

SCRM: Secure and Controllable Similarity Retrieval in Multiuser Settings

Yingying Li¹, Member, IEEE, Feng Li, Gaopan Hou, Yu Guan¹, Zhiquan Liu¹, and Qi Xie¹

Abstract—Cloud computing has become an essential paradigm for facilitating large-scale and privacy-preserving encrypted image retrieval in Internet of Things (IoT) environments. However, existing encrypted image retrieval schemes face challenges in balancing retrieval efficiency and data security, which hinders their practical adoption. On one hand, retrieval-efficient schemes based on secure k-Nearest Neighbor (kNN) are prone to known-plaintext attacks; on the other hand, highly secure schemes based on homomorphic encryption often suffer from excessive computational and storage overhead. Furthermore, supporting multiuser environments and enforcing fine-grained access control over query users (QUs) are critical challenges in IoT-based retrieval systems. To tackle these issues, we propose a secure and controllable similarity retrieval scheme in multiuser settings (SCRM), which achieves a practical tradeoff between efficiency and security while enabling multiuser management. First, we design an efficient and privacy-preserving similarity computation method that is resilient against known-plaintext attacks. Second, we introduce a key conversion protocol that enables similarity retrieval in multiuser settings without requiring key sharing. Third, we integrate attribute-based encryption to enforce fine-grained access control and trace QUs who may leak decryption keys. A correctness analysis confirms that SCRM ensures accurate similarity retrieval while supporting access control. Furthermore, a formal security analysis demonstrates that SCRM effectively protects data privacy against known-plaintext attacks. Finally, extensive experiments on real-world image dataset validate the efficiency and effectiveness of SCRM.

Index Terms—Similarity retrieval, multiuser settings, access control, traceability.

Received 23 January 2025; revised 5 April 2025; accepted 24 April 2025. Date of publication 28 April 2025; date of current version 9 July 2025. This work was supported in part by the National Natural Science Foundation of China under Grant 62402155; in part by the Zhejiang Provincial Natural Science Foundation of China under Grant LQ24F020013; and in part by the Hangzhou Joint Fund of the Zhejiang Provincial Natural Science Foundation of China under Grant LHZSZ24F020002. (Corresponding authors: Gaopan Hou; Qi Xie.)

Yingying Li is with the School of Mathematics, Hangzhou Normal University, Hangzhou 311121, China (e-mail: lyylyingying@163.com).

Feng Li is with the School of Computing and Information Systems, Singapore Management University, Singapore 178902 (e-mail: fengli@smu.edu.sg).

Gaopan Hou and Yu Guan are with the Hangzhou Institute of Technology, Xidian University, Hangzhou 311231, Zhejiang, China (e-mail: adsen13@163.com; 23031212282@stu.xidian.edu.cn).

Zhiquan Liu is with the College of Cyber Security, Jinan University, Guangzhou 510632, China (e-mail: zqliu@vip.qq.com).

Qi Xie is with the Key Laboratory of Cryptography of Zhejiang Province, Hangzhou Normal University, Hangzhou 311121, China (e-mail: qixie68@126.com).

Digital Object Identifier 10.1109/JIOT.2025.3564935

I. INTRODUCTION

WITH the rise of cloud computing, Internet-of-Things (IoT) devices continuously generate vast amounts of data, which are uploaded to cloud servers (CSs) for storage and retrieval [1]. However, CSs can easily obtain private information from plaintext data, leading to privacy leakage. To mitigate this, encrypting data is an effective way to address data privacy and security. Various privacy-preserving retrieval schemes for encrypted data have been proposed, such as keyword search [2], [3], [4], spatial data search [5], [6], [7], and image similarity retrieval [8], [9], [10]. Image similarity retrieval aims to identify images in a dataset that resemble a given query image, with applications in medical diagnosis [10], biometrics [11], and social networks [12].

With the development of ciphertext computing technologies, such as secure k-Nearest Neighbor (kNN) [13] and homomorphic encryption, many privacy-preserving encrypted image retrieval schemes have been proposed [10], [14], [15], [16], [17], [18], [19], [20], [21], [22], [23], [24], [25], [26], [27], [28], [29]. They utilize ciphertext computing technologies to compute similarity over encrypted feature vectors. The secure kNN technology encrypts feature vectors using random matrices, offering the advantage of low computational overhead. However, it has been shown to be vulnerable to ciphertext-only attacks [30]. In contrast, homomorphic encryption provides strong security, such as resistance to chosen-plaintext attacks, but its high computational cost hinders practical deployment. To improve retrieval efficiency, extracting global features or constructing sublinear indexes can effectively reduce the number of similarity computations. However, the resulting decline in retrieval accuracy becomes another critical concern. Therefore, developing a similarity computation method that balances security and efficiency remains a critical challenge.

Additionally, with an ever-increasing demand from users, similarity retrieval should adapt to more complex and practical scenarios, such as multiuser settings, flexible access control, and user traceability. In a multiuser setting, query users (QUs) should not share encryption keys, requiring the CS to transform ciphertexts encrypted with different keys into a form that is encrypted with the same key. Otherwise, the data owner (DO) would need to generate a separate searchable index for each QU, leading to inefficiency. Furthermore, user management in multiuser settings requires a more comprehensive approach. While polynomial-based access control mechanisms can be integrated with the secure kNN technology to enable fine-grained access control, their computational overhead

increases significantly as the number of roles grows [8], [14], [25], [26], [27]. Existing schemes employ Ciphertext-Policy Attribute-Based Encryption (CP-ABE) to achieve flexible access control, allowing only users who satisfy the specified access policy to decrypt them [22], [28], [29]. However, these schemes lack the capability to trace authorized users who leak their keys. Therefore, achieving controllable similarity retrieval in multiuser settings poses a significant challenge.

A. Motivation Factors

The following shortcomings of existing encrypted image retrieval schemes motivate us to construct a novel scheme.

- 1) Schemes utilizing secure kNN technology can at most resist known-background attacks and still leak the similarity between the query trapdoor and the index.
- 2) Although multikey homomorphic encryption enables multiuser settings and resists chosen-plaintext attacks, it incurs significant computational overhead.
- 3) The polynomial-based access control mechanism is limited by the total number of roles and lacks the capability to trace users who leak their keys.

B. Contributions

To address these limitations, we propose a secure and controllable similarity retrieval in multiuser settings (SCRM). Specifically, SCRM designs a similarity computation method and a two-layer index to enable efficient similarity retrieval while ensuring resistance against known-plaintext attacks. Additionally, SCRM integrates a key conversion mechanism and CP-ABE to provide flexible access control and user traceability. Compared with existing schemes, SCRM offers enhanced security and practicality without compromising efficiency. The main contributions of SCRM are summarized as follows.

- 1) *Enhanced Security*: SCRM performs similarity retrieval by directly computing similarity comparison results, avoiding similarity leakage. Moreover, it achieves higher security than existing secure kNN-based schemes and effectively resists known-plaintext attacks.
- 2) *Support for Multiuser Settings With Nonshared Keys*: SCRM ensures that encryption keys are not shared among QUs. The CS transforms query trapdoors encrypted with different keys into a unified encryption form, enabling secure similarity retrieval with the encrypted index.
- 3) *Access Control and User Traceability*: SCRM enforces attribute-based access control, ensuring that only QUs whose attributes satisfy the predefined access policy can obtain the decryption key. Furthermore, if an authorized QU distributes their decryption key to unauthorized users, SCRM can successfully track and identify the source of the key leakage.

The remainder of this article is organized as follows. In Section II, we introduce some related work. Then, we describe some preliminaries in Section III and introduce our system model, threat model, and design goals in Section IV. Later, we present our scheme in Section V, followed by theoretical

TABLE I
RETRIEVAL SCHEMES: A COMPARATIVE SUMMARY

Schemes	$\mathbb{F}_1$	$\mathbb{F}_2$	$\mathbb{F}_3$	$\mathbb{F}_4$	$\mathbb{F}_5$	$\mathbb{F}_6$
[8]	Tree	Secure kNN	Polynomial	KBA	✗	✗
[9]	LSH	Secure kNN	✗	KBA	✗	✗
[10]	Tree	Secure kNN	✗	KBA	✗	✗
[14]	Linear	Secure kNN	Polynomial	KBA	✓	✗
[15]	Tree	Secure kNN	CP-ABE	KBA	✓	✗
[16]	Linear	Secure kNN	CP-ABE	KBA	✓	✓
[17]	Linear	Secure kNN	✗	KBA	✓	✗
[18]	Linear	Secure kNN	✗	KBA	✓	✗
[19]	Linear	Secure kNN	✗	KBA	✓	✗
[20]	Linear	HE	✗	CPA	✗	✗
[21]	Linear	HE	✗	CPA	✓	✗
[22]	Linear	HE	CP-ABE	CPA	✓	✗
[23]	Linear	HE	✗	CPA	✗	✗
[24]	Bitmap	HE	✗	CPA	✗	✗
[25]	Linear	Secure kNN	Polynomial	KBA	✓	✓
[26]	Linear	Secure kNN	Polynomial	KBA	✓	✗
[27]	Linear	Secure kNN	Polynomial	KBA	✗	✗
[28]	Linear	HE	CP-ABE	KBA	✗	✗
[29]	BF	Hash	CP-ABE	CPA	✗	✗
SCRM	Tree	Matrix	CP-ABE	KPA	✓	✓

Notes.

- “ $\mathbb{F}_1$ ”: Index structure; “ $\mathbb{F}_2$ ”: Feature encryption method; “ $\mathbb{F}_3$ ”: Access control method; “ $\mathbb{F}_4$ ”: Secure against attacks; “ $\mathbb{F}_5$ ”: Multi-user settings; “ $\mathbb{F}_6$ ”: Traceable.
- “KBA”: Known-background attack; “KPA”: Known-plaintext attack; “CPA”: Chosen-plaintext attack.
- “✓”: the scheme satisfies the objective; “✗”: the scheme does not satisfy the objective.

analysis and performance evaluation in Sections VI and VII, respectively. Finally, we conclude our work in Section VIII.

II. RELATED WORK

In this section, we will review some related work in terms of feature encryption methods and access control mechanisms. Table I presents a comparative summary of previous encrypted image retrieval works and our scheme.

The mainstream feature encryption technologies are secure kNN and homomorphic encryption. Yuan et al. [8] proposed a lightweight secure image retrieval scheme based on secure kNN, so that the computational overhead on the ciphertext has the same complexity as that on the plaintext. Xia et al. [9] constructed multiple hash tables based on locality-sensitive hashing (LSH) and used the secure kNN technology to protect the feature vectors stored in the hash tables. However, this scheme extracted MPEG-7 features to compute similarity, resulting in low retrieval accuracy. Schemes [10], [14], [15] extracted image features with convolutional neural networks (CNN) and encrypted the features with secure kNN, which improved the retrieval accuracy. With the learning with errors (LWE), Wang et al. [16] designed a privacy-preserving image retrieval framework in multiowner multiuser settings, which can resist known-background attack. Similarly, based on the

enhanced secure kNN technology, schemes [17], [18] implemented privacy-preserving encrypted image retrieval under known-background attack model. In [19], secure kNN and differential privacy are combined to ensure the confidentiality of index and query trapdoor.

Based on the Paillier encryption algorithm, Guo et al. [20] proposed a secure and efficient scheme to find the exact nearest neighbor over encrypted medical images. Yang et al. [21] implemented a multiuser encrypted image retrieval scheme based on the double trapdoor decryption algorithm and the Paillier encryption algorithm. Based on a multikey homomorphic encryption algorithm, Zhang et al. [22] designed a privacy-preserving image retrieval system, in which a CS and a trusted proxy server jointly complete the retrieval. Chen et al. [23] utilized the fully homomorphic encryption and private information retrieval (PIR) algorithms to design a content-based image retrieval scheme in the cloud-assisted IoT. By using an additional homomorphic bitmap index, Li et al. [24] designed a verifiable encrypted image retrieval incorporating reversible data hiding. However, the efficiency of schemes based on homomorphic encryption is significantly lower than that of schemes based on secure kNN technology.

In addition to making ciphertext searchable, some encrypted image retrieval schemes also implement access control. Yuan et al. [8] proposed a lightweight access control mechanism based on polynomial. Specifically, they constructed a polynomial with attributes as roots for each image. When user's attributes make the value of the polynomial zero, it means that the user meets the image's access control. Later, schemes [25], [26], [27] implemented polynomial-based access control in their images retrieval schemes. With the CP-ABE technology, schemes [15], [16], [22], [28] encrypt the encryption key of the image. When the attributes of the QU satisfy the access structure, the encryption key of the image can be decrypted. In [29], Zhang et al. also employed CP-ABE to achieve access control and constructed a bloom filter (BF) index to accelerate the retrieval process. In addition to its use in image retrieval for implementing access control, CP-ABE is more commonly employed in keyword search [31], [32], [33], [34], where it offers comprehensive support for access control, traceability, and revocation.

III. PRELIMINARIES

This section reviews related background knowledge, such as access structure, linear secret sharing schemes (LSSS), and Bilinear mapping.

A. Definitions

Definition 1 (Access Structure [35]): Let U denote the attribute universe. A collection $\mathbb{A} \in 2^U$ of nonempty sets of attributes is an access structure on U . The sets in $\mathbb{A}$ are called the authorized sets, and the sets not in $\mathbb{A}$ are called the unauthorized sets. A collection $\mathbb{A} \in 2^U$ is called monotone if $\forall B, C \in \mathbb{A}$: if $B \in \mathbb{A}$ and $B \subseteq C$, then $C \in \mathbb{A}$.

Definition 2 (LSSS [35], [36]): Let U denote the attribute universe and p denote a prime. A secret-sharing scheme Π

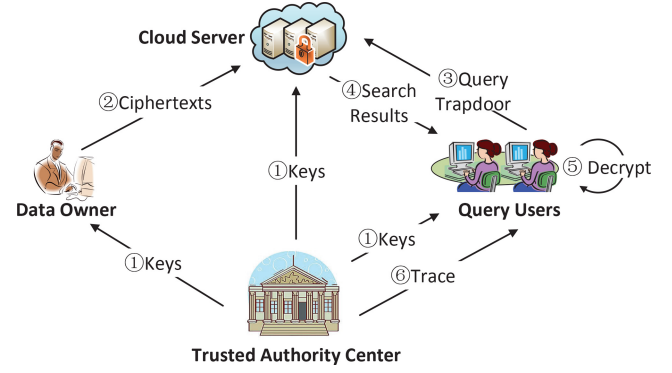


Fig. 1. System model of SCRM.

with domain of secrets $\mathbb{Z}_p$ realizing access structure on U is called linear if

- 1) The shares of a secrets $s \in \mathbb{Z}_p$ for each attribute form a vector over $\mathbb{Z}_p$.
- 2) For each access structure $\mathbb{A}$ on U , there exists a share-generating matrix $\tilde{M}$ with l rows and γ columns and a function ρ labels row i of $\tilde{M}$ with attribute $\rho(i)$ from the attribute universe U , i.e., $\rho : [l] \rightarrow U$. When we consider the column vector $y = (s, y_2, \dots, y_n)^\top$, where $y_2, \dots, y_n \in \mathbb{Z}_p$ are random numbers. Then $\tilde{M}y \in \mathbb{Z}_p^{l \times 1}$ is the vector of l shares of the secret s according to Π . The share $(\tilde{M}y)_i$ "belongs" to attribute $\rho(i)$, where $i \in [l]$.

B. Bilinear Mapping

Let $\mathbb{G}$ and $\mathbb{G}_T$ be two multiplicative cyclic groups of prime order p . Furthermore, g is a generator of $\mathbb{G}$. Define the mapping relation $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$, while the following three properties should be satisfied.

- 1) **Bilinearity:** $\forall g_0, g_1 \in \mathbb{G}$ and $\forall a, b \in \mathbb{Z}_p$, $e(g_0^a, g_1^b) = e(g_0, g_1)^{ab}$.
- 2) **Nondegeneracy:** $\exists g_0, g_1 \in \mathbb{G}$ such that $e(g_0, g_1) \neq 1$.
- 3) **Computability:** $\forall g_0, g_1 \in \mathbb{G}$, there is an efficient algorithm to compute $e(g_0, g_1)$.

IV. MODELS AND DESIGN GOALS

In this section, we describe the system model, threat model, and design goals of SCRM.

A. System Model

We consider a typical single cloud-assisted similarity retrieval in multiuser settings, which consists of four types of entities: DO, QU, CS, and trusted authority center (TA), as shown in Fig. 1.

- 1) TA is responsible for initializing the system, generating, distributing, and managing system parameters and keys. TA can track down the malicious user who leaked the key when a suspicious key appears.
- 2) DO generates an encrypted index and encrypts the plaintext images into ciphertext images. After that, DO

encrypts the image encryption key and finally uploads the ciphertext of the index, images, and key to CS.

- 3) QU generates a query trapdoor based on the query image and sends it to CS. After receiving the retrieval result from CS, QU decrypts the ciphertext of key and then obtains the plaintext image using the key.
- 4) CS is responsible for storing the data sent by DO and providing retrieval services for QU. When receiving a query trapdoor from QU, CS is responsible for converting the query trapdoor, retrieving the encrypted index and images to obtain the ciphertext image that meets the query trapdoor, and finally returning it to QU together with the ciphertext of the key.

As shown in Fig. 1, TA first generates system parameters and keys, and then distributes them to DO, QU, and CS, respectively (step ①). Then, DO builds an encrypted index, encrypts the image with a symmetric encryption algorithm, and encrypts the symmetric key with CP-ABE. The ciphertexts of index, images, and key are sent to CS (step ②). Next, QU generates a query trapdoor with their key and sends it to CS (step ③). When CS receives the query trapdoor, it first performs key conversion on the query trapdoor, then retrieves the encrypted index to obtain the image identifier that meets the query requirements. After that, CS retrieves the encrypted images according to the identifier. Finally, CS returns the image ciphertext and key ciphertext to QU as the results (step ④). After QU obtains the results, he/she decrypts the ciphertexts of key and image to get the plaintext of image (step ⑤). If a malicious QU leaks the key to other users, TA uses this key to obtain the QU's identifier, thereby tracking the malicious QU (step ⑥).

B. Threat Model

In our scheme, TA and DO are considered to be fully *trusted* because they are the initiators of the system. For QU, we assume that he/she may become *malicious* for profit, such as leaking the decryption key to other users. As for CS, it is considered to be *honest-but-curious*. That is, it will honestly store DO's data and provide retrieval services for QU. However, it will also be curious about some private information, such as the plaintext of index, query trapdoor, and retrieval results.

C. Design Goals

Our design goals are to propose a secure and controllable similarity retrieval scheme in multiuser settings. Specifically, SCRM should satisfy the following requirements.

- 1) *Security*: SCRM should ensure the confidentiality of DO's data, query trapdoor, and retrieval results.
- 2) *Controllability*: SCRM should be able to control different users in the multiuser setting. Specifically, SCRM should allow authorized users to access data while preventing unauthorized users from accessing data. Moreover, SCRM should be able to track authorized users who leak decryption keys to other users.
- 3) *Efficiency*: Under the premise of ensuring data privacy and supporting access control and traceability, SCRM

TABLE II
NOTATIONS

Notations	Descriptions
$[k]$	$1, 2, \dots, k$
PP, MSK	The public parameters and master key
S	The attribute set
M, CM	The plaintext and ciphertext of image set
n	The number of images in the image set
d	The dimension of feature vector
f_i, CF_i	The feature vector plaintext and ciphertext of IMG_i
k_{img}	The encryption key of image
SK_{DO}	The feature vector encryption key of DO
SK_{id}	The feature vector encryption key of QU_{id}
$SK_{id,CS}$	The conversion key of QU_{id}
$sk_{id,S}$	The decryption key of QU_{id}
CI	The encrypted index
CT	The ciphertext of k_{img}
CQ_{id}	The query trapdoor generated by QU_{id}
R_q	The identifier of image that matches the query trapdoor
IMG	The plaintext of image that matches the query trapdoor
$CIMG$	The ciphertext of image that matches the query trapdoor

should reduce overhead as much as possible and improve retrieval efficiency.

V. OUR PROPOSED SCHEME

We propose a secure and controllable similarity retrieval scheme over ciphertext images, which can efficiently retrieve the image most similar to the query image while resisting the known-plaintext attack. In addition, key conversion and CP-ABE introduced in multiuser settings enable nonshared keys, flexible attributes, and malicious user traceability. Table II gives some important notations used in SCRM.

A. Main Idea of SCRM

Similarity Retrieval: Given two feature vectors f_i, f_j of $IMG_i, IMG_j \in IMG$, and a feature vector q of IMG_q , similarity retrieval is to determine whether IMG_i is more similar than IMG_j for IMG_q . In SCRM, Euclidean distance $d(f_i, q)$ is used to measure the similarity of IMG_i and IMG_q . To determine whether $d(f_i, q) \leq d(f_j, q)$ or not, we compute $d(f_i, q) - d(f_j, q)$ as a whole in SCRM. First, we extend f_i to $(f_i, -(1/2)\|f_i\|^2)^\top$ and q to $(q, 1)$. Then, $d(f_i, q) - d(f_j, q) \leq 0$ means

$$\text{Sim} = \begin{bmatrix} f_i \\ -\frac{1}{2}\|f_i\|^2 \end{bmatrix} \cdot [q \ 1] - \begin{bmatrix} f_j \\ -\frac{1}{2}\|f_j\|^2 \end{bmatrix} \cdot [q \ 1] \geq 0. \quad (1)$$

Further, we extend $(q, 1)$ to

$$\begin{bmatrix} \text{diag}(q, 1) & \mathbf{0} \\ \mathbf{0} & \text{diag}(q, 1) \end{bmatrix}$$

where $\text{diag}(q, 1)$ is a $(d+1) \times (d+1)$ -dimensional diagonal matrix with $(q, 1)$ as the main diagonal elements, and $\mathbf{0}$ is a $(d+1) \times (d+1)$ -dimensional zero matrix. (1) can be expressed as

$$\text{Sim} = \begin{bmatrix} f_i & -\frac{1}{2}\|f_i\|^2 & -\mathbf{1}_{d+1} \end{bmatrix} \cdot \begin{bmatrix} \text{diag}(q, 1) & \mathbf{0} \\ \mathbf{0} & \text{diag}(q, 1) \end{bmatrix} \cdot \begin{bmatrix} \mathbf{1}_{d+1}^\top \\ f_j^\top \\ -\frac{1}{2}\|f_j\|^2 \end{bmatrix} \geq 0 \quad (2)$$

where $\mathbf{1}_{d+1}$ is a $d+1$ -dimensional vector and all elements are 1. As an example, assume that $d = 2$, $f_1 = (4, 2)$, $f_2 = (5, 7)$

and $q = (6, 3)$ are feature vectors of IMG_1 , IMG_2 and IMG_q , respectively. We compute

$$\text{Sim} = \begin{bmatrix} 4 & 2 & -10 & -1 & -1 & -1 \end{bmatrix} \cdot \begin{bmatrix} \text{diag}(6,3,1) & \mathbf{0} \\ \mathbf{0} & \text{diag}(6,3,1) \end{bmatrix} \cdot \begin{bmatrix} 1 & 1 & 1 & 5 & 7 & -37 \end{bmatrix} \\ = 6 > 0.$$

Therefore, $d(f_1, q) - d(f_2, q) < 0$, that is IMG_1 is more similar with IMG_q than IMG_2 . In fact, $d(f_1, q) = \sqrt{5}$, $d(f_2, q) = \sqrt{17}$, $d(f_1, q) < d(f_2, q)$, which denote (2) is correct.

To protect the vectors in (2), we add random numbers to them. After simplification, (2) can be regarded as

$$r_1 \cdot \text{Sim} + r_2 > 0 \quad (3)$$

where $r_1 > r_2 > 0$, and $d(f_i, q) - d(f_j, q) \leq 0$ iff $r_1 \cdot \text{Sim} + r_2 > 0$. In SCRM, we add sufficient random numbers to vectors for higher security. In addition, we perform the same operation on vectors as in secure kNN, randomly splitting them into two parts and multiplying them by random matrices. After these randomization operations, the final result is still as shown in (3). We will prove the correctness of similarity computation in Section VI-A1.

Multiuser Settings: A key conversion mechanism is introduced to convert query trapdoors encrypted with different keys into the form encrypted with the same key, which can be effectively retrieved over the same index. Specifically, SCRM randomly divides the key M used to generate the query trapdoor in secure kNN into M_{id} and $M_{id,CS}$, where $M = M_{id} \cdot M_{id,CS}$, M_{id} is sent to QU_{id} and $M_{id,CS}$ is sent to CS. Although each QU generates different query trapdoors with different M_{id} , CS multiplies the query trapdoor with the corresponding $M_{id,CS}$ to become a query trapdoor encrypted by M . This key conversion mechanism meets the requirement of nonshared keys in multiuser settings.

Access Control and User traceability: SCRM applies CP-ABE to multiuser settings to achieve effective user management. First, SCRM uses CP-ABE to encrypt the image encryption key, allowing QUs whose attributes satisfy the access control policy to obtain the image decryption key. Moreover, the size of the attribute set in SCRM is unbounded and not polynomially bounded, and the size of the public parameters does not grow linearly with the number of attributes. Besides, to prevent QUs from maliciously distributing decryption keys, SCRM uses traceability of CP-ABE to identify who exactly leaked the decryption key.

B. Constructions of SCRM

1) **Setup**(1^λ) $\rightarrow$ (PP, MSK): Given the security parameter λ , TA generates the public parameters PP and a master key MSK for the system.

First, TA generates a bilinear mapping $e : \mathbb{G} \times \mathbb{G}_T$, where $\mathbb{G}$ and $\mathbb{G}_T$ are two cyclic groups of order p . Then, TA randomly selects $\alpha, a \in \mathbb{Z}_p$ and $g_1, g_2, g_3, g_4, g_5 \in \mathbb{G}$. Next, TA chooses a probabilistic encryption scheme (Enc, Dec) and k'_1, k'_2 as two secret keys. Furthermore, TA generates a polynomial $f(x)$ and $t' - 1$ points $(x_1, y_1), (x_2, y_2), \dots, (x_{t'-1}, y_{t'-1})$ for the

Shamir's (t', n') threshold scheme. Finally, the PP and MSK are generated as

$$PP = \{g_1, g_2, g_3, g_4, g_5, g_1^\alpha, e(g_1, g_1)^\alpha\} \\ MSK = \{\alpha, a, k'_1, k'_2\}.$$

2) **KeyGen**(S, PP, MSK, d) $\rightarrow$ ($k_{img}, SK_{DO}, SK_{id}, SK_{id,CS}, sk_{id,S}$): Given the attribute set $S = \{A_1, A_2, \dots, A_k\} \subseteq \mathbb{Z}_p$, the public parameter PP , the master key MSK , and the dimension d of feature vector, TA generates the encryption key k_{img} of images, the encryption key SK_{DO} of feature vector, the encryption key SK_{id} and the conversion key $SK_{id,CS}$ of query vector, and the decryption key $sk_{id,S}$ of $QU_{id,S}$.

First, TA selects a key $k_{img} \in \mathbb{G}_T$ of symmetric encryption algorithm (e.g., AES) as the image encryption key. Then, TA generates a set of $(2d + 6) \times (2d + 6)$ -dimensional invertible matrices

$$SK_{DO} = \left\{ M_1^{u,v,w}, (\tilde{M}_1^{u,v,w})^{-1}, M_2^{u,v,w}, (\tilde{M}_2^{u,v,w})^{-1} \right\}_{u,v,w=1}^2$$

as the encryption key of feature vectors. For a QU $QU_{id,S}$ who owns the identity id and the attribute set S , TA randomly splits these matrices into two matrices, respectively, such that

$$M_{1,id}^{u,v,w} \cdot [M]_{1,id}^{u,v,w} = M_1^{u,v,w}, \tilde{M}_{1,id}^{u,v,w} \cdot [\tilde{M}]_{1,id}^{u,v,w} = \tilde{M}_1^{u,v,w} \\ M_{2,id}^{u,v,w} \cdot [M]_{2,id}^{u,v,w} = M_2^{u,v,w}, \tilde{M}_{2,id}^{u,v,w} \cdot [\tilde{M}]_{2,id}^{u,v,w} = \tilde{M}_2^{u,v,w}.$$

Thus, the encryption key of query vector is

$$SK_{id} = \left\{ (M_{1,id}^{u,v,w})^{-1}, (\tilde{M}_{1,id}^{u,v,w})^{-1}, M_{2,id}^{u,v,w}, \tilde{M}_{2,id}^{u,v,w} \right\}_{u,v,w=1}^2.$$

and the conversion key of query vector is

$$SK_{id,CS} = \left\{ ([M]_{1,id}^{u,v,w})^{-1}, ([\tilde{M}]_{1,id}^{u,v,w})^{-1} \right. \\ \left. [M]_{2,id}^{u,v,w}, [\tilde{M}]_{2,id}^{u,v,w} \right\}_{u,v,w=1}^2.$$

Next, TA computes $x = Enc_{k'_1}(id)$, $y = f(x)$, $c = Enc_{k'_2}(x|y)$ for $QU_{id,S}$, and randomly chooses $r, r_1, r_2, \dots, r_k \in \mathbb{Z}_p$ to generate the decryption key $sk_{id,S}$ of $QU_{id,S}$,

$$sk_{id,S} = \{K = g_1^{\alpha/(a+c)} g_2^{r_1}, K' = c, L = g_1^{r_1}, L' = g_1^{ar} \\ \{K_{\tau,1} = g_1^{r_\tau}, K_{\tau,2} = (g_3^{A_\tau} g_4)^{r_\tau} g_5^{-(a+c)r_\tau}\}_{\tau \in [k]}\}.$$

Finally, TA distributes $\{k_{img}, SK_{DO}\}$ to DO, $\{SK_{id}, sk_{id,S}\}$ to $QU_{id,S}$, and $SK_{id,CS}$ to CS.

3) **DataEnc**($\mathcal{M}, SK_{DO}, k_{img}$) $\rightarrow$ (CI, CM): Given the image dataset $\mathcal{M} = \{IMG_i\}_{i=1}^n$, the feature vector encryption key SK_{DO} , and the image encryption key k_{img} , DO generates the encrypted index CI and the encrypted images CM .

DO executes the Algorithm 1 to build an encrypted two-layer index CI . First, DO extracts the feature vectors of $\mathcal{M}$ and groups them into K clusters (lines 1–3). Then, DO encrypts n feature vectors (lines 4 and 5) and K cluster center vectors (lines 6 and 7) to obtain the ciphertexts $\{CF_i\}_{i=1}^n$ and $\{CC_i\}_{i=1}^K$. As shown in Fig. 2, the first layer stores $\{CC_i\}_{i=1}^K$ and the second layer stores $\{CF_i\}_{i=1}^n$. Besides, DO encrypts $\mathcal{M}$ with k_{img} and symmetric encryption algorithm (e.g., AES)

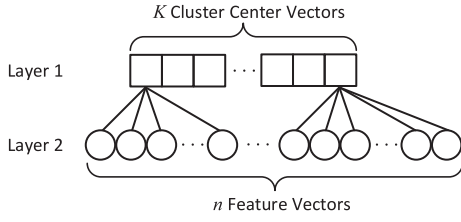


Fig. 2. Two-layer index structure of SCRM.

Algorithm 1: Generate Encrypted Index CI **Input:** The image dataset $\mathcal{M}$ and the encryption key SK_{DO} .**Output:** The encrypted index CI .

```

1 for each image  $IMG_i$  in  $\mathcal{M}$  do
2   Extract the feature vector  $f_i$ ;
3   Group  $FV \leftarrow \{f_1, f_2, \dots, f_n\}$  into  $K$  clusters
    $CL \leftarrow \{C_1, C_2, \dots, C_K\}$ ;
4   for  $1 \leq i \leq n$  do
5      $CF_i \leftarrow \text{EncryptVector}(FV_i, SK_{DO})$ ;
6   for  $1 \leq i \leq K$  do
7      $CC_i \leftarrow \text{EncryptVector}(CL_i, SK_{DO})$ ;
8    $CI \leftarrow \{\{CF_i\}_{i=1}^n, \{CC_i\}_{i=1}^K\}$ ;
9   return  $CI$ ;

Function EncryptVector( $f, SK_{DO}$ ):
1  $f_1 \leftarrow (f, -\frac{1}{2}\|f\|^2, -\mathbf{1}_{d+1})^\top$  //  $\mathbf{1}_{d+1}$  stands for a
    $(d+1)$ -dimensional vector and all elements
   are 1
2  $f_2 \leftarrow (\mathbf{1}_{d+1}, f, -\frac{1}{2}\|f\|^2)^\top$ ;
3 Select random numbers  $\beta_1 > \theta_1 > 0, \beta_2 > \theta_2 > 0, \tilde{\beta}_1 > \tilde{\theta}_1 > 0, \tilde{\beta}_2 > \tilde{\theta}_2 > 0, \delta_1, \delta_2, \tilde{\delta}_1, \tilde{\delta}_2$ ;
4  $F_1 \leftarrow (\beta_1 \cdot f_1, \delta_1, \delta_1, \theta_1, 1)^\top$ ;
5  $F_2 \leftarrow (\beta_2 \cdot f_2, \delta_2, -\delta_2, \theta_2, 1)^\top$ ;
6  $\tilde{F}_1 \leftarrow (\tilde{\beta}_1 \cdot f_1, \tilde{\delta}_1, \tilde{\delta}_1, \tilde{\theta}_1, 1)^\top$ ;
7  $\tilde{F}_2 \leftarrow (\tilde{\beta}_2 \cdot f_2, \tilde{\delta}_2, -\tilde{\delta}_2, \tilde{\theta}_2, 1)^\top$ ;
8 Randomly split  $F_1$  to  $F_{1,1}, F_{1,2}$  such that  $F_{1,1} + F_{1,2} = F_1$ ;
9 Randomly split  $F_2$  to  $F_{2,1}, F_{2,2}$  such that  $F_{2,1} + F_{2,2} = F_2$ ;
10 Randomly split  $\tilde{F}_1$  to  $\tilde{F}_{1,1}, \tilde{F}_{1,2}$  such that  $\tilde{F}_{1,1} + \tilde{F}_{1,2} = \tilde{F}_1$ ;
11 Randomly split  $\tilde{F}_2$  to  $\tilde{F}_{2,1}, \tilde{F}_{2,2}$  such that  $\tilde{F}_{2,1} + \tilde{F}_{2,2} = \tilde{F}_2$ ;
12 for  $1 \leq u \leq 2$  do
13   for  $1 \leq v \leq 2$  do
14     for  $1 \leq w \leq 2$  do
15        $CF_1^{u,v,w} \leftarrow F_{1,u}^\top M_{1,v,w}^{u,v,w}$ ;
16        $CF_2^{u,v,w} \leftarrow (M_{2,v,w}^{u,v,w})^{-1} F_{2,w}$ ;
17        $\tilde{CF}_1^{u,v,w} \leftarrow \tilde{F}_{1,u}^\top \tilde{M}_{1,v,w}^{u,v,w}$ ;
18        $\tilde{CF}_2^{u,v,w} \leftarrow (\tilde{M}_{2,v,w}^{u,v,w})^{-1} \tilde{F}_{2,w}$ ;
19 return  $\{CF_1^{u,v,w}, CF_2^{u,v,w}, \tilde{CF}_1^{u,v,w}, \tilde{CF}_2^{u,v,w}\}_{u,v,w=1}^2$ ;

```

into $\mathcal{CM} = \{\text{CIMG}_i\}_{i=1}^n$. Finally, DO sends the encrypted two-layer indexes $CI = \{\{CF_i\}_{i=1}^n, \{CC_i\}_{i=1}^K\}$ and the encrypted images $\mathcal{CM}$ to CS.

4) **KeyEnc**($k_{img}, PP, \mathbb{A}$) $\rightarrow CT$: Given the image encryption key $k_{img} \in \mathbb{G}_T$, the public parameters PP , and an access policy $\mathbb{A} = (\hat{M}, \rho) \in (\mathbb{Z}_p^{l \times \gamma}, [l] \rightarrow \mathbb{Z}_p)$, DO encrypts k_{img} to realize the access control.

DO randomly chooses $y = (s, y_2, \dots, y_\gamma)^\top \in \mathbb{Z}_p^{\gamma \times 1}$, and computes $\lambda_i = \hat{M}_i y$ for $i \in [\gamma]$, where $\hat{M}_i$ is the i -th row of $\hat{M}$. Then, DO randomly selects $t_1, t_2, \dots, t_\gamma$ from $\mathbb{Z}_p$ and

generates the ciphertext

$$CT = \left\{ (\hat{M}, \rho), C = k_{img} \cdot e(g_1, g_1)^{\alpha s}, C_0 = g_1^s, C'_0 = g_1^{as} \right. \\ \left. \left\{ C_{i,1} = g_2^{\lambda_i} g_5^{t_i}, C_{i,2} = (g_3^{\rho(i)} g_4)^{-t_i}, C_{i,3} = g_1^{t_i} \right\}_{i \in [\gamma]} \right\}.$$

DO sends CT to CS.

5) **QueryGen**(IMG_q, SK_{id}) $\rightarrow CQ_{id}$: With the encryption key SK_{id} of query vector, $QU_{id,S}$ generates an encrypted query trapdoor CQ_{id} to CS to obtain the most similar image of IMG_q .

$QU_{id,S}$ extracts the d -dimensional feature vector q of IMG_q and extends it to $q' = (q, 1)$. Based on q' , $QU_{id,S}$ constructs a $(2d+2) \times (2d+2)$ -dimensional matrix

$$Q' = \begin{bmatrix} \text{diag}(q') & \mathbf{0} \\ \mathbf{0} & \text{diag}(q') \end{bmatrix}$$

where $\text{diag}(q')$ is a diagonal matrix with q' as the diagonal element and $\mathbf{0}$ is a zero matrix. Then, $QU_{id,S}$ selects random numbers $\beta_q > \theta_q > 0, \tilde{\beta}_q > \tilde{\theta}_q > 0, \delta_q, \tilde{\delta}_q, \xi_q$ from $\mathbb{R}$ and constructs two matrices

$$Q = \begin{bmatrix} \beta_q Q' & 0 & 0 & 0 & 0 \\ 0 & \delta_q & 0 & 0 & 0 \\ 0 & 0 & \delta_q & 0 & 0 \\ 0 & 0 & 0 & \theta_q & 0 \\ 0 & 0 & 0 & 0 & \xi_q \end{bmatrix}, \tilde{Q} = \begin{bmatrix} \tilde{\beta}_q Q' & 0 & 0 & 0 & 0 \\ 0 & \tilde{\delta}_q & 0 & 0 & 0 \\ 0 & 0 & \tilde{\delta}_q & 0 & 0 \\ 0 & 0 & 0 & \tilde{\theta}_q & 0 \\ 0 & 0 & 0 & 0 & -\xi_q \end{bmatrix}.$$

$QU_{id,S}$ splits Q and $\tilde{Q}$ into two matrices, respectively, such that

$$Q_1 + Q_2 = Q, \tilde{Q}_1 + \tilde{Q}_2 = \tilde{Q}.$$

Next, $QU_{id,S}$ uses SK_{id} to encrypt them as

$$CQ_{id}^{u,v,w} = (M_{1,i}^{u,v,w})^{-1} Q_v M_{2,id}^{u,v,w} \\ \tilde{CQ}_{id}^{u,v,w} = (\tilde{M}_{1,i}^{u,v,w})^{-1} \tilde{Q}_v \tilde{M}_{2,id}^{u,v,w}$$

where $u, v, w \in \{1, 2\}$. At last, the query trapdoor $CQ_{id} = \{CQ_{id}^{u,v,w}, \tilde{CQ}_{id}^{u,v,w}\}_{u,v,w=1}^2$ is sent to CS.

6) **Search**(CI, CQ_{id}, SK_{id}, CS) $\rightarrow R_q$: Given $CF_i, CF_j \in CI$, to find the image more similar to IMG_q , CS computes

$$z = \sum_{u,v,w=1}^2 \left(CF_{1,i}^{u,v,w} \cdot ([M]_{1,id}^{u,v,w})^{-1} \cdot CQ_{id}^{u,v,w} \right. \\ \left. \cdot [M]_{2,id}^{u,v,w} \cdot CF_{2,j}^{u,v,w} \right) \\ \tilde{z} = \sum_{u,v,w=1}^2 \left(\tilde{CF}_{1,i}^{u,v,w} \cdot ([\tilde{M}]_{1,id}^{u,v,w})^{-1} \cdot \tilde{CQ}_{id}^{u,v,w} \right. \\ \left. \cdot [\tilde{M}]_{2,id}^{u,v,w} \cdot \tilde{CF}_{2,j}^{u,v,w} \right).$$

Then, CS computes $z + \tilde{z}$ to filter the image more similar to IMG_q . If $z + \tilde{z} > 0$, the more similar image is IMG_i , otherwise IMG_j . To find the image most similar to IMG_q in $\mathcal{M}$, CS executes the Algorithm 2. First, CS finds the cluster most similar to the query in CC (line 2). Then, CS finds the identifier of the most similar feature in the most similar

Algorithm 2: Find The Identifier of The Most Similar Image

Input: The encrypted index CI and the query trapdoor CQ_{id} .

Output: The identifier of the most similar image.

```

1  $R \leftarrow \{CC\};$ 
2  $R_C \leftarrow \text{FindSimilar}(R, CQ_{id});$ 
3  $R_F \leftarrow \text{FindSimilar}(R_C.\text{nextlayer}, CQ_{id});$ 
4 return  $R_F.id;$ 

```

Function $\text{FindSimilar}(R, CQ_{id})$:

```

1 while  $|R| > 1$  do
2   for  $1 \leq i < |R|$  do
3      $t \leftarrow \text{FindUnsimilar}(R_i, R_{i+1}, CQ_{id});$ 
4     Delete  $R_i$  from  $R$ ;
5      $i \leftarrow i + 2;$ 
6 return  $R;$ 

```

Function $\text{FindUnsimilar}(CF_i, CF_j, CQ_{id})$:

```

1  $z \leftarrow \sum_{u,v,w=1}^2 (CF_{1,i}^{u,v,w} \cdot ([M]_{1,id}^{u,v,w})^{-1} \cdot CQ_{id}^{u,v,w} \cdot [M]_{2,id}^{u,v,w} \cdot CF_{2,j}^{u,v,w});$ 
2  $\tilde{z} \leftarrow \sum_{u,v,w=1}^2 (\tilde{CF}_{1,i}^{u,v,w} \cdot ([\tilde{M}]_{1,id}^{u,v,w})^{-1} \cdot \tilde{CQ}_{id}^{u,v,w} \cdot [\tilde{M}]_{2,id}^{u,v,w} \cdot \tilde{CF}_{2,j}^{u,v,w});$ 
3 if  $z + \tilde{z} > 0$  then
4   return  $j;$ 
5 else
6   return  $i;$ 

```

cluster¹ (line 3 to 4). Finally, CS returns the encrypted image CIMG according to the identity as well as the ciphertext CT of key.

7) $\text{ImgDec}(PP, MSK, sk_{id,S}, CT, CIMG) \rightarrow IMG$ or $\perp$: With the public parameters PP , the master key MSK , the decryption key $sk_{id,S}$, the ciphertext CT of image encryption key, the ciphertext CIMG of most similar image, $QU_{id,S}$ either gets the plaintext IMG of CIMG or nothing $\perp$.

$QU_{id,S}$ first computes the set I of rows in $\tilde{M}$ that can produce a share to attributes in S , $I = \{i | \rho(i) \in S\}$. If the attribute set S is not an authorized set on $(\tilde{M}, \rho)$, then the algorithm outputs $\perp$ and terminates the process; otherwise, let $\{\omega_i \in \mathbb{Z}_p\}_{i \in I}$ be $|I|$ constants such that $\sum_{i \in I} \tilde{M}_i \omega_i = (1, 0, \dots, 0)$, where $\tilde{M}_i$ is the i th row of $\tilde{M}$. Thus, $\sum_{i \in I} \lambda_i \omega_i = s$. $QU_{id,S}$ computes

$$\begin{aligned}
 B_1 &= e(K, C_0^{K'} C_0') \\
 B_2 &= \prod_{i \in I} \left(e(L^{K'} L', C_{i,1}) e(K_{\tau,1}, C_{i,2}) e(K_{\tau,2}, C_{i,3}) \right)^{\omega_i} \\
 B_3 &= B_1 / B_2,
 \end{aligned}$$

where τ is the attribute $\rho(i)$'s index in S . The decryption key of the image is $k_{img} = C/B_3$. Finally, $QU_{id,S}$ uses the symmetric decryption algorithm (e.g., AES) and k_{img} to obtain the plaintext IMG of CIMG.

¹ Although SCRM only returns the most similar results, readers can further use heap sort to return the top-k most similar results.

8) $\text{UserTrace}(PP, MSK, sk_{id,S}) \rightarrow id$ or $\perp$: For the suspicious key $sk_{id,S}$, TA uses the public parameters PP and the master key MSK to obtain the id of the user to whom the key belongs.

First, TA performs a key sanity check. The decryption key $sk_{id,S}$ passes the key sanity check if

- 1) $sk_{id,S}$ is in the form of $(K, K', L, L', \{K_{\tau,1}, K_{\tau,2}\}_{\tau \in [k]})$;
- 2) $e(L', g_1) = e(L, g_1^a)$;
- 3) $e(K, g_1^a g_1^{K'}) = e(g_1, g_1)^\alpha e(L' L^{K'}, g_2)$;
- 4) $\exists \tau \in [k], \text{ s.t. } e(K_{\tau,2}, g_1) e(L^{K'} L', v) = e(K_{\tau,1}, g_4) e(K_{\tau,1}, g_3)^{A_\tau}$.

If $sk_{id,S}$ cannot pass the key sanity check, the algorithm outputs $\perp$. Otherwise, $sk_{id,S}$ is a well-formed decryption key, and the algorithm extracts $(x^* = x, y^* = y)$ by decrypting K' with k'_2 . If $(x^* = x, y^* = y) \in \{(x_1, y_1), (x_2, y_2), \dots, (x_{t-1}, y_{t-1})\}$, the algorithm decrypts x^* with k'_1 to obtain id . According to id , the malicious user can be identified. Otherwise, the algorithm recovers the secret s_0^* of Shamir's threshold scheme based on $\{(x_1, y_1), (x_2, y_2), \dots, (x_{t-1}, y_{t-1})\}$ and $(x^* = x, y^* = y)$. If $s_0^* = f(0)$, the algorithm decrypts x^* with k'_1 to obtain id of the malicious user. Otherwise, the algorithm outputs $\perp$.

Remark: When the system needs to revoke a user's access rights due to key leakage, the most straightforward approach is for the CS to maintain a revocation list. This list stores the IDs of revoked users. If a user's ID appears on the list, the CS will deny their query requests, preventing them from obtaining query results, let alone decrypting them.

VI. THEORETICAL ANALYSIS

We analyze SCRM from both correctness and security aspects. Specifically, we verify the correctness of retrieval and access control, as well as the security of retrieval. The security and traceability of traceable attribute-based encryption have already been proven in [36], so we will not elaborate on them in this article.

A. Correctness Analysis

1) *Correctness of Retrieval:* In the *Search* algorithm, CS first finds the candidate cluster center vector that is most similar to the query trapdoor, then finds the most similar image feature vector within that cluster, and finally returns the identifier corresponding to the vector. Next, we prove that CS can find the correct result after computation and comparison. For convenience, we take IMG_i and IMG_j as examples. First, CS computes

$$\begin{aligned}
 z &= \sum_{u,v,w=1}^2 \left(CF_{1,i}^{u,v,w} \cdot ([M]_{1,id}^{u,v,w})^{-1} \cdot CQ_{id}^{u,v,w} \cdot [M]_{2,id}^{u,v,w} \cdot CF_{2,j}^{u,v,w} \right) \\
 &= \beta_{1,i} \cdot \beta_q \cdot \beta_{2,j} \cdot f_{1,i}^\top Q' f_{2,j} + \theta_{1,i} \cdot \theta_q \cdot \theta_{2,j} + \xi_q \\
 \tilde{z} &= \sum_{u,v,w=1}^2 \left(\tilde{CF}_{1,i}^{u,v,w} \cdot ([\tilde{M}]_{1,id}^{u,v,w})^{-1} \cdot \tilde{CQ}_{id}^{u,v,w} \cdot [\tilde{M}]_{2,id}^{u,v,w} \cdot \tilde{CF}_{2,j}^{u,v,w} \right) \\
 &= \tilde{\beta}_{1,i} \cdot \tilde{\beta}_q \cdot \tilde{\beta}_{2,j} \cdot f_{1,i}^\top Q' f_{2,j} + \tilde{\theta}_{1,i} \cdot \tilde{\theta}_q \cdot \tilde{\theta}_{2,j} - \xi_q.
 \end{aligned}$$

Therefore,

$$z + \tilde{z} = (\beta_{1,i} \cdot \beta_q \cdot \beta_{2,j} + \tilde{\beta}_{1,i} \cdot \tilde{\beta}_q \cdot \tilde{\beta}_{2,j}) \cdot f_{1,i}^\top Q' f_{2,j} + (\theta_{1,i} \cdot \theta_q \cdot \theta_{2,j} + \tilde{\theta}_{1,i} \cdot \tilde{\theta}_q \cdot \tilde{\theta}_{2,j}).$$

Since

$$\begin{aligned} \beta_{1,i} &> \theta_{1,i} > 0, \tilde{\beta}_{1,i} > \tilde{\theta}_{1,i} > 0 \\ \beta_{2,j} &> \theta_{2,j} > 0, \tilde{\beta}_{2,j} > \tilde{\theta}_{2,j} > 0 \\ \beta_q &> \theta_q > 0, \tilde{\beta}_q > \tilde{\theta}_q > 0 \end{aligned}$$

we have

$$\begin{aligned} \beta_{1,i} \cdot \beta_q \cdot \beta_{2,j} + \tilde{\beta}_{1,i} \cdot \tilde{\beta}_q \cdot \tilde{\beta}_{2,j} &> \\ \theta_{1,i} \cdot \theta_q \cdot \theta_{2,j} + \tilde{\theta}_{1,i} \cdot \tilde{\theta}_q \cdot \tilde{\theta}_{2,j} &> 0. \end{aligned}$$

Then,

$$\begin{aligned} z + \tilde{z} > 0 &\Leftrightarrow f_{1,i}^\top Q' f_{2,j} \geq 0 \\ &\Leftrightarrow [f_i - \frac{1}{2}\|f_i\|^2 - \mathbf{1}_{d+1}] \cdot \begin{bmatrix} \text{diag}(q') & \mathbf{0} \\ \mathbf{0} & \text{diag}(q') \end{bmatrix} \cdot \begin{bmatrix} \mathbf{1}_{d+1}^\top \\ f_j^\top \\ -\frac{1}{2}\|f_j\|^2 \end{bmatrix} \geq 0 \\ &\Leftrightarrow f_i \cdot q^\top - \frac{1}{2}\|f_i\|^2 - q \cdot f_j^\top + \frac{1}{2}\|f_j\|^2 \geq 0 \\ &\Leftrightarrow \left(\frac{1}{2}\|f_i\|^2 - f_i \cdot q^\top + \frac{1}{2}\|q\|^2 \right) \\ &\quad - \left(\frac{1}{2}\|q\|^2 - q \cdot f_j^\top + \frac{1}{2}\|f_j\|^2 \right) \leq 0 \\ &\Leftrightarrow \|f_i - q\|^2 - \|f_j - q\|^2 \leq 0 \\ &\Leftrightarrow d(f_i, q) - d(f_j, q) \leq 0. \end{aligned}$$

Therefore, compared to IMG_j , IMG_i is closer to IMG_q . Similarly, for all vectors in CI , CS can get the most similar result.

2) *Correctness of Access Control*: In the *ImgDec* algorithm, if the attributes of $QU_{id,S}$ satisfy the access control policy, then $QU_{id,S}$ can obtain the decryption key of the image after calculating C/B_3 . Specifically, $QU_{id,S}$ calculates

$$\begin{aligned} B_3 &= B_1/B_2 = \frac{e(g_1, g_1)^{\alpha s} e(g_2, g_1)^{(a+c)sr}}{\prod_{i \in I} D_1 \cdot D_2 \cdot D_3 \cdot D_4 \cdot D_5} \\ &= \frac{e(g_1, g_1)^{\alpha s} e(g_2, g_1)^{(a+c)sr}}{e(g_1, g_2)^{(a+c)r \sum_{i \in I} \lambda_i \omega_i}} = e(g_1, g_1)^{\alpha s} \end{aligned}$$

where

$$\begin{aligned} D_1 &= e(g_1, g_2)^{(a+c)r\lambda_i \omega_i}, \quad D_2 = e(g_1, g_5)^{(a+c)rt_i \omega_i} \\ D_3 &= e(g_1, g_3^{\rho_i} g_4)^{-r_\tau t_i \omega_i}, \quad D_4 = e(g_3^{\rho_i} g_4, g_1)^{r_\tau t_i \omega_i} \\ D_5 &= e(g_5, g_1)^{-(a+c)rt_i \omega_i}. \end{aligned}$$

Then, $QU_{id,S}$ computes

$$\frac{C}{B_3} = \frac{k_{img} \cdot e(g_1, g_1)^{\alpha s}}{e(g_1, g_1)^{\alpha s}} = k_{img}.$$

Therefore, $QU_{id,S}$ with the attribute set S can obtain the decryption key k_{img} .

B. Security Analysis

Same as [37], we prove that SCRM is selectively secure in the real/ideal world security model. The leakage $\mathcal{L}$ of SCRM is the intermediate results during retrieval. We define the real and ideal experiments based on $\mathcal{L}$.

Real Experiment: The real experiment consists of two participants, i.e., a Probabilistic Polynomial Time (PPT) adversary $\mathcal{A}$ and a challenger C , and they interact as follows.

1) *Setup phase*: $\mathcal{A}$ chooses p_1 images, extracts their feature vectors and sends them to C . Then, C generates the secret key

$$\begin{aligned} SK_{DO} &= \left\{ M_1^{u,v,w}, (\tilde{M}_1^{u,v,w})^{-1}, M_2^{u,v,w}, (\tilde{M}_2^{u,v,w})^{-1} \right\}_{u,v,w=1}^2 \\ SK_{id} &= \left\{ \left(M_{1,id}^{u,v,w} \right)^{-1}, \left(\tilde{M}_{1,id}^{u,v,w} \right)^{-1}, M_{2,id}^{u,v,w}, \tilde{M}_{2,id}^{u,v,w} \right\}_{u,v,w=1}^2 \\ SK_{id,CS} &= \left\{ \left([M]_{1,id}^{u,v,w} \right)^{-1}, \left([\tilde{M}]_{1,id}^{u,v,w} \right)^{-1} \right. \\ &\quad \left. [M]_{2,id}^{u,v,w}, [\tilde{M}]_{2,id}^{u,v,w} \right\}_{u,v,w=1}^2. \end{aligned}$$

Besides, C generates the encrypted index CI and encrypted images CM for $1 \leq i \leq p_1$.

2) *Query phase 1*: $\mathcal{A}$ chooses p_2 query images and extracts their feature vectors $\{q_j\}_{j=1}^{p_2}$, where p_2 is a polynomial number. Then, $\mathcal{A}$ sends $\{q_j\}_{j=1}^{p_2}$ to C . After that, C encrypts $\{q_j\}_{j=1}^{p_2}$ into $\{CQ_{j,id}\}_{j=1}^{p_2}$ and returns them to $\mathcal{A}$.

3) *Challenge phase*: C returns CI and CM to $\mathcal{A}$.

4) *Query phase 2*: $\mathcal{A}$ chooses $p'_2 - p_2$ images, and gets their $\{CQ_{j,id}\}_{j=p_2+1}^{p'_2}$ in the same way as *Query phase 1*.

Ideal Experiment: The ideal experiment consists of two participants, i.e., a PPT adversary $\mathcal{A}$ and a simulator $\mathcal{S}$ with the leakage $\mathcal{L}$, and they interact as follows.

1) *Setup*: $\mathcal{A}$ chooses p_1 images, extracts their feature vectors and sends them to $\mathcal{S}$. After that, $\mathcal{S}$ generates some random matrices as the ciphertext index CI^S of p_1 images, where the matrices have the same size as those generated in SCRM.

2) *Query phase 1*: $\mathcal{A}$ chooses p_2 query images and extracts their feature vectors $\{q_j\}_{j=1}^{p_2}$, where p_2 is a polynomial number. Then, $\mathcal{A}$ sends $\{q_j\}_{j=1}^{p_2}$ to $\mathcal{S}$. After that, $\mathcal{S}$ uses the leakage function $\mathcal{L}$ and CI^S to generate the ciphertext of $\{q_j\}_{j=1}^{p_2}$. For each q_j , $\mathcal{S}$ generates its ciphertext $CQ_{j,id}^S$ as follows. According to the comparison results in the leakage function $\mathcal{L}$, assume that vector f_{i_1} is more similar to q_j than f_{i_2} , if f_{i_1} is on the left of f_{i_2} when calculating z and $\tilde{z}$, then the calculation result $z + \tilde{z}$ is set to a random number r'_{i_1,j,i_2} greater than 0, otherwise it is set to a random number r'_{i_1,j,i_2} less than 0. Then a set of random matrices is selected as the ciphertext of q_j , so that the similarity comparison result between f_{i_1}, f_{i_2}, q_j is equal to r_{i_1,j,i_2} or r'_{i_1,j,i_2} (depending on the calculation order). Finally, $\mathcal{S}$ returns $\{CQ_{j,id}^S\}_{j=1}^{p_2}$ to $\mathcal{A}$.

3) *Challenge phase*: $\mathcal{S}$ returns CI^S to $\mathcal{A}$.

4) *Query phase 2*: $\mathcal{S}$ chooses $p'_2 - p_2$ images, and gets their $\{CQ_{j,id}^S\}_{j=p_2+1}^{p'_2}$ in the same way as *Query phase 1*.

In the real experiment, the view of $\mathcal{A}$ is $View_{\mathcal{A},Real} = \{CI, CM, \{CQ_{j,id}\}_{j=1}^{p'_2}\}$. In the ideal experiment, the view of

$\mathcal{A}$ is $\text{View}_{\mathcal{A}, \text{Ideal}} = \{CI^S, CM^S, \{CQ_{j,id}^S\}_{j=1}^{p'_2}\}$. Then, based on the views of $\mathcal{A}$, we define the security of SCRM.

Definition 3 (Security of SCRM): Our SCRM is selectively secure with the leakage $\mathcal{L}$ iff there exists a simulator $\mathcal{S}$ such that the advantage that any PPT adversary $\mathcal{A}$ can distinguish the views of real and ideal experiments is negligible, i.e., $|\Pr[\text{View}_{\mathcal{A}, \text{Real}} = 1] - \Pr[\text{View}_{\mathcal{A}, \text{Ideal}} = 1]|$ is negligible.

Theorem 1: SCRM is selectively secure with $\mathcal{L}$.

Proof: We prove that SCRM is selectively secure with the leakage $\mathcal{L}$ by proving that $\mathcal{A}$ cannot distinguish the views of real and ideal experiments. Since the ciphertexts in the ideal experiment are randomly generated by $\mathcal{S}$, distinguishing the views of real and ideal experiments is equivalent to distinguishing the ciphertexts in the real experiment from random ciphertexts. In the following, we prove indistinguishability regarding indexes, query trapdoors, and results.

- 1) $CI = \{\{CF_i\}_{i=1}^n, \{CC_i\}_{i=1}^K\}$ is indistinguishable from random ciphertexts. According to the *DataEnc* algorithm, $CF_i = \{CF_{1,i}^{u,v,w}, CF_{2,i}^{u,v,w}, \tilde{CF}_{1,i}^{u,v,w}, \tilde{CF}_{2,i}^{u,v,w}\}_{u,v,w=1}^2$, where

$$CF_{1,i}^{u,v,w} = F_{1,i,u}^\top M_1^{u,v,w}, \quad CF_{2,i}^{u,v,w} = (M_2^{u,v,w})^{-1} F_{2,i,w}$$

$$\tilde{CF}_{1,i}^{u,v,w} = \tilde{F}_{1,i,u}^\top \tilde{M}_1^{u,v,w}, \quad \tilde{CF}_{2,i}^{u,v,w} = (\tilde{M}_2^{u,v,w})^{-1} \tilde{F}_{2,i,w}.$$

$F_{1,i}, F_{2,i}, \tilde{F}_{1,i}, \tilde{F}_{2,i}$ with many random numbers added are randomly split, and then one of the shares are multiplied by the random matrices, so the final CF_i is indistinguishable from the random ciphertext. Similarly, CC_i is also indistinguishable from the random ciphertext. Therefore, the PPT adversary $\mathcal{A}$ cannot distinguish CI from the random ciphertext.

- 2) $\{CQ_{j,id}\}_{j=1}^{p'_2}$ are indistinguishable from random ciphertexts. According to the *QueryGen* algorithm, $CQ_{j,id} = \{CQ_{j,id}^{u,v,w}, \tilde{CQ}_{j,id}^{u,v,w}\}_{u,v,w=1}^2$, where

$$CQ_{j,id}^{u,v,w} = (M_{1,id}^{u,v,w})^{-1} Q_{v,j} M_{2,id}^{u,v,w}$$

$$\tilde{CQ}_{j,id}^{u,v,w} = (\tilde{M}_{1,id}^{u,v,w})^{-1} \tilde{Q}_{v,j} \tilde{M}_{2,id}^{u,v,w}.$$

Since many random numbers are added to q , and then the expanded q is randomly split, and one of the shares is multiplied by the random matrix, the final CQ is indistinguishable from the random ciphertext. Therefore, the PPT adversary $\mathcal{A}$ cannot distinguish $\{CQ_{j,id}\}_{j=1}^{p'_2}$ from random ciphertext.

- 3) $\{z_{i_1,j,i_2} + \tilde{z}_{i_1,j,i_2} | 1 \leq i_1, i_2 \leq p_2; 1 \leq j \leq p'_2\}$ are indistinguishable from random numbers. In the *Search* algorithm, z_{i_1,j,i_2} and $\tilde{z}_{i_1,j,i_2}$ contain many random numbers. With these random numbers, $z_{i_1,j,i_2} + \tilde{z}_{i_1,j,i_2}$ is indistinguishable from a random number. Thus, the PPT adversary $\mathcal{A}$ cannot distinguish $\{z_{i_1,j,i_2} + \tilde{z}_{i_1,j,i_2} | 1 \leq i_1, i_2 \leq p_2; 1 \leq j \leq p'_2\}$ from random ciphertext.

As a result, we can deduce that $\text{View}_{\mathcal{A}, \text{Real}}$ is indistinguishable from the random ciphertexts. That is, $\text{View}_{\mathcal{A}, \text{Real}}$ is indistinguishable from $\text{View}_{\mathcal{A}, \text{Ideal}}$, and the PPT adversary $\mathcal{A}$ cannot distinguish the views of real and ideal experiments. Therefore, our SCRM is selectively secure with $\mathcal{L}$. ■

VII. PERFORMANCE EVALUATION

In this section, we evaluate the performance of our SCRM and compare it with the fast nearest neighbor search scheme over encrypted images (called FNNS) [20].

A. Experimental Setting

1) *Experiment Environment:* We deployed SCRM and FNNS on a machine with Intel Xeon Gold 5318Y CPU at 2.10 GHz and 14-GB RAM. We implemented all operations in Python (Python 3.10) language on the Ubuntu 20.04 platform. In the experiments, we set 256 bits key size for CP-ABE in SCRM and 1024 bits key size for Paillier in FNNS.

2) *Experiment Dataset:* We used the classical Caltech-256 [38] to test the actual experimental performance. The Caltech-256 dataset contains 256 different classes, each with a label and hundreds of images. We selected 50 classes and 40, 50, 60, 70, and 80 images per class, resulting in 2000, 2500, 3000, 3500, and 4000 images as experimental datasets. We tested the accuracy, computational overhead, and storage overhead on different total numbers of images.

B. Experimental Results

1) *Accuracy:* SCRM implements nearest neighbor retrieval, that is, retrieving the image in the dataset that is most similar to the query image. We measure the accuracy by determining whether the returned image is from the same class as the query image. After testing, the accuracy of SCRM has remained at 100% when the dataset changes from 2000 to 4000. Moreover, from the correctness analysis, it can be seen that the random numbers we introduced when encrypting the index will not affect the final similarity measurement results. Therefore, the encryption method used in SCRM to protect the features will not affect the retrieval accuracy. Since SCRM constructs a two-layer index, the clustering effect will affect the retrieval accuracy. The evaluation in SCRM uses the classic K -means clustering algorithm. In order to maintain excellent retrieval accuracy on large-scale data, other more advanced clustering algorithms are also optional.

2) *Computational Overhead:* Fig. 3 shows the computational overhead of SCRM, where Fig. 3(a)–(f) show the computational overhead associated with the retrieval, and Fig. 3(g) shows the computational overhead associated with the access control. In Fig. 3(a), the time of SCRM generating the key for retrieval is displayed. At this stage, SCRM generates a symmetric encryption key and some random matrices. When the dimension d of the feature vector increases, the key generation time of SCRM also increases accordingly. It should be noted that Fig. 3(a) does not include the time of generating keys for access control.

The *DataEnc* algorithm generates an encrypted index and an encrypted dataset, and the time consumed is shown in Fig. 3(b) and (c). When the total number of clusters K is fixed, the time consumed increases with the increase of dimension d and the number of images n . This is because when generating an encrypted index, it is necessary to perform a clustering operation on n vectors and then perform $n+K$ vector randomization operations. Both clustering and vector randomization

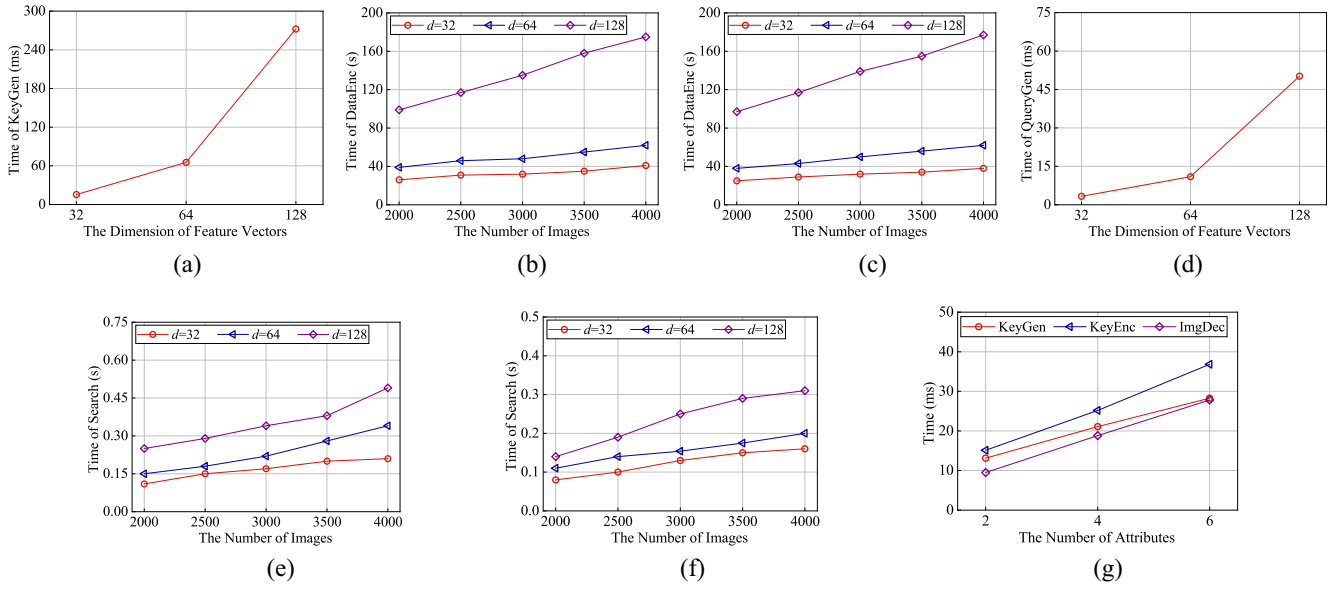


Fig. 3. Computational overhead of SCRM. (a) KeyGen, Without Access Control. (b) DataEnc, $K = 5$. (c) DataEnc, $K = 10$. (d) QueryGen. (e) Search, $K = 5$. (f) Search, $K = 10$. (g) Access Control.

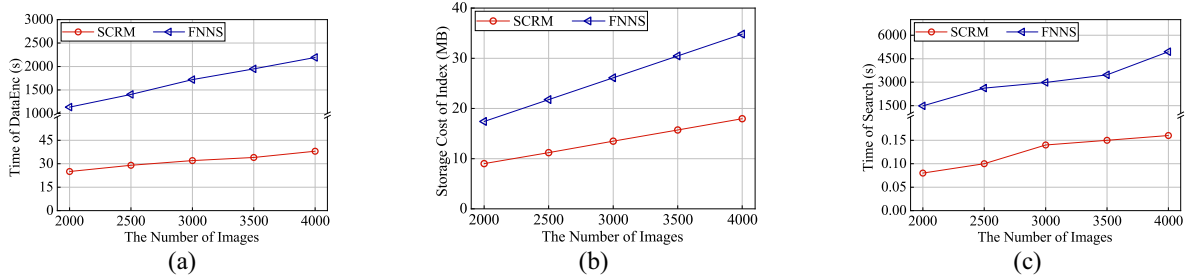


Fig. 4. Comparison of computational and storage overhead. (a) DataEnc. (b) DataEnc. (c) Search.

operations are closely related to d and n . In addition, when generating an encrypted dataset, it is necessary to perform n symmetric encryption algorithms. When K changes from 5 to 10, the clustering operation time changes, and 5 vector randomization operations are added. However, these changes do not have a significant impact on the overhead, so the time of DataEnc in Fig. 3(b) and (c) is similar.

Fig. 3(d) shows the relationship between query trapdoor generation time and the dimension d . Generating a query trapdoor requires only one vector randomization operation. Specifically, when d increases from 32 to 128, the query trapdoor generation time increases from 3.28 to 50.16 ms. This time overhead is acceptable in IoT environments. Even for resource-constrained devices, such as smartphones, it is sufficient to meet the demand for querying at any time and from anywhere. For example, on a smartphone (Snapdragon 8 Gen 1, 2.8 GHz, 6-GB RAM), the overhead of generating a query trapdoor is theoretically about three times that on a laptop. Therefore, the query trapdoor generation time for IoT devices is in the range of seconds, which is within an acceptable range.

Fig. 3(e) and (f) show the time of Search algorithm. The search process needs to match $K + |R_C.nextlayer|$ vectors and perform about $2(K + |R_C.nextlayer|)$ computations, where

$|R_C.nextlayer|$ represents the total number of leaf nodes contained in the cluster center that is most similar to the query. When the total number of clusters K is fixed, the time required for the search increases with the increase of n and d . When K changes from 5 to 10, $|R_C.nextlayer|$ becomes smaller, so the search time also decreases. If K increases more, the search time will be reduced more.

In Fig. 3(g), the overhead associated with access control is shown, specifically the key generation time, key encryption time, and image decryption time. Since the setup process is only performed once, we do not show the time it consumes. As can be seen from Fig. 3(g), the key generation time, key encryption time, and image decryption time increase almost linearly with the increase of the number of attributes. In particular, when the total number of attributes is six, the total time required to implement access control is within 100 ms. The execution time of ImgDec on resource-constrained devices, such as smartphones, is also within seconds. In addition, user traceability is divided into multiple situations. We simulated the UserTrace algorithm and verified its correctness under different situations. In short, it is efficient and feasible to implement access control and user traceability through attribute-based encryption.

3) *Compared With FNNS [20]*: Since FNNS does not implement access control and user traceability, we mainly compared the overhead of key steps, such as data encryption and search with FNNS. We set $d = 32$ and $K = 10$. As can be seen from Fig. 4(a), when the total number of images changes from 2000 to 4000, the data encryption overhead of SCRM is much lower than that of FNNS. This is because SCRM mainly encrypts feature vectors through randomization, while FNNS needs to perform multiple time-consuming modular exponentiation operations. In addition, the encrypted index of SCRM contains $(n + K) \times 8 \times (2d + 6)$ real numbers, while that of FNNS contains $n \times (d + 2)$ large integers. Therefore, the index storage overhead of SCRM in Fig. 4(b) is lower than that of FNNS. When retrieving the index, FNNS needs to perform multiple homomorphic additions, multiplications, and decryptions, while SCRM only needs to perform multiple vector multiplications. Therefore, the search overhead of SCRM in Fig. 4(c) is significantly lower than that of FNNS. It can be observed that SCRM holds a significant advantage over FNNS in the nearest neighbor retrieval of ciphertext images.

VIII. CONCLUSION

In this article, we proposed a similarity retrieval scheme in multiuser settings that effectively resists known-plaintext attacks while supporting fine-grained access control and traceability. Specifically, we developed a two-layer encrypted index based on features to enable secure ciphertext image retrieval. Additionally, we designed a key conversion protocol and incorporated attribute-based encryption to implement access control and traceability, enhancing controllability in multiuser settings. Through comprehensive theoretical analysis and experimental evaluations, we demonstrated that our SCRM achieves an effective balance between security and efficiency, offering a practical solution for encrypted image sharing applications.

REFERENCES

- [1] Q. Xie and Q. Xie, "Security analysis on a three-factor authentication scheme of 5G wireless sensor networks for IoT system," *IEEE Internet Things J.*, vol. 11, no. 8, pp. 15038–15042, Apr. 2024.
- [2] Z. Wan and R. H. Deng, "VPSearch: Achieving verifiability for privacy-preserving multi-keyword search over encrypted cloud data," *IEEE Trans. Depend. Secure Comput.*, vol. 15, no. 6, pp. 1083–1095, Nov./Dec. 2018.
- [3] F. Li, J. Ma, Y. Miao, X. Liu, J. Ning, and R. H. Deng, "A survey on searchable symmetric encryption," *ACM Comput. Surv.*, vol. 56, no. 5, pp. 1–42, 2023.
- [4] S. K. Pasupuleti, S. Ramalingam, and R. Buyya, "An efficient and secure privacy-preserving approach for outsourced data of resource constrained mobile devices in cloud computing," *J. Netw. Comput. Appl.*, vol. 64, pp. 12–22, Apr. 2016.
- [5] Y. Zheng et al., "Efficient and privacy-preserving spatial-feature-based reverse kNN query," *IEEE Trans. Services Comput.*, vol. 16, no. 4, pp. 2620–2631, Jul./Aug. 2022.
- [6] S. Zhang et al., "Performance enhanced secure spatial keyword similarity query with arbitrary spatial ranges," *IEEE Trans. Inf. Forensics Security*, vol. 19, pp. 5272–5285, 2024.
- [7] Q. Tong, X. Li, Y. Miao, Y. Wang, X. Liu, and R. H. Deng, "Beyond result verification: Efficient privacy-preserving spatial keyword query with suppressed leakage," *IEEE Trans. Inf. Forensics Security*, vol. 19, pp. 2746–2760, 2024.
- [8] J. Yuan, S. Yu, and L. Guo, "SEISA: Secure and efficient encrypted image search with access control," in *Proc. IEEE Conf. Comput. Commun.*, 2015, pp. 2083–2091.
- [9] Z. Xia, N. N. Xiong, A. V. Vasilakos, and X. Sun, "EPCBIR: An efficient and privacy-preserving content-based image retrieval scheme in cloud computing," *Inf. Sci.*, vol. 387, pp. 195–204, May 2017.
- [10] D. Li, Q. Lü, X. Liao, T. Xiang, J. Wu, and J. Le, "AVPMIR: Adaptive verifiable privacy-preserving medical image retrieval," *IEEE Trans. Depend. Secure Comput.*, vol. 21, no. 5, pp. 4637–4651, Sep./Oct. 2024.
- [11] P. Drozdzowski, F. Stockhardt, C. Rathgeb, D. Osorio-Roig, and C. Busch, "Feature fusion methods for indexing and retrieval of biometric data: Application to face recognition with privacy protection," *IEEE Access*, vol. 9, pp. 139361–139378, 2021.
- [12] C. Liu, T. Zhu, J. Zhang, and W. Zhou, "Privacy intelligence: A survey on image privacy in online social networks," *ACM Comput. Surv.*, vol. 55, no. 8, pp. 1–35, 2022.
- [13] W. K. Wong, D. W.-l. Cheung, B. Kao, and N. Mamoulis, "Secure kNN computation on encrypted databases," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, 2009, pp. 139–152.
- [14] C. Mao, Z. Shen, K. Chen, Y. Liu, Q. Meng, and F. Wang, "DCIRM: Dynamic and controllable image retrieval scheme in multi-owner multi-user settings," *IEEE Trans. Services Comput.*, vol. 17, no. 4, pp. 1435–1448, Jul./Aug. 2024.
- [15] S. Khan, H. Abbas, and W. Iqbal, "Verifiable privacy-preserving image retrieval in multi-owner multi-user settings," *IEEE Trans. Emerg. Topics Comput. Intell.*, vol. 8, no. 2, pp. 1640–1655, Apr. 2024.
- [16] X. Wang, J. Ma, X. Liu, and Y. Miao, "Search in my way: Practical outsourced image retrieval framework supporting unshared key," in *Proc. IEEE Conf. Comput. Commun.*, 2019, pp. 2485–2493.
- [17] Y. Li, J. Ma, Y. Miao, L. Liu, X. Liu, and K.-K. R. Choo, "Secure and verifiable multikey image search in cloud-assisted edge computing," *IEEE Trans. Ind. Informat.*, vol. 17, no. 8, pp. 5348–5359, Aug. 2021.
- [18] T. Yang et al., "Secure and traceable multikey image retrieval in cloud-assisted Internet of Things," *IEEE Internet Things J.*, vol. 11, no. 24, pp. 40875–40887, Dec. 2024.
- [19] Z. Tang, H. Fan, X. Gu, Y. Li, B. Li, and X. Wang, "ELSEIR: A privacy-preserving large-scale image retrieval framework for outsourced data sharing," in *Proc. Int. Conf. Multimedia Retrieval*, 2024, pp. 488–496.
- [20] C. Guo, S. Su, K.-K. R. Choo, and X. Tang, "A fast nearest neighbor search scheme over outsourced encrypted medical images," *IEEE Trans. Ind. Informat.*, vol. 17, no. 1, pp. 514–523, Jan. 2021.
- [21] T. Yang et al., "MU-TEIR: Traceable encrypted image retrieval in the multi-user setting," *IEEE Trans. Services Comput.*, vol. 16, no. 2, pp. 1282–1295, Mar./Apr. 2022.
- [22] L. Zhang et al., "PIC: Enable large-scale privacy preserving content-based image search on cloud," *IEEE Trans. Parallel Distrib. Syst.*, vol. 28, no. 11, pp. 3258–3271, Nov. 2017.
- [23] L. Chen et al., "Efficient and secure content-based image retrieval in cloud-assisted Internet of Things," *IEEE Internet Things J.*, vol. 12, no. 5, pp. 6001–6013, Mar. 2025.
- [24] M. Li, Y. Zhu, R. Du, and C. Jia, "Verifiable encrypted image retrieval with reversible data hiding in cloud environment," *IEEE Trans. Cloud Comput.*, vol. 13, no. 1, pp. 397–410, Jan.–Mar. 2025.
- [25] Y. Li et al., "Traceable and controllable encrypted cloud image search in multi-user settings," *IEEE Trans. Cloud Comput.*, vol. 10, no. 4, pp. 2936–2948, Oct.–Dec. 2020.
- [26] Q. Tong et al., "VFIRM: Verifiable fine-grained encrypted image retrieval in multi-owner multi-user settings," *IEEE Trans. Services Comput.*, vol. 15, no. 6, pp. 3606–3619, Nov./Dec. 2022.
- [27] Y. Duan, Y. Li, L. Lu, and Y. Ding, "A faster outsourced medical image retrieval scheme with privacy preservation," *J. Syst. Archit.*, vol. 122, Jan. 2022, Art. no. 102356.
- [28] S. Li, L. Wu, W. Meng, Z. Xu, C. Qin, and H. Wang, "DVPPIR: Privacy-preserving image retrieval based on DCNN and VHE," *Neural Comput. Appl.*, vol. 34, no. 17, pp. 14355–14371, 2022.
- [29] S. Zhang, Q. Liu, T. Wang, W. Liang, K.-C. Li, and G. Wang, "FSAIR: Fine-grained secure approximate image retrieval for mobile cloud computing," *IEEE Internet Things J.*, vol. 11, no. 13, pp. 23297–23308, Jul. 2024.
- [30] R. Li, A. X. Liu, Y. Liu, H. Xu, and H. Yuan, "Insecurity and hardness of nearest neighbor queries over encrypted data," in *Proc. IEEE Int. Conf. Data Eng.*, 2019, pp. 1614–1617.
- [31] X. Shen, X. Li, H. Yin, C. Cao, and L. Zhang, "Lattice-based multi-authority ciphertext-policy attribute-based searchable encryption with attribute revocation for cloud storage," *Comput. Netw.*, vol. 250, Aug. 2024, Art. no. 110559.
- [32] L. Chen, S. Xu, H. Zhang, and J. Weng, "Fair and exculpable attribute-based searchable encryption with revocation and verifiable outsourced decryption using smart contract," *IEEE Internet Things J.*, vol. 12, no. 4, pp. 4302–4317, Feb. 2025.

- [33] U. S. Varri and S. K. Pasupuleti, "Traceable and revocable multi-authority attribute-based keyword search for cloud storage," *J. Syst. Archit.*, vol. 132, Nov. 2022, Art. no. 102745.
- [34] U. S. Varri, S. Kasani, S. K. Pasupuleti, and K. Kadambari, "FELT-ABKS: Fog-enabled lightweight traceable attribute-based keyword search over encrypted data," *IEEE Internet Things J.*, vol. 9, no. 10, pp. 7559–7571, May 2022.
- [35] A. Beimel, "Secure schemes for secret sharing and key distribution," Ph.D. dissertation, Technion Israel Inst. Technol., Haifa, Israel, 1996.
- [36] J. Ning, X. Dong, Z. Cao, L. Wei, and X. Lin, "White-box traceable ciphertext-policy attribute-based encryption supporting flexible attributes," *IEEE Trans. Inf. Forensics Security*, vol. 10, pp. 1274–1288, 2015.
- [37] Y. Zheng, R. Lu, S. Zhang, J. Shao, and H. Zhu, "Achieving practical and privacy-preserving kNN query over encrypted data," *IEEE Trans. Depend. Secure Comput.*, vol. 21, no. 6, pp. 5479–5492, Dec. 2024.
- [38] G. Griffin, A. Holub, and P. Perona, *Caltech-256 Object Category Dataset*, California Inst. Technol., Pasadena, CA, USA, 2007.

Yingying Li (Member, IEEE) received the B.S. degree in mathematics from Shaanxi Normal University, Xi'an, China, in 2017, and the Ph.D. degree from the School of Cyber Engineering, Xidian University, Xi'an, in 2022.

She is currently a Lecture with the School of Mathematics, Hangzhou Normal University, Hangzhou, China. Her research interests include data security and applied cryptography.

Feng Li received the Ph.D. degree from the School of Cyber Engineering, Xidian University, Xi'an, China, in 2024.

He is currently a Postdoctoral Fellow with Singapore Management University, Singapore. His research interests include information security and applied cryptography.

Gaopan Hou received the B.S and Ph.D. degrees in mathematics and cyberspace security from Xidian University, Xi'an, China, in 2015 and 2021, respectively.

He is currently a Postdoctoral Fellow with the Hangzhou Institute of Technology, Xidian University. His research interests include applied cryptography and data security.

Yu Guan received the B.E. degree in computer science and technology from Beijing University of Civil Engineering and Architecture, Beijing, China, in 2022. He is currently pursuing the master's degree with Hangzhou Institute of Technology, Xidian University, Xi'an, China.

His research interests include federated learning and data security.

Zhiqian Liu received the B.S. degree from the School of Science, and the Ph.D. degree from the School of Computer Science and Technology, Xidian University, Xi'an, China, in 2012 and 2017, respectively.

He is currently a Full Professor with the College of Cyber Security, Jinan University, Guangzhou, China. His current research focuses on security, trust, privacy, and intelligence in vehicular networks.

Prof. Liu currently serves as the Area Editor, Associate Editor, or Academic Editor of more than ten SCI-index journals, such as IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, IEEE INTERNET OF THINGS JOURNAL, *Information Fusion*, IEEE NETWORK, *Computer Networks*, and *Journal of Systems Architecture*. His homepage is <https://www.zqliu.com>.

Qi Xie received the Ph.D. degree in applied mathematics from Zhejiang University, Hangzhou, China, in 2005.

He was a Visiting Scholar with the Department of Computer Science, University of Birmingham, Birmingham, U.K., from 2009 to 2010, and a Visiting Scholar with the Department of Computer Science, The City University of Hong Kong, Hong Kong, in 2012. He is a Professor with Hangzhou Normal University, Hangzhou, and the Director of the Key Laboratory of Cryptography of Zhejiang Province, Hangzhou Normal University. He has published over 100 research papers in international journals and conferences, such as IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY and IEEE INTERNET OF THINGS JOURNAL. His research interests include applied cryptography, including digital signatures, authentication, and key agreement protocols.

Prof. Xie served as a reviewer for over 40 international journals and served as the General Co-Chair for ISPEC2012 and ACM ASIACCS2013.