leaking their sensitive information. Specifically, our proposed approach is one of the first to combine both CMF and homomorphic encryption (HE), in order to facilitate the cross-modal retrieval in the encrypted domain. Specifically, HE allows us to perform calculation of secure unified feature vectors and protect the confidentiality of data. In order to speed up the searching process, we construct a secure index using LSH. We will demonstrate that our scheme has more (powerful) capabilities than simple-modal schemes and provide better privacy protection than competing cross-modal schemes in the plaintext domain, later in this article.

In the next section, we will explain the system and security models, and relevant background materials.

III. Problem Formulation

A. System Model

In this article, we consider an SCMR scheme comprising the following three entity types—see also Fig. 1.

1) A data owner (DO) who holds private data of different modalities. To reduce the storage requirements and the computational complexity, as well as ensuring the security of the private data, the DO would like to encrypt the data sets and outsource them to the CS. Thus, the DO must first execute CMF using the original image-text pairs to obtain the parameters and unified feature vectors. Then, the DO encrypts all of these files and outsources them to the CS.
2) An honest-but-curious CS, which establishes a secure index using LSH. When a user submits a query, the CS computes the unified feature vector of the query using encrypted parameters. Then, the CS calculates the hash value in a privacy-preserving manner to locate the exact hash bucket before returning all corresponding results to the user.
3) Several users. When a user requests for relevant images or texts, (s)he must establish a connection with the DO to obtain a secret key to be used for decryption. Then, (s)he extracts the feature vector of an image or a text in the same manner as the DO and sends the query to the CS. When the user receives the encrypted results from the CS, (s)he could use the secret key to decrypt the encrypted results.

B. Security Model

In this article, we assume that the CS is honest, but curious. In other words, the CS will faithfully follow the proposed scheme and return the correct results to the users, and the CS will not collude with others to obtain the secret key. However, the CS may be sufficiently curious about the outsourced data and attempt to deduce information beyond the ciphertexts. Based on the limited data that are available (encrypted data sets, encrypted unified feature vectors, encrypted hash values, encrypted projection matrices, and users' queries), we conclude the attack model of an adversary is a ciphertext only attack (COA) model, in which the CS has the encrypted data sets, encrypted unified feature vectors, encrypted hash values, encrypted projection matrices, and user queries.

Under this attack model, our proposed scheme needs to provide semantic security for both DO and user. In particular, the following aspects of security should be ensured.

1) *File Privacy:* Since the original data contains sensitive information, the CS cannot learn their plaintext by simply analyzing the encrypted data.
2) *Parameter Privacy:* Since the parameters (i.e., projection matrices) reflect the correlation of original data and their unified feature, they cannot be deduced by the CS.
3) *Trapdoor and Index Privacy:* Since the query and index reflect the relations between the query and corresponding plaintext, the CS cannot deduce their content by simply analyzing the encrypted information.

C. Preliminaries

1) Collective Matrix Factorization: CMF, proposed by Singh and Gordon [29], can deeply mine the latent semantic relationship between heterogeneous entities to facilitate prediction and recommendation. CMF jointly factorizes multiple relations of different types and learns a common space that contains the semantic information. We use CMF to learn the unified feature vector of an image-text pair to address cross-modal retrieval.

2) Homomorphic Encryption: In order to compute the unified feature vector of a query, the DO encrypts the parameters using the Paillier cryptosystem [30], which supports homomorphic computation. The details are described below.

First, choose two large primes, p and q, and let $N = pq$. Also, let $Z_{N^2} = \{0, 1, \ldots, N^2 - 1\}$ and $Z^*_{N^2} \subset Z_{N^2}$ denote the set of non-negative integers that have multiplicative inverse modulo, N^2. Select $g \in Z^*_{N^2}$ which satisfies $gcd(L(g^\lambda \bmod N^2), N) = 1$, where $\lambda = \text{lcm}(p-1, q-1)$. Let λ denote the private key and (N, g) be the public key.

Given (N, g) and the plaintext denoted as $m \in Z_N (m < N)$, the ciphertext of m is computed as

$$c = E_{pk}(m, r) = g^m r^N \bmod N^2. \tag{1}$$

In the above formula, $r \in Z^*_N \subset \{0, 1, \ldots, N-1\}$ denotes the randomly chosen number that enables the Paillier cryptosystem to satisfy the semantic security.

To decrypt the ciphertext, c, we compute the following formula using the private key λ:

$$m = D_{sk}(c, \lambda) = \frac{L\left(c^\lambda \bmod N^2\right)}{L\left(g^\lambda \bmod N^2\right)} \bmod N. \tag{2}$$

In the above formula, $L(u) = (u-1)/N$.

The Paillier cryptosystem provides additive homomorphism because

$$\begin{aligned} c_1 \times c_2 &= E_{pk}(m_1, r_1) \times E_{pk}(m_2, r_2) \\ &= g^{(m_1+m_2)}(r_1 r_2)^N \bmod N^2. \end{aligned} \tag{3}$$

It also provides plaintext multiplication

$$D_{sk}([E_{pk}(m_1, r_1)]^{m_2} \bmod N^2) = (m_1 \times m_2) \bmod N. \tag{4}$$

3) Locality-Sensitive Hashing: LSH was proposed by Gionis *et al.* [31]. LSH can hash relevant objects to the same bucket with a very high probability, and dissimilar data points probably will be hashed to different buckets. LSH has been used for solving approximate nearest neighbor (ANN) problems. The definition of LSH is shown as follows.

Let S be the set of data objects and D be the distance, we have $B(q, r) = \{p : D(q, p) \geq r\}$, where q is a query object.

Definition 1: A function family $\mathcal{H} = \{h : S \to U\}$ is called (r_1, r_2, p_1, p_2) sensitive for D if for any q, p, $p' \in S$:

1) if $p \in B(q, r_1)$ then $Pr_{\mathcal{H}}[h(q) = h(p)] \geq p_1$;
2) if $p \notin B(q, r_2)$ then $Pr_{\mathcal{H}}[h(q) = h(p')] \leq p_2$.

If D is a dissimilarity measure, there must be $p_1 > p_2$ and $r_1 < r_2$. If D is a similarity measure, there must be $p_1 > p_2$ and $r_1 > r_2$.

IV. Proposed Scheme

In this section, we introduce our proposed SCMR method. First, we present SCMR in the image-text case, because it is easily understood.

A. Overview of SCMR

Suppose that $\mathcal{O} = \{o_i\}_{i=1}^n$ is the set of objects, and $X^{(1)} = [x_1^{(1)}, \ldots, x_n^{(1)}]$, and $X^{(2)} = [x_1^{(2)}, \ldots, x_n^{(2)}]$ are two different modalities of $\mathcal{O}$, where $x_i^{(1)} \in \mathbb{R}^{d_1}, x_i^{(2)} \in \mathbb{R}^{d_2}$ (usually $d_1 \neq d_2$). Given a query, our proposed scheme is supposed to compute the unified feature vector, f_i, for o_i, $i = 1, 2, \ldots, n$, and satisfy that f_i, f_j preserve the similarity between o_i, o_j with high probability.

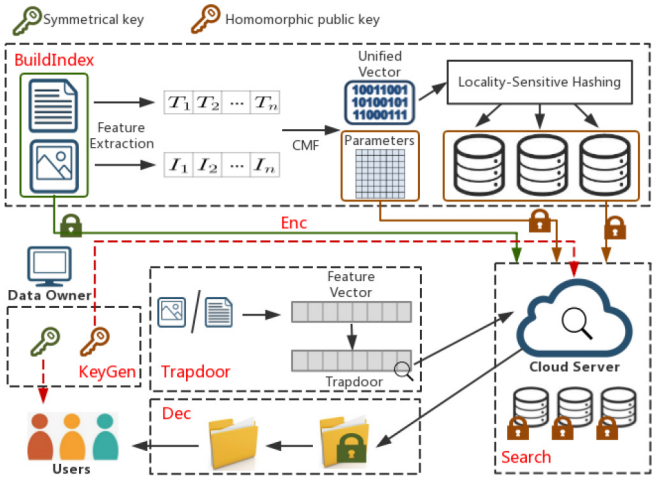


Fig. 2. Framework of the proposed scheme.

Our proposed SCMR method comprises the following six algorithms: *KeyGen*, *BuildIndex*, *Enc*, *Trapdoor*, *Search*, and *Dec*.

$(K, pk, sk) \leftarrow$ KeyGen (1^k): A security parameter k is chosen as the input, and this algorithm generates the symmetrical key, K, and the homomorphic key pair, (pk, sk).

$Index \leftarrow$ BuildIndex ($\mathcal{O}$): Given the original data sets, the DO executes this algorithm to determine the unified feature vectors of each object in $\mathcal{O}$ using CMF and computes the corresponding hash value as its index using LSH. Then, this algorithm outputs the index structure in plaintext form.

$(\mathcal{I}, \mathcal{P}) \leftarrow$ Enc ($Index$, $Param$, K, pk): Given the *Index*, training parameters *Param*, symmetrical key K, and homomorphic public key pk, the DO runs this algorithm to output the ciphertext, $\mathcal{P}$, of *Param* and the encrypted index structure, $\mathcal{I}$. Only the original data sets are encrypted by K, and the other data are encrypted by the Paillier cryptosystem.

$T_q \leftarrow$ Trapdoor ($Query$): Given a query image or a query text, the user runs this algorithm with the help of the DO to generate the search token (or trapdoor), T_q, and sends it to CS.

$\{c\} \leftarrow$ Search ($T_q, \mathcal{I}$): After receiving the search token, T_q, CS runs this algorithm to search for relevant encrypted files and returns them to the user. $\{c\}$ is the set of the ciphertext of the results.

$\{m\} \leftarrow$ Dec ($\{c\}, K$): This algorithm decrypts the ciphertext and obtains the original data set, $\{m\}$, with the help of K received from the DO.

As shown in Fig. 2, our SCMR scheme can be divided into two parts, namely, the offline setup phase and the online searching phase.

In the setup phase, the DO first uses the algorithm *KeyGen* to create the secret keys. Here, two different kinds of cryptosystems are used for encryption. More specifically, the DO uses symmetrical encryption for the original data sets because these data sets contain large amounts of files, and symmetrical encryption will reduce the computational complexity and

increase the efficiency. However, the parameters are encrypted by the Paillier cryptosystem because they are indispensable computational components in the following phases. Then, the DO executes *BuildIndex* to train the original data sets and assort these data to different hash buckets according to their hash values. The hashing structure is a very fast way of searching, and it is feasible to construct the index structure. Finally, all of the prepared data are encrypted by *Enc* and outsourced onto CS.

In the searching phase, the authorized user that has a search requirement runs *Trapdoor* to generate a search token. During this part, the DO sends information concerning how to extract the features and sends the symmetrical key, K, to the user, so that the user can process the query and generate the corresponding trapdoor. Once a trapdoor has been submitted, the CS carries out the *Search* algorithm. Particularly, the CS securely computes the unified feature vector of the query and obtains its hash value. Then, the CS locates the corresponding bucket and returns the candidates in the bucket to the user. Finally, the user runs *Dec* to decrypt the encrypted data using K given by the DO.

In addition, as the database increases in size, the DO could retrain the data to learn new projection matrices and update them with the CS, in order to provide more accurate search results.

B. Construction of SCMR

In this part, we introduce the details of the construction of our SCMR scheme.

1) $(K, pk, sk) \leftarrow$ KeyGen (1^k): In this article, the DO uses symmetrical encryption for the original data sets, such as DES and AES. As for encrypting the parameters, we introduce the details of the Paillier cryptosystem in Section III-B.

2) Index $\leftarrow$ BuildIndex $(\mathcal{O})$: First, we introduce the details of how to learn the unified feature vectors.

As mentioned in Section III-A, matrix factorization is a feasible approach to learn the latent semantic information of the original data using the following formula:

$$X^{(t)} = U_t V_t \ \forall t \tag{5}$$

where $U_t \in \mathbb{R}^{d_t \times k}$, $V_t \in \mathbb{R}^{k \times n}$, and k is the length of the latent semantic feature. More specifically, each column, v_i, of V is the latent semantic feature vector of the corresponding column, x_i, of X. As for multiple modalities of data, we assume that the similar heterogeneous objects should have the same latent semantic feature, based on which we jointly decompose $X^{(1)}, X^{(2)}$ with the constraint $V_1 = V_2 = V$

$$\lambda \left\| X^{(1)} - U_1 V \right\|_F^2 + (1-\lambda) \left\| X^{(2)} - U_2 V \right\|_F^2 \tag{6}$$

where λ is a balance parameter. This formula is only applicable for objects in $X^{(1)}$ and $X^{(2)}$; as for out-of-sample objects, a projection function is learned to transform an instance to the corresponding latent semantic feature

$$\mathcal{Y}_t(x^{(t)}) = P_t x^{(t)} + a_t \ \forall t \tag{7}$$

where $P_t \in \mathbb{R}^{k \times d_t}$ is the projection matrix and $a_t \in \mathbb{R}^k$ is the offset unit vector.

Algorithm 1 CMF

Input:
Data matrix $X^{(t)}, t = 1, 2$, parameters λ, μ, γ, k
Output:
Unified feature vectors V, projection matrices $P_t, t = 1, 2$
Initialize U_t, P_t by random matrices, $t = 1, 2$.
repeat
Fix U_t, P_t, update V by Formula (6), $t = 1, 2$;
Fix U_t, V update P_t by Formula (7), $t = 1, 2$;
Fix P_t, V update U_t by Formula (8), $t = 1, 2$;
until convergency.
return V, P_t, $t = 1, 2$

To summarize the loss mentioned above, the overall objective loss function consists of the CMF loss in (2), the projection loss in (3), and the regularization term

$$\underset{U_1, U_2, P_1, P_2, V}{\text{minimize}} \ \mathcal{L}(U_1, U_2, P_1, P_2, V) \tag{8}$$

where

$$\begin{aligned} \mathcal{L} = {} & \lambda \left\| X^{(1)} - U_1 V \right\|_F^2 + (1-\lambda) \left\| X^{(2)} - U_2 V \right\|_F^2 \\ & + \mu \left(\left\| V - P_1 X^{(1)} \right\|_F^2 + \left\| V - P_2 X^{(2)} \right\|_F^2 \right) \\ & + \gamma R(U_1, U_2, P_1, P_2, V) \end{aligned} \tag{9}$$

where μ and γ are tradeoff parameters, and $R(\cdot) = ||\cdot||_F^2$ is the regularization term to avoid overfitting.

The nonconvex optimization problem (8) becomes solvable only if we learn one matrix at a time with the other four variables fixed.

Fix P_t, V, let $(\partial G / \partial U_t) = 0$, $t = 1, 2$, then

$$U_t = X^{(t)} V^T \left(V V^T + \frac{\gamma}{\lambda_t} I \right)^{-1} \tag{10}$$

where $\lambda_1 = \lambda, \lambda_2 = 1 - \lambda$, I is the identity matrix.

Fix U_t, V, let $(\partial G / \partial P_t) = 0$, $t = 1, 2$, then

$$P_t = V_t {X^{(t)}}^T \left(X^{(t)} {X^{(t)}}^T + \frac{\gamma}{\mu} I \right)^{-1}. \tag{11}$$

Fix U_t, P_t, let $(\partial G / \partial V) = 0$, $t = 1, 2$, then

$$\begin{aligned} \mathcal{V} = {} & \left(\sum_{t=1}^{2} \lambda_t U_t^T U_t + (2\mu + \gamma) I \right)^{-1} \\ & \times \left(\sum_{t=1}^{2} \left(\lambda_t U_t^T + \mu P_t \right) X^{(t)} \right). \end{aligned} \tag{12}$$

Algorithm 1 shows the entire procedure.

After obtaining the unified feature vectors of each object, the DO must compute their hash values and construct the index structure.

We chose the normal Euclidean distance as the assessment criterion of the relevance between two objects, and its LSH formula is computed as follows:

$$\mathbb{H}(v) = \frac{|vR + b|}{a} \tag{13}$$

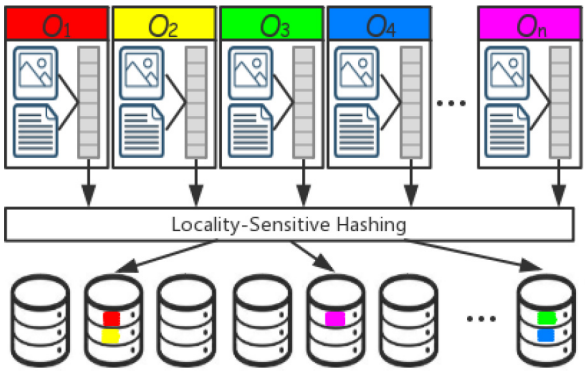


Fig. 3. Index structure.

Algorithm 2 Ciphertext Genaration

Input:
Index $= \{\mathcal{O}, \mathcal{F}, \mathcal{H}\}$, *Param* $= \{P_t\}(t = 1, 2), K, pk$
Output:
$\mathcal{I}, \mathcal{P}$
Initialize U_t, P_t by random matrices, $t = 1, 2$.
for each object $o_i \in \mathcal{O}$ **do**
Encrypt o_i with symmetrical key K;
end for
for each unified feature vector $f_i \in \mathcal{F}$ **do**
Encrypt f_i with public key pk;
end for
for each hash value $h_i \in \mathcal{H}$ **do**
Encrypt h_i with public key pk;
end for
for each projection matrix $P_t \in Param$ **do**
Generate a random invertible $d_t \times d_t$ matrix M_t;
Encrypt $P_t M_t$ with public key pk;
end for
return $\mathcal{I}, \mathcal{P}$

where R is a random vector, a is the size of each bucket, and $b \in [0, a]$ is a uniformly distributed random variable. Note that it is $(a/2, 2a, 1/2, 1/3)-$sensitive, and (R, a, b) is available to CS.

Fig. 3 shows the structure of the final index. Assume that o_1 and o_2 have the same hash value and that o_3 and o_4 have the same hash value. Different objects that have the same hash value will be assorted into the same hash bucket, which means there is a very high probability that they are similar.

3) $(\mathcal{I}, \mathcal{P}) \leftarrow$ *Enc (Index, Param, K, pk):* Fig. 3 shows that an index structure contains the original data set $\mathcal{O} = \{o_i\}_{i=1}^n$, the unified feature vectors $\mathcal{F} = \{f_i\}_{i=1}^n$, and the hash values $\mathcal{H} = \{h_i\}_{i=1}^n$. The training parameters of CMF consist of the projection matrices $P_t(t = 1, 2)$. The encryption algorithm is shown in Algorithm 2.

4) $T_q \leftarrow$ *Trapdoor (Query):* In this part, we assume that the user has been authorized and that the data communications are conducted over secure channels, which can be established using standard mechanisms such as SSL.

Under these circumstances, the DO tells the user how to extract the feature in the same way as the DO does, and the user obtains feature vector, v_t, of *Query*. Also, the DO sends the random invertible matrices, M_t, and the symmetrical key, K, to the user. Then, the user computes $T_q = M_t^{-1} v$ and submits T_q to CS.

5) $\{c\} \leftarrow$ *Search* $(T_q, \mathcal{I})$*:* Once search token T_q has been received, CS computes the secure unified feature vector of T_q using the additive homomorphism and plaintext multiplication properties of the Paillier cryptosystem mentioned in Section III-B.

First, we introduce Algorithm 3 to homomorphically compute the product of an encrypted matrix and a plaintext vector. Assume that W is an $m \times n$ matrix and that w_{ij} is the element located in the ith row and jth column. Similarly, $E(W)$ is the encrypted form of W, and $E(w_{ij})$ is the element located in the ith row and jth column. In addition, V is an $n \times 1$ vector, and v_i is the ith element of V.

Algorithm 3 can compute the product of $E(W)$ and V securely. Initially, it generates a random, n-dimensional vector, $R_{n\times 1}$, for the temporary storage of data. For $1 \le i \le n$, it computes the product of the ith row of $E(W)$ and V using the properties of the Paillier cryptosystem, including plaintext multiplication $(E(a)^b = E(ab))$ and additive homomorphism $(E(a) \times E(b) = E(a+b))$ and gives the result to the ith element of R. Finally, it outputs the multiplication result of $E(W)$ and V in encrypted form.

Algorithm 3 Homomorphic Multiplication

Input:
Encrypted matrix $E(W_{m\times n})$
Plaintext vector $V_{n\times 1}$
Output:
$E(WV)$
Generate a vector $R_{n\times 1}$
for $1 \le i \le m$ **do**
sum = E(0);
for $1 \le j \le n$ **do**
Compute temp $= E(wij)^v j$;
Compute sum = sum$\times$ temp;
end for
R_i = sum;
end for
return R

According to Algorithm 3, CS can securely compute the unified feature vector of the query

$$\begin{aligned} E_{pk}(f_q) &= E_{pk}(P_t M_t)^{T_q} \\ &= E_{pk}(P_t M_t)^{M_t^{-1} v} \\ &= E_{pk}\left(P_t M_t M_t^{-1} v\right) \\ &= E_{pk}(P_t v). \end{aligned} \tag{14}$$

Next, CS computes the secure hash value of the query. Similarly, CS can calculate the hash value as follows:

$$\begin{aligned} E_{pk}(h_q) &= \left(E_{pk}(f_q)^R \times E_{pk}(b)\right)^{1/a} \\ &= \left(E_{pk}(f_q R) \times E_{pk}(b)\right)^{1/a} \\ &= \left(E_{pk}(f_q R + b)\right)^{1/a} \\ &= E_{pk}\left(\frac{f_q R + b}{a}\right). \end{aligned} \tag{15}$$

Then, CS can locate the hash bucket according to the secure hash value of the query with the time complexity of $\mathcal{O}(1)$ and return the encrypted files {c} in the bucket to the user.

6) $\{m\} \leftarrow Dec\ (\{c\}, K)$*:* In this part, the user receives the encrypted candidates, $\{c\}$, from CS. These files were encrypted under symmetrical key, K, which was sent from the DO earlier. Therefore, the user only needs to decrypt the secure files to obtain the original image/text files.

V. Evaluation

In this section, we evaluate the performance of our proposed SCMR scheme in terms of security and searching accuracy.

A. Security Analysis

In this part, we discuss how our SCMR scheme resists the attacks from an honest-but-curious CS and protects the privacy under COA.

1) Parameter Privacy: Assume there have been s rounds of the retrieval of different queries, which means the CS has accumulated a set of trapdoors defined as $T_p = \{t_1, t_2, \ldots, t_s\}$ and their corresponding unified feature vectors $\{E_{pk}(F_q) = E_{pk}(f_1), E_{pk}(f_2), \ldots, E_{pk}(f_s)\}$. For ease of explanation, we define the secure parameter matrices, $E_{pk}(P_tM_t), t = 1, 2$, as $E_{pk}(X)$. Accordingly, the CS can formulate the following equations:

$$\begin{cases} \prod_{i=1}^{d} E_{pk}(x_{1i})^{t_{i1}} = E_{pk}(f_{i1}) \\ \prod_{i=1}^{d} E_{pk}(x_{1i})^{t_{i2}} = E_{pk}(f_{i2}) \\ \cdots \\ \prod_{i=1}^{d} E_{pk}(x_{1i})^{t_{is}} = E_{pk}(f_{is}). \end{cases} \tag{16}$$

Then, the CS computes the logarithm to the base g, where g is the public key used for HE. Thus, (16) can be converted as follows:

$$\begin{cases} \sum_{i=1}^{d} t_{i1} \log_g\big(E_{pk}(x_{1i})\big) = \log_g\big(E_{pk}(f_{i1})\big) \\ \sum_{i=1}^{d} t_{i2} \log_g\big(E_{pk}(x_{1i})\big) = \log_g\big(E_{pk}(f_{i2})\big) \\ \cdots \\ \sum_{i=1}^{d} t_{is} \log_g\big(E_{pk}(x_{1i})\big) = \log_g\big(E_{pk}(f_{is})\big). \end{cases} \tag{17}$$

Ideally, $E_{pk}(x_{1i}) = g^{x_{1i}} r^N \bmod N^2$. Thus, (17) can be computed as

$$\begin{cases} \sum_{i=1}^{d} t_{i1}x_{1i} + N \log_g r \cdot \sum_{i=1}^{d} t_{i1} = \log_g\big(E_{pk}(f_{i1})\big) \\ \sum_{i=1}^{d} t_{i2}x_{1i} + N \log_g r \cdot \sum_{i=1}^{d} t_{i2} = \log_g\big(E_{pk}(f_{i2})\big) \\ \cdots \\ \sum_{i=1}^{d} t_{is}x_{1i} + N \log_g r \cdot \sum_{i=1}^{d} t_{is} = \log_g\big(E_{pk}(f_{is})\big) \end{cases} \tag{18}$$

when $s > d$, there exists solution(s) for these linear equations; thus, the CS may calculate the plaintext of $E_{pk}(P_tM_t), t = 1, 2$ after several rounds of searching.

However, the probability of recovering a pseudo-random permutation is negligible. In other words, even if the CS knows the value of $P_tM_t, t = 1, 2$, (s)he cannot guess the value of $P_t, t = 1, 2$. Thus, we can say that our scheme protects the privacy of parameters.

2) File Privacy: In this article, the original data sets were encrypted under symmetrical cryptosystem, and they are independent from the following searching part. Only the authorized users have the secret key to decrypt them, which means that CS and illegal users have no access to these data. Thus, we can say that the privacy of the original files is well protected.

3) Trapdoor Privacy: In this article, the CS must compute the unified feature vector of each query using homomorphic addition and plain-text multiplication.

Let A be an adversary from a random oracle O. We define the advantage for A getting any additional information of the query image/text to be Adv_A^{PPIS}.

Lemma 1: $Adv_A^{\text{PPIS}} \leq Pr[E_1] + Pr[E_2] \leq \epsilon_1 + \epsilon_2$, where E_1 is the event that A recovers a pseudo-random permutation, and E_2 is the event that A recovers an image or a content only with its feature vector.

Proof: In the Trapdoor algorithm, the user extracts the feature vector of query and permutates it before submitting to the CS. Hence, the advantage Adv_A^{PPIS} for A is the addition of $\Pr[E_1]$ and $\Pr[E_2]$, at most.

Shoup [32] proved that the advantage for an adversary, A, breaking a pseudo-random permutation was negligible. For example, suppose that the length of the feature vector is d. If the adversary wants to guess the correct vector before permutation, the probability is $1/O(d!)$. In the polynomial time, it is impossible for the adversary to guess the original feature vector, which means that $Pr[E_1] \leq \epsilon_1$ is negligible.

Feature extracting is a unidirectional process and, obviously, the probability of recovering an image or a text only by its feature vector is negligible.

Therefore, the advantage $Adv_A^{PPIS} \leq Pr[E_1] + Pr[E_2] \leq \epsilon_1 + \epsilon_2$ for A to obtain any additional information from O is negligible. ■

4) Index Privacy: Let A be an adversary from a random oracle, O. We define the advantage for $A's$ getting any additional information from the index to be Adv_A^{PPIS}.

Lemma 2: $Adv_A^{PPIS} \leq Pr[E_3] + Pr[E_4] \leq \epsilon_3 + \epsilon_4$, where E_3 is the event that A recovers a hashing function and E_4 is the event that A distinguishes the differences between the two sequences, $E_{pk}(X)$ and $E_{pk}(Y)$, using the Paillier cryptosystem.

Proof: During the Secure algorithm, CS calculates the hash value of the query and locates the corresponding hash bucket to obtain correlative candidates in the homomorphic cryptosystem. Hence, the advantage Adv_A^{PPIS} for A is, at most, the addition of $\Pr[E_3]$ and $\Pr[E_4]$.

Hashing is a unidirectional process, in the polynomial time, and it is impossible for the adversary to recover an input by its hash value, which means that $Pr[E_3] \leq \epsilon_3$ is negligible.

If an event, E_2, occurs with a probability greater than ϵ_2, it means that A could construct a simulator that has the advantage greater than ϵ_2 to break the Paillier cryptosystem. As long as the Paillier cryptosystem provides semantic security, it is inconsistent with the fact. Therefore, the advantage for A breaking the Paillier cryptosystem is $Pr[E_4] \leq \epsilon_4$.

Finally, the advantage $Adv_A^{PPIS} \leq Pr[E_3] + Pr[E_4] \leq \epsilon_3 + \epsilon_4$ for A to obtain any information from O is negligible. ■

B. Experimental Performance

In this section, we evaluated our SCMR scheme by carrying out several experiments using different data sets. These evaluations were performed using an Intel Core i5-3230M CPU at 2.60 and 3.85 GB of memory. The aim was to verify the efficiency and accuracy of SCMR.

1) Data Sets:

Wikipedia [33]: It is a set of 2866 image–text pairs continually selected by Wikipedia's editors. In this collection, each image has a corresponding SIFT feature, and each text was split into several sections and represented by the 10 most popular categories among the 29 categories.

NUS-WIDE [34]: It is a collection of more than 260 000 real-world images. Each of these images has at least one tag among 5018 nonrepetitive tags provided by Flickr's users. It also provides various types of feature representations for different experimental requirements.

IAPR-TC12 [35]: This data set is comprised of 20 000 images collected from various scenarios. Each of them is pruned to the size of $256\times256\times3$ and has at least one text caption. Every text describing its data point was a 2912-D bag of words (BOW) feature vector.

2) Baseline Methods: In this article, we evaluated our proposed approach with five other state-of-art cross-modal methods, namely, CVH [20], IMH [22], CMSSH [18], DVSH [27], and DCMH [28]. CVH, IMH, and CMSSH are classical hashing-based methods that embed data of different modalities into a common subspace. DVSH and DCMH are deep-learning-based methods with high accuracy. We implemented these methods in the plaintext domain only for comparing the searching accuracy.

We chose the following two criteria to evaluate the retrieval performance: 1) the precision–recall curve and 2) the mean average precision (mAP). The average precision (AP) can be calculated as

$$AP = \frac{1}{L}\sum_{i=1}^{R} P(i) \times \delta(i). \tag{19}$$

In the above equation, L is the number of relevant data points that are recognized correctly, $P(i)$ is the precision of the top i searching results, and $\delta(i)$ is 1 if the ith result is relevant to the query, and it is 0 otherwise.

3) Results and Discussion: Table I lists the mAP values for our proposed SCMR scheme and the other five baseline protocols. Although DCMH has the highest mAP values, our proposed SCMR is almost comparable with it. SCMR focuses on achieving accurate cross-modal searching, and it also protects the DO and users from leaking sensitive information to unauthorized entities (as discussed in Section V-A).

The searching performance over multilabel data sets was better than the single-class data set, Wikipedia. This is reasonable because multiple labels provide much more semantic information. Another factor that influences the searching performance is the size of the hash codes, because longer hash codes decrease the probability of collision and provide more space for embedded information. However, the indiscriminative size of hash codes may become useless for improving the accuracy of searches because the low bits in the hash codes decrease as the size increases, ultimately resulting in the loss of semantic information.

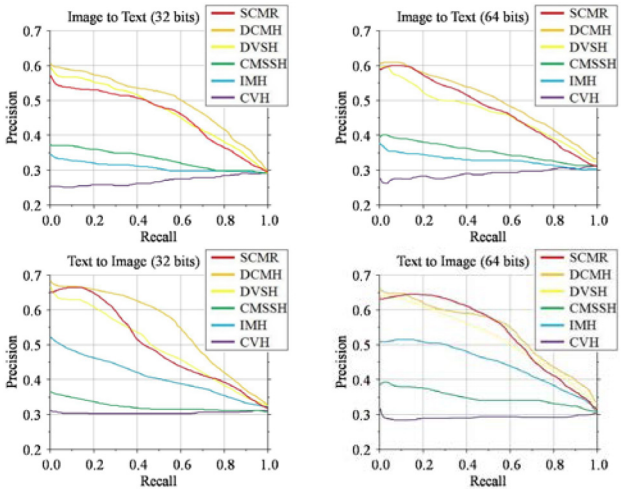


Fig. 4. PR-curves on wikipedia varying hash code length.

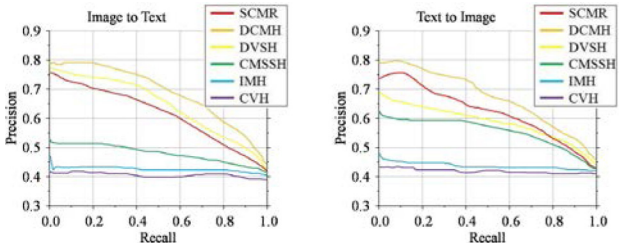


Fig. 5. PR-curves on NUS-WIDE.

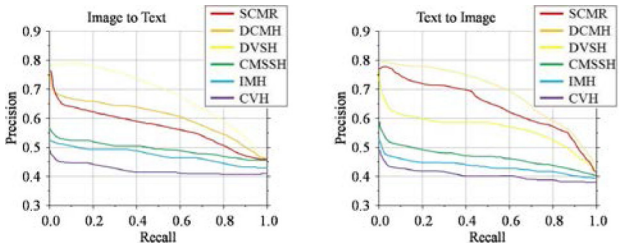


Fig. 6. PR-curves on IAPR-TC12.

Fig. 4 shows the PR-curves in the Wikipedia data set that vary the length of the hash codes. Fig. 5 shows the PR-curves on NUS-WIDE, and Fig. 6 shows the PR-curves on IAPR-TC12. It is apparent that our SCMR scheme performed comparably with these deep-learning, cross-modal schemes. In addition, searching images by texts is more accurate than the opposite way, because images have more abundant and abstractive semantic information than words.

The introduction of Paillier HE ensures the semantic security of our proposed SCMR scheme as demonstrated in Section V-A. However, the computation of query's unified feature vector and its hash value is executed in the encrypted domain and the time cost is another significant focus because it relates to user experience. Here, we mainly discussed the time cost of *Search* phase because it involves most of the homomorphic operations.

TABLE I
MAP COMPARISON TASKS

Tasks	Image to Text									Text to Image								
Datasets	Wikipedia			NUS-WIDE			IAPR-TC12			Wikipedia			NUS-WIDE			IAPR-TC12		
bits	16	32	64	16	32	64	16	32	64	16	32	64	16	32	64	16	32	64
CVH	0.203	0.164	0.134	0.372	0.363	0.406	0.352	0.343	0.398	0.296	0.195	0.134	0.404	0.399	0.448	0.398	0.362	0.441
IMH	0.207	0.212	0.203	0.472	0.475	0.467	0.454	0.465	0.449	0.358	0.366	0.387	0.473	0.481	0.474	0.431	0.452	0.465
CMSSH	0.202	0.211	0.201	0.497	0.519	0.514	0.487	0.503	0.501	0.293	0.273	0.276	0.505	0.513	0.501	0.499	0.503	0.512
DVSH	0.385	0.393	0.398	0.701	0.712	0.724	0.621	0.631	0.646	0.622	0.631	0.637	0.669	0.678	0.712	0.643	0.674	0.695
DCMH	**0.415**	**0.433**	**0.442**	**0.711**	0.723	**0.733**	**0.698**	**0.709**	**0.717**	**0.633**	**0.639**	**0.644**	**0.751**	**0.764**	**0.783**	**0.679**	**0.707**	**0.726**
SCMR	0.391	0.412	0.422	0.709	**0.724**	0.731	0.653	0.687	0.703	0.612	0.631	0.639	0.661	0.692	0.726	0.645	0.677	0.692

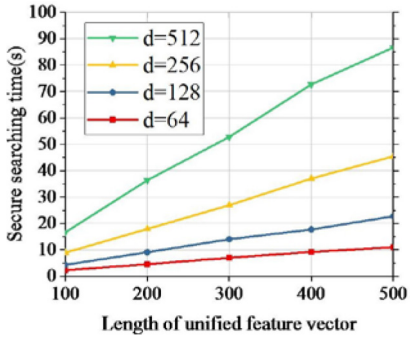


Fig. 7. Time cost of secure searching phase.

As described in Section IV-B, CS securely computes the query's unified feature vector according to Algorithm 3. It can be observed that the time cost of Algorithm 3 is affected by the size of input, and there are two main operations, namely, homomorphic addition and plaintext multiplication. Assume that the length of a query vector is d and the length of unified feature vector is k, there will be $k \times d$ times of both homomorphic addition and plaintext multiplication operations. In addition, the calculation of hash value involves two times of both homomorphic addition and plaintext multiplication.

Fig. 7 shows the time cost of different length of k and d. It can be observed that the time cost grows almost linearly with the increase of k. In this article, the scheme we proposed is a similar retrieval and not an exact retrieval. In other words, more candidates will be better than less. Thus, a shorter k will remove some inconsequential attributes and provide more results, and at the same time lead to reduced time cost. Even for a 512-D query vector, the time cost to compute the 500-D secure unified vector is less than 1.5 m, which is acceptable for a semantic secure scheme.

VI. CONCLUSION

In this article, we introduced an SCMR process to achieve efficient and accurate cross-modal searching in the encrypted domain. SCMR combines the semantic security of HE and the inherent characteristics of CMF for similar searching. Using LSH to build the structure of the index further increases the efficiency and security of SCMR. The experimental results on different kinds of data sets demonstrated that SCMR outperforms several state-of-the-art cross-modal searching schemes.

Future research includes deploying a prototype of SCMR, in collaboration with a real-world service provider to evaluate its scalability and real-world utility.

REFERENCES

[1] H. Tao, M. Z. A. Bhuiyan, A. N. Abdalla, M. M. Hassan, J. M. Zain, and T. Hayajneh, "Secured data collection with hardware-based ciphers for IoT-based healthcare," *IEEE Internet Things J.*, vol. 6, no. 1, pp. 410–420, Feb. 2019.

[2] G. Chu, N. Apthorpe, and N. Feamster, "Security and privacy analyses of Internet of Things children's toys," *IEEE Internet Things J.*, vol. 6, no. 1, pp. 978–985, Feb. 2019.

[3] B. Jiang, J. Yang, Z. Lv, and H. Song, "Wearable vision assistance system based on binocular sensors for visually impaired users," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 1375–1383, Apr. 2019.

[4] M. Li, L. Zhu, and X. Lin, "Efficient and privacy-preserving carpooling using blockchain-assisted vehicular fog computing," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4573–4584, Jun. 2019.

[5] Y. Luo, T. Liu, D. Tao, and C. Xu, "Multiview matrix completion for multilabel image classification," *IEEE Trans. Image Process.*, vol. 24, no. 8, pp. 2355–2368, Aug. 2015.

[6] M. Yu, L. Liu, and L. Shao, "Structure-preserving binary representations for RGB-D action recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 8, pp. 1651–1664, Aug. 2016.

[7] L. Shao, L. Liu, and M. Yu, "Kernelized multiview projection for robust action recognition," *Int. J. Comput. Vis.*, vol. 118, no. 2, pp. 115–129, Jun. 2016.

[8] C. Xu, T. Liu, D. Tao, and C. Xu, "Local rademacher complexity for multi-label learning," *IEEE Trans. Image Process.*, vol. 25, no. 3, pp. 1495–1507, Mar. 2016.

[9] D. X. Song, D. Wagner, and A. Perrig, "Practical techniques for searches on encrypted data," in *Proc. IEEE Symp. Security Privacy S&P*, Berkeley, CA, USA, 2000, pp. 44–55.

[10] G. Xu, H. Li, Y. Dai, K. Yang, and X. Lin, "Enabling efficient and geometric range query with access control over encrypted spatial data," *IEEE Trans. Inf. Forensics Security*, vol. 14, no. 4, pp. 870–885, Apr. 2019.

[11] B. Wang, S. Yu, W. Lou, and Y. T. Hou, "Privacy-preserving multi-keyword fuzzy search over encrypted data in the cloud," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, Toronto, ON, Canada, Apr. 2014, pp. 2112–2120.

[12] Y. Fan, X. Lin, G. Tan, Y. Zhang, W. Dong, and J. Lei, "One secure data integrity verification scheme for cloud storage," *Future Gener. Comput. Syst.*, vol. 96, pp. 376–385, Jul. 2019.

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

[14] K. Xu, Y. Guo, L. Guo, Y. Fang, and X. Li, "My privacy my decision: Control of photo sharing on online social networks," *IEEE Trans. Dependable Secure Comput.*, vol. 14, no. 2, pp. 199–210, Mar./Apr. 2017.

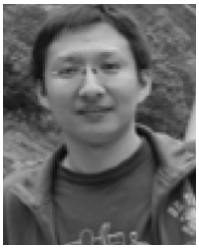
[15] D. Tao, L. Jin, Y. Yuan, and Y. Xue, "Ensemble manifold rank preserving for acceleration-based human activity recognition," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 6, pp. 1392–1404, Jun. 2016.

[16] D. Tao, J. Cheng, M. Song, and X. Lin, "Manifold ranking-based matrix factorization for saliency detection," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 6, pp. 1122–1134, Jun. 2016.

[17] D. Tao, Y. Guo, M. Song, Y. Li, Z. Yu, and Y. Y. Tang, "Person re-identification by dual-regularized KISS metric learning," *IEEE Trans. Image Process.*, vol. 25, no. 6, pp. 2726–2738, Jun. 2016.

[18] J. Song, Y. Yang, Y. Yang, Z. Huang, and H. T. Shen, "Inter-media hashing for large-scale retrieval from heterogeneous data sources," in *Proc. ACM SIGMOD Int. Conf. Manag. Data*, New York, NY, USA, 2013, pp. 785–796.

[19] G. Ding, Y. Guo, J. Zhou, and Y. Gao, "Large-scale cross-modality search via collective matrix factorization hashing," *IEEE Trans. Image Process.*, vol. 25, no. 11, pp. 5427–5440, Nov. 2016.
[20] M. M. Bronstein, A. M. Bronstein, F. Michel, and N. Paragios, "Data fusion through cross-modality metric learning using similarity-sensitive hashing," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, San Francisco, CA, USA, 2010, pp. 3594–3601.
[21] Z. Lin, G. Ding, M. Hu, and J. Wang, "Semantics-preserving hashing for cross-view retrieval," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Boston, MA, USA, 2015, pp. 3864–3872.
[22] S. Kumar and R. Udupa, "Learning hash functions for cross-view similarity search," in *Proc. 22nd Int. Joint Conf. Artif. Intell. (IJCAI)*, Barcelona, Spain, 2011, pp. 1360–1365.
[23] H. Lai, Y. Pan, Y. Liu, and S. Yan, "Simultaneous feature learning and hash coding with deep neural networks," in *Proc. CVPR*, Boston, MA, USA, 2015, pp. 3270–3278.
[24] K. Li, G.-J. Qi, J. Ye, and K. A. Hua, "Linear subspace ranking hashing for cross-modal retrieval," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 9, pp. 1825–1838, Sep. 2017.
[25] J. Lin, Z. Li, and J. Tang, "Discriminative deep hashing for scalable face image retrieval," in *Proc. 26th Int. Joint Conf. Artif. Intell. (IJCAI)*, Melbourne, VIC, Australia, 2017, pp. 2266–2272.
[26] H. Zhu, M. Long, J. Wang, and Y. Cao, "Deep hashing network for efficient similarity retrieval," in *Proc. 13th AAAI Conf. Artif. Intell. (AAAI)*, Phoenix, AZ, USA, 2016, pp. 2415–2421.
[27] Y. Cao, M. Long, J. Wang, Q. Yang, and P. S. Yu, "Deep visual-semantic hashing for cross-modal retrieval," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Disc. Data Min.*, San Francisco, CA, USA, 2016, pp. 1445–1454.
[28] Q.-Y. Jiang and W.-J. Li, "Deep cross-modal hashing," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Honolulu, HI, USA, 2017, pp. 3232–3240.
[29] A. P. Singh and G. J. Gordon, "Relational learning via collective matrix factorization," in *Proc. 14th ACM SIGKDD Int. Conf. Knowl. Disc. Data Min. (KDD)*, Las Vegas, NV, USA, 2008, pp. 650–658.
[30] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *Proc. Int. Conf. Theory Appl. Cryptograph. Tech.*, Prague, Czech Republic, 1999, pp. 223–238.
[31] A. Gionis, P. Indyk, and R. Motwani, "Similarity search in high dimensions via hashing," in *Proc. 25th Int. Conf. Very Large Data Bases (VLDB)*, Edinburgh, U.K., 1999, pp. 518–529.
[32] V. Shoup, "Sequences of games: A tool for taming complexity in security proofs," *IACR Cryptology ePrint Archive*, vol. 2004, p. 332, Nov. 2004.
[33] N. Rasiwasia *et al.*, "A new approach to cross-modal multimedia retrieval," in *Proc. 18th ACM Int. Conf. Multimedia (MM)*, Florence, Italy, 2010, pp. 251–260.
[34] T.-S. Chua, J. Tang, R. Hong, H. Li, Z. Luo, and Y. Zheng, "NUS-WIDE: A real-world Web image database from national university of Singapore," in *Proc. ACM Int. Conf. Image Video Retrieval (CIVR)*, 2009, p. 48.
[35] H. J. Escalante *et al.*, "The segmented and annotated IAPR TC-12 benchmark," *Comput. Vis. Image Understanding*, vol. 114, no. 4, pp. 419–428, Apr. 2010.



Cheng Guo (Member, IEEE) received the B.S. degree in computer science from the Xi'an University of Architecture and Technology, Xi'an, China, in 2002, and the M.S. degree and the Ph.D. degree in computer application and technology from the Dalian University of Technology, Dalian, China, in 2006 and 2009, respectively.

From July 2010 to July 2012, he was a Postdoctoral Fellow with the Department of Computer Science, National Tsing Hua University, Hsinchu, Taiwan. Since 2013, he has been an Associate Professor with the School of Software Technology, Dalian University of Technology. His current research interests include information security, cryptology, and cloud security.

Jing Jia received the B.S. degree in software engineering from the Dalian University of Technology, Dalian, China, in 2017, where she is currently pursuing the M.S. degree with the School of Software Technology.

Her research interests include image search and security, cloud storage and security, and cryptography.

Yingmo Jie received the B.S. degree in information and computing science from the Tianjin University of Technology and Education, Tianjin, China, in 2011, the M.S. degree in applied mathematics from the Civil Aviation University of China, Tianjin, in 2015, and the Ph.D. degree from the School of Mathematical Sciences, Dalian University of Technology, Dalian, China, in 2019.

Since 2019, she has been a Postdoctoral Fellow with the School of Software Technology, Dalian University of Technology. Her current research interests include information security, resources optimization, and game theory.

Charles Zhechao Liu received the Ph.D. degree in management information systems from the University of Pittsburgh, Pittsburgh, PA, USA.

He is currently an Associate Professor with the University of Texas at San Antonio, San Antonio, TX, USA. His research has been published in *Management Information Systems Quarterly*, *Information Systems Research*, the *Journal of Management Information Systems*, the *Communications of the ACM*, and the *Communications of the Association for Information Systems*. His current research interests include the economics of information systems and cybersecurity, mobile apps, and data analytics.

Dr. Liu is a recipient of the Net Institute Research Grant, the 2018 UTSA College of Business Dean's Distinguished Research Award, and the UTSA College of Business E. Lou Curry Teaching Excellence Award in 2019. He is an ICIS Doctoral Consortium Fellow.

Kim-Kwang Raymond Choo (Senior Member, IEEE) received the Ph.D. degree in information security from the Queensland University of Technology, Brisbane, QLD, Australia, in 2006.

He is currently a Cloud Technology Endowed Associate Professor with the University of Texas at San Antonio (UTSA), San Antonio, TX, USA.

Dr. Choo is the recipient of the 2019 IEEE Technical Committee on Scalable Computing Award for Excellence in Scalable Computing (Middle Career Researcher), the 2018 UTSA College of Business Col. Jean Piccione and Lt. Col. Philip Piccione Endowed Research Award for Tenured Faculty, the British Computer Society's 2019 Wilkes Award Runner-up, the 2019 *EURASIP Journal on Wireless Communications and Networking* Best Paper Award, the Korea Information Processing Society's *Journal of Information Processing Systems* Survey Paper Award (Gold) in 2019, the IEEE Blockchain 2019 Outstanding Paper Award, the International Conference on Information Security and Cryptology (Inscrypt 2019) Best Student Paper Award, the IEEE TrustCom 2018 Best Paper Award, the ESORICS 2015 Best Research Paper Award, the 2014 Highly Commended Award by the Australia New Zealand Policing Advisory Agency, the Fulbright Scholarship in 2009, the 2008 Australia Day Achievement Medallion, and the British Computer Society's Wilkes Award in 2008. In 2018, he was named Outstanding Associate Editor of 2018 for IEEE ACCESS, in 2016 he was named the Cybersecurity Educator of the Year—APAC (Cybersecurity Excellence Awards are produced in cooperation with the Information Security Community on LinkedIn), and in 2015 he and his team won the Digital Forensics Research Challenge organized by Germany's University of Erlangen-Nuremberg. He is a Fellow of the Australian Computer Society, and the Co-Chair of IEEE Multimedia Communications Technical Committee's Digital Rights Management for Multimedia Interest Group.