Privacy-Preserving Image Retrieval Scheme Using Combined Features in Cloud Computing

Jing Liang*, Yuxuan Wang*, Tingting Song*, Ce Zheng† and Peiya Li‡
* College of Cyber Security, Jinan University, Guangzhou, China
E-mail: liangjing99@stu.jnu.edu.cn, wyyyxx02@163.com, tingtingsong@jnu.edu.cn
† Department of Network Intelligence, Peng Cheng Laboratory, Shenzhen, China
E-mail: zhengc@pcl.ac.cn
‡ College of Cyber Security, Jinan University, Guangzhou, China
E-mail: lpy0303@jnu.edu.cn

***Abstract*—With the development of cloud computing, more and more resource-constrained data owners tend to store their images in the cloud. The existing image retrieval schemes face challenges such as suboptimal retrieval accuracy, low retrieval efficiency and security. Therefore, in this article we propose a privacy-preserving image retrieval scheme based on feature combination. Firstly, we extract the features derived from convolutional neural network and deep hash model for the images and then fuse them into new feature descriptors. Subsequently, we design a secure feature encryption method based on differential privacy and secure K-nearest neighbors to generate an encrypted hierarchical indexing structure. Experimental evaluation shows that the proposed scheme achieves sub-linear retrieval without privacy leakage under low storage resources.**

I. Introduction

In recent years, with the rapid advancement of mobile cloud technology, resource-constrained individuals or institutions are increasingly inclined to upload image data to the cloud for storage [1]. Currently, the demand for retrieving specific images accurately from large image repositories has grown significantly in practical application domains such as smart healthcare and intelligent transportation, etc. This trend further promotes the extensive sharing and dissemination of multimedia data, particularly image information [2]. Despite the convenience of cloud-based image storage, security concerns remain a major barrier to widespread adoption. Images often contain sensitive information, such as facial features and surveillance content, that risks privacy leakage if uploaded in plain images. Encryption can mitigate these risks, but significantly affects the searchability of traditional images [3], [4]. As a result, research has increasingly focused on enabling accurate and efficient content-based image retrieval while preserving user privacy.

In cloud environments, constrained computational resources at the terminal end pose challenges for secure and efficient image retrieval. Although traditional Content-Based Image Retrieval (CBIR) methods balance accuracy and privacy protection, they often overlook practical requirements, such as computational performance and storage costs. Therefore, current schemes continue to focus on several challenges as follows [5].

Low retrieval accuracy: To enhance retrieval security, some CBIR schemes encrypt images before extracting features from the ciphertext images. While this improves privacy, the limited information in encrypted images leads to weak feature representation and reduced retrieval accuracy. For example, approaches like probabilistic encryption [6], Histograms of Oriented Gradients (HOGs) [7], Local Binary Patterns (LBP) [8], and Bag-of-Encrypted-Words (BOEWs) [9], while employing diverse encryption and feature extraction strategies, features with weak representational capacity. To enhance accuracy, researchers have increasingly turned to Convolutional Neural Network (CNN) for extracting deep features. Relevant work such as [10], [11], demonstrates that CNN can extract more discriminative features, significantly outperforming traditional schemes.

Low retrieval efficiency: While feature encryption enhances image privacy protection, it simultaneously impacts retrieval efficiency. For instance, Homomorphic Encryption (HE) [12] and Secure Multi-Party Computation (SMC) [13], despite offering strong security guarantees, often suffer from high computational overhead and ciphertext expansion issues, leading to inefficient retrieval. To mitigate this problem, Li et al. [14] proposed a semi-hash index structure, which effectively improves system performance. Yuan et al. [15] proposed a k-Nearest-Neighbor (kNN) based privacy-preserving content-based image retrieval scheme and integrates K-means clustering to accelerate the retrieval process. Similarly, [4] employs K-means clustering to construct a hierarchical index tree, outperforming traditional linear indexing methods [16] in efficiency.

High storage cost: Numerous CBIR methods rely on high-dimensional feature representations to enhance image retrieval precision. While this enhances discriminative power, it concomitantly increases storage overhead for feature indexing substantially. Early encrypted image retrieval schemes employed Order-Preserving Encryption (OPE) and MinHash [17], constructing high-dimensional features using extensive visual words. Methods such as Fisher Vectors [15] also utilize

‡Corresponding author

979-8-3315-7206-8/25/$31.00

relatively high-dimensional representations. The experimental results in [4] demonstrate that the reduction in dimensionality inevitably results in a degraded retrieval precision. Consequently, although high-dimensional features improve accuracy, they impose significant spatial storage burdens in large-scale data retrieval scenarios, presenting a critical trade-off requiring optimization.

To address the above challenges, this paper proposes a Privacy-Preserving Image Retrieval scheme using Combined Features (PPIRCF). We first extract the features derived from CNN and deep hash model for the images and then fuse them into new feature descriptors. Then, we adopt the clustering algorithm to design the hierarchical index structure and design the encryption algorithm based on differential privacy and Learning With Errors (LWE) [18]. The main technical contributions are as follows:

1) A novel combined feature mechanism: For the first time, the high-level semantic features extracted by CNN are effectively fused with the compact binary features generated by deep hashing. While maintaining high retrieval accuracy, it significantly reduces the feature storage and computational overhead.

2) A secure and efficient hierarchical index structure and encryption method: A hierarchical index is constructed through clustering algorithms, and LWE-based kNN encryption and differential privacy technology are integrated to provide targeted protection for CNN features and binary features respectively. While achieving rapid retrieval in the ciphertext domain, it supports sublinear query efficiency.

3) A systematic privacy protection solution covering the entire process: It achieves comprehensive protection for image content, feature indexes, and user queries within the same framework, capable of resisting various attack models such as known ciphertext and known background, and has high practicality and security.

The rest of the paper is organized as follows. In Section 2, our proposed scheme and its detailed procedure, threat models are presented. Section 3 analyzes security and experimental results. We conclude this paper in Section 4.

II. Proposed Scheme

A. System Model

Our scheme contains four entities, namely authentication center, image owner, cloud server and users, which are shown in Fig. 1. The authentication center is a trusted third party that authorizes users, generates keys, and distributes keys to the image owners and search users. The image owner extracts the image features to construct an index, encrypts the index and the images, then uploads them to the cloud server. Users build trapdoors and submit them to the cloud server. After receiving the search results from the cloud server, the search users decrypt results with the authorized keys. Cloud server stores encrypted index and images. When receiving queries, the cloud server implements a retrieval algorithm and returns top-k most similar images to search users.

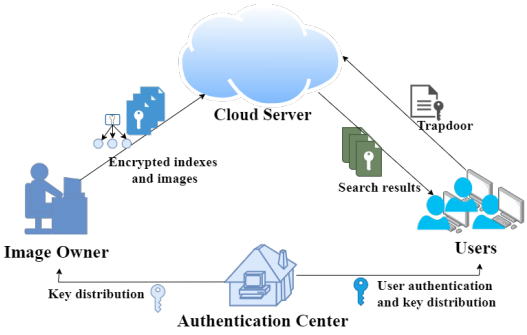

Fig. 1. System model of our scheme.

B. Threat Model

In our system, the cloud server is considered to be honest but curious. That is, the cloud server will follow our protocol to perform the service provider role correctly, but it may tamper or forge sensitive information from encrypted images, index, or trapdoor. The authentication center, image owner and users are always trusted. Based on information available to the cloud server, we consider the following threat models.

Known Ciphertext Attack Model. The cloud server only has access to all encrypted images and indexes and trapdoor. In particular, we aim at preventing the cloud server from learning valuable information related to image content, encrypted indexes, trapdoor and encrypted search results.

Known Background Attack Model. In this stronger threat model, the cloud server possesses all information consistent with the *Known Ciphertext Attack Model*. Moreover, the cloud server will obtain additional information, such as some plaintexts, which would be used to infer more plaintexts from inner products calculating results.

C. Detailed Constructions

Our scheme construction consists 6 algorithms: **KeyGen**, **IndexGen**, **ImgEnc**, **TrapGen**, **Search** and **ImgDec**. In the **KeyGen** algorithm, the authentication center generates keys to set up the system. Then, by using the **IndexGen** algorithm, the image owner builds a searchable encrypted index for image search. For privacy protection purpose, the image owner utilizes the **ImgEnc** algorithm to encrypt all images in the database. The **TrapGen** algorithm allows users to generate secure image search requests. On receiving users' request, the cloud server runs the **Search** algorithm to obtain search results and sends them to the users. After receiving ciphertexts for similar images, the users use **ImgDec** algorithm to decrypt them and obtain final similar images. We now introduce the detail of each algorithm in our scheme.

KeyGen: With the security parameter λ, the authentication center generates the feature vector encryption key $k_f = \{\gamma, \pi, \epsilon, M, M^{-1}\}$. Specifically, $\gamma \in \mathbb{Z}_{p_1}$ is a random number and p_1 is an integers that define the range of number, π is a random permutation, ϵ is the privacy budget parameter, M is a $(d_1 + \alpha) \times (d_1 + \alpha)$-dimensional random invertible matrix,

M^{-1} is inverse of M, d_1 and α are two integers. In order to limit the parameter range later, we set the integer p_2, where $p_1 \gg p_2$. For the set of N images $\{m_i\}_{i=1}^N$, the authentication center generates the image encryption key $k_m = \{k_{m,i}\}_{i=1}^N$ using symmetric encryption method AES. The authentication center publishs $\{p_1, p_2\}$ and sends $\{k_m, k_f\}$ to image owner and search users.

IndexGen: The index generation algorithm consists of four steps, the first step is to extract feature of each image in $\{m_i\}_{i=1}^N$ with CNN and deep hash model, the second step is to perform feature combination, the third step is to generate the hierarchical index, and the fourth step is to encrypt the index.

1) Feature extraction. For each image m_i in the set $\{m_i\}_{i=1}^N$, after extracting features using the CNN model [19], we adopt the PCA [20] technology for dimension reduction processing to obtain feature $H_i^{(1)}$, and use the deep hash model [21] to extract feature $H_i^{(2)}$. The powerful semantic representation ability of CNN features can facilitate retrieval. Deep hash features are stored in binary form, saving the space cost of features and facilitating the construction of large-scale image retrieval.

2) Feature combination. For each image m_i in the set $\{m_i\}_{i=1}^N$, the combined feature $f_i = H_i^{(1)} \| H_i^{(2)} = \{h_{i,1}^{(1)}, h_{i,2}^{(1)}, ..., h_{i,d_1}^{(1)}, h_{i,1}^{(2)}, h_{i,2}^{(2)}, ..., h_{i,d_2}^{(2)}\}$, where d_1 is the dimension of CNN feature $H_i^{(1)}$, d_2 is the dimension of deep hash feature $H_i^{(2)}$ and d_1, d_2 are both integers.

3) Hierarchical index generation. Taking $H_i^{(1)}$ as the input, the K-means algorithm is adopted to cluster the images. Since $H_i^{(1)}$ contains rich detailed information, after using K-means, we can obtain K clustering centers $\{C_l\}_{l=1}^K$. The index is constructed by two layers, where $C_l = \{c_{l,1}, c_{l,2}, ..., c_{l,d_1}\}$ is the top-level clustering level. When the Euclidean distance between $H_i^{(1)}$ of an image m_i and the center C_l of cluster l is the smallest, this image will be placed in the cluster represented by that C_l. In each cluster, several images are organized with their feature vectors $H_i^{(2)} = \{h_{i,1}^{(2)}, h_{i,2}^{(2)}, ..., h_{i,d_2}^{(2)}\}$ as the second layer of the hierarchical index.

4) Index encryption. Since CNN features are usually real number vectors, while hash features are binary, we adopt different encryption algorithms to protect C_l and $H_i^{(2)}$. Since the clustering center C_l are generated based on CNN features, they are encrypted with LWE-based kNN encryption, which is a commonly used encryption method for CNN features. Image owner first extends C_l to $C_l' = \{c_{l,1}, c_{l,2}, ..., c_{l,d_1}, \|C_l\|, \theta\}$, where $\|C_l\|$ represents the modules length of vector C_l and $\theta \in \mathbb{Z}_{p_2}^{\alpha-1}$ is a random vector. Then, image owner encrypts C_l' as

$$\hat{C}_l = (\gamma \cdot C_l' + \varepsilon_i) \cdot M, \tag{1}$$

where $\varepsilon_i \in \mathbb{Z}^{d_1+\alpha}$ is a random integer noise vector and $2|max(\varepsilon_i)| \ll \gamma$, $|max(\varepsilon_i)|$ denotes the maximum absolute value of ε_i's elements.

Based on the characteristic that the deep hash feature $H_i^{(2)}$ is binary, we adopt the encryption method based on differential privacy (DP) [22]. Give a function F: $D \to \mathcal{R}$ over a dataset D, the mechanism $\mathcal{M}$ satisfies ϵ-DP: $\mathcal{M}(D) = F(D) + Lap(\frac{\Delta F}{\epsilon})$, where ΔF is the sensitivity of the function F and Lap is represented as the Laplace mechanism [23]. In our specific setup, we add Laplace disturbance to $H_i^{(2)}$ with a sensitivity of $\Delta F = 2$, as $H_i^{(2)}$ is a vector containing only 1 and -1. We first use the random permutation π to shuffle the component elements of $H_i^{(2)}$ to obtain $H_i^{(2)\prime}$. Then, the perturbed result $H_i^{(2)\prime}$ is processed as follows:

$$\hat{H}_i^{(2)} = \{sgn(H_i^{(2)\prime} + Lap(\frac{2}{\epsilon}))\}, \tag{2}$$

where $sgn()$ is a sign function, it outputs 1 if the input value is greater than 0 and outputs -1 otherwise. It is known that $H_i^{(2)\prime} + Lap(\frac{2}{\epsilon})$ provides ϵ-DP. Due to the post-processing properties of DP, $\hat{H}_i^{(2)}$ also satisfies ϵ-DP.

ImgEnc: In order to protect the privacy of images from the cloud, the image owner will encrypt all images. The image owner utilizes the conventional encryption method AES and the key k_m to encrypt image set $\{m_i\}_{i=1}^N$, then outputs encrypted image set $\{c_i\}_{i=1}^N$. The image owner sends the encrypted images and indexes to cloud server.

TrapGen: To perform an encrypted image search, the user generates a query trapdoor for the queried image m_q. The user first extracts feature vectors $f_q = (H_q^{(1)} \| H_q^{(2)})$ and encrypts $H_q^{(2)}$ using the same method as $H_i^{(2)}$ to obtain $\hat{H}_q^{(2)}$. For $H_q^{(1)}$, it is first expanded to obtain $H_q^{(1)\prime} = \{-2r_q \cdot H_q^{(1)}, 1, \beta\}$, where $\beta \in \mathbb{Z}_{p_2}^{\alpha-1}$, $r_q \in \mathbb{Z}_{p_2}$ and positive in each query. After that, $H_q^{(1)\prime}$ is encrypted as

$$\hat{H}_q^{(1)} = M^{-1} \cdot (\gamma \cdot H_q^{(1)\prime\top} + \varepsilon_q^\top), \tag{3}$$

where $\varepsilon_q \in \mathbb{Z}_{p_2}^{d_1+\alpha}$ is a random integer noise vector that is different in each query, $H_q^{(1)\prime\top}$ and $\varepsilon_q^\top$ are column vectors of $H_q^{(1)\prime}$ and ε_q, respectively. Finally, the encrypted trapdoor will be formally denoted as $\mathcal{TD} = \{\hat{H}_q^{(1)} \| \hat{H}_q^{(2)}\}$ and sent to the cloud for retrieving similar images.

Search: On receiving the search request from users, the cloud server first finds the clustering center to which the queried image belongs according to its $\hat{H}_q^{(1)}$, and then finds the corresponding image in this cluster and returns the top-k most similar results. The detailed process is described as follows.

First, cloud server calculates the inner product of $\hat{C}_l$ and $\hat{H}_q^{(1)}$:

$$\begin{aligned} EnD_{l,q} &= \lceil \frac{\hat{C}_l \cdot \hat{H}_q^{(1)}}{\gamma^2} \rfloor_{p_1} \\ &= r_q(\|C_l - H_q^{(1)}\|^2 - \|H_q^{(1)}\|^2) + \sum_{i=1}^{\alpha-1} \theta_i \beta_i, \end{aligned} \tag{4}$$

where $\lceil \rfloor_{p_1}$ is the nearest integer with modulus p_1. From the calculation result we can see that the inner product of the encrypted feature vectors between the query and stored image approximates to the Euclidean distance of their plaintexts. This allows the cloud server to rank the results. Cloud server

performs an ascending sort based on the scores $EnD_{l,q}$ and enters the specific cluster C_l.

Second, after finding the specific cluster, the cloud server calculates the similarity between $\hat{H}_i^{(2)}$ and $\hat{H}_q^{(2)}$ as $EnH_{i,q} = \sum_{j=1}^{d_1}(\hat{H}_{i,j}^{(2)} \oplus \hat{H}_{q,j}^{(2)})$. The smaller the distance $EnH_{i,q}$, the more similar the retrieved image is to the queried image. After the computation and ranking, the cloud server sends the top-k most similar encrypted images back to users.

ImgDec: When users get the search results, they use the pre-assigned symmetric decryption key to get the original plaintext image.

III. Security and Performance Analysis

To illustrate that our scheme is secure and feasible in practice, we present the detailed security and performance analysis separately.

A. Security Analysis

In this section, we analysis the security from image privacy, index and trapdoor privacy.

1) Image Privacy. In our scheme, the image set $m_i \in \{m_i\}_{i=1}^N$ is encrypted by the conventional encryption method AES with key k_m. Since both the image owner and users are trusted and the key k_m is secure, the content of the image set can be well protected.

2) Index and Query Privacy. We first define the Learning With Errors (LWE) problem [18], then prove the index and query privacy under *known ciphertext attack model* and *known background attack model* given in Section II-B.

Define 1 (LWE problem): Given polynomial many samples of ($x_i \in \mathbb{Z}_{p_1}^{d+\alpha+2\rho+2}$) with $y_i = f \cdot x_i^\top + \varepsilon_i$, where the error term $\varepsilon_i \in \mathbb{Z}_{p_1}$ is drawn from some probability distribution, it is computationally infeasible to recover the vector f with non-negligible probability.

Theorem 1: Our scheme can protect the index and query privacy under the known ciphertext attack model.

Proof: In our scheme, each clustering center C_l in the index is encrypted as $\hat{C}_l = (\gamma \cdot C_l' + \varepsilon_i) \cdot M$.

Since C_l' and ε_i are two vectors, we can convert $\hat{C}_l$ into $d_1 + \alpha$ dot products of $d_1 + \alpha$-dimensional vectors as $\hat{C}_l(k) = \gamma \cdot C_l' \cdot M(k) + \varepsilon_i \cdot M(k)$, where $1 \le k \le d_1 + \alpha$, $\hat{C}_l(k)$ is the kth element of $\hat{C}_l$, and $M(k)$ is the kth column of M. We denote $\gamma \cdot M(k)$ as $\widetilde{M}(k)$ and $\varepsilon_i \cdot M(k)$ as $\widetilde{\varepsilon_i}$, then we have $d_1 + \alpha$ samples ($\widetilde{M}(k)$, $\hat{C}_l(k)$) with $\hat{C}_l(k) = C_l' \cdot \widetilde{M}(k) + \widetilde{\varepsilon_i}$.

Therefore, recovering C_l' from the encryption vector $\hat{C}_l(k)$ becomes the LWE problem given in *Definition 1*. For the polynomial-time adversary, it is computationally infeasible to solve the LWE problem. Moreover, matrix M and $\widetilde{M}$ are not available to the adversary, thus recovering C_l' from $\hat{C}_l$ becomes more difficult than solving the LWE problem.

For each image feature $H_i^{(2)}$, the order of its elements is all scrambled using random permutation encryption. To break this encryption, one has to explore all $d!$ permutations, which leads to exponential computational costs proportional to $O(d!)$. NIST [24] recommends a security strength of over 112 bits as being secure enough. We choose d values like 32, 48, 64, 128 for experiments. With d = 32 (the smallest), the security strength is around $log_2(32!) \approx 117$ bits, meaning even the lowest parameter setting surpasses the recommended security strength and ensures the safety of our scheme. Moreover, although the permutation's security strength already meets practical needs, we further boost security by adding differential privacy technique.

With regard to query ciphertext $\hat{f}_q = \{\hat{H}_q^{(1)} \| \hat{H}_q^{(2)}\}$, its security can be proved by using a method similar to that of the index. Due to space limitations, it will not be elaborated further. In summary, the privacy of feature vectors in index and queries can be protected under *known ciphertext attack model* is proved.

Theorem 2: Our scheme can protect the index and query privacy under the known background attack model.

Proof: In the *known background attack model*, the adversary may attempt a linear analysis attack to recover the plaintexts from the results of inner product calculations. Specifically, given a query vector $H_q^{(1)}$ and any two index vectors C_{la}, C_{lb}, the adversary can construct an equation as $EnD_{a,b} = r_q(\|C_{la} - f_q\|^2 - \|C_{lb} - f_q\|^2)$. The linear analysis attack is that if the adversary can obtain more than $2d$ query vectors, it can construct $2d$ for $2d$ unknown index vectors (C_{la}, C_{lb}) and recover C_{la}, C_{lb} from them. Since the random number r_q in each query is different, this kind of attack is not feasible in our scheme. Therefore, our scheme can protect the privacy of index and query under the *known background attack model*.

B. Performance Analysis

The performance of our scheme construction is evaluated from two aspects of accuracy and efficiency. We perform a series of experimental evaluations using Python 3.11 on the Windows 11 operating system with Intel Core (TM) i7-12700H CPU. All the images are taken from the Corel-10K dataset, which contains 100 categories of images and each category contains 100 similar images. In order to illustrate the feasibility and efficiency of our scheme, we replicated the code from three state-of-art and most relevant schemes, which are DVREI [4], FGIR [10] and TBIR [16]. Both FGIR [10] and DVREI [4] construct hierarchical indexes, while TBIR [16] employs a linear image index. We compare our PPIRCF with these three schemes in terms of accuracy, index construction efficiency, trapdoor generation efficiency and retrieval efficiency.

1) Accuracy. The measure of accuracy evaluation is precision at top-k (P@k), defined as $P@k = \frac{k_c}{k}$. Here, k is the number of returned similar images, and k_c is the number of images in the returned k similar images that belong to the same category as the query image.

In our PPIRCF scheme, the dimensionality of the combined features will affect the accuracy. Higher dimensionality may enhance retrieval accuracy, yet it increases computational complexity and index storage size. Therefore, we need to find the optimal feature vector dimensionality to obtain the optimal solution for vector dimensionality that ensures high retrieval

TABLE I
RETRIEVAL ACCURACY OF $H^{(1)} + H^{(2)}$ IN DIFFERENT DIMENSIONS

$H^{(1)} + H^{(2)}$ \ P@k	10	20	30	40	50
32+32	0.825	0.815	0.806	0.784	0.771
32+64	0.841	0.843	0.819	0.795	0.776
32+128	0.852	0.842	0.825	0.804	0.774
64+32	0.883	0.866	0.848	0.829	0.813
64+64	0.905	0.870	0.839	0.819	0.807
64+128	0.912	0.875	0.843	0.822	0.808
128+32	0.908	0.887	0.870	0.845	0.825
128+64	0.907	0.885	0.871	0.848	0.830
128+128	0.932	0.897	0.872	0.853	0.838

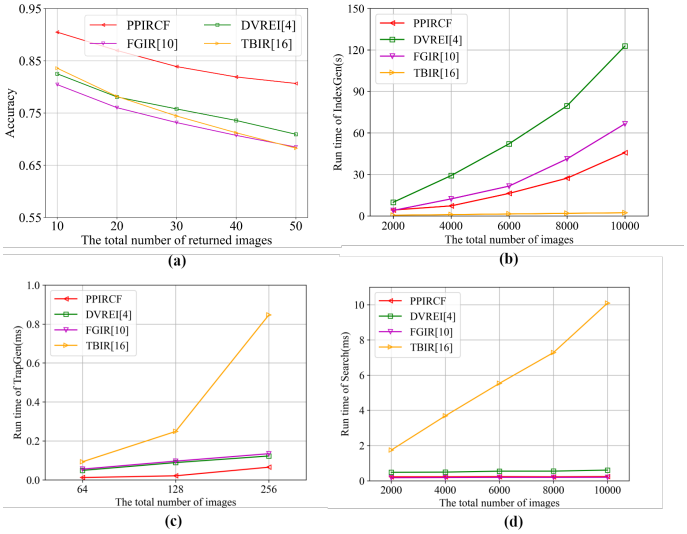

Fig. 2. (a) Retrival accuracy in different schemes. (b) Run time IndexGen. (d)Run time of TrapGen. (d) Run time of Search.

accuracy and efficiency within limited storage space. Table I shows the P@k values of the $H^{(1)} + H^{(2)}$ in different dimensions when different numbers of top-k images are returned in Corel-10K. In order to facilitate the comparison with other schemes in the same dimension, we select a 64-dimensional vector as the feature vector $H^{(1)}$ and 64-dimensional vector as the $H^{(2)}$ in the subsequent experiments.

In Fig. 2(a), we calculate the average accuracy of 1000 image retrieval using (64+64)-dimensional combined features and 128-dimensional features of other schemes when the number of returned images is set from 10 to 50. It can be observed that our PPIRCF outperforms other schemes in terms of retrieval accuracy. This is attributed to the fact that after we use CNN feature clustering, the hash features in the collection is relatively more regular and has more commonalities, thereby improving the overall retrieval accuracy.

2) Efficiency. Fig. 2(b) depicts the computational overhead of index generation for PPIRCF and other schemes when the total number of images ranges from 2000 to 10000. As FGIR [10] and DVREI [4] employ the index based on the tree structure, it take more time compared to the linear index structure of TBIR [16]. Our PPIRCF only performs the clustering operation once, and the time for generating the index is lower than that of the multi-layer index schemes FGIR [10] and DVREI [4]. This indicates the advantages of our scheme in the construction of hierarchical indexes.

TABLE II
THE STORAGE OVERHEAD IN DIFFERENT SCHEMES

Scheme	PPIRCF	FGIR [10]	DVREI [4]	TBIR [16]
Index($\times$MB)	**1.28**	2.25	2.313	137.69
Trapdoor($\times$KB)	**2**	3	3	139

In Fig. 2(c), when calculating the time cost of the **TrapGen** algorithm, the features of 32+32, 64+64 and 128+128 dimensions are respectively compared with other schemes of 64, 128 and 256 dimensions. It can be seen that our time cost is the lowest in terms of the generation of trapdoor. This is because we need to encrypt the k cluster centers after dimensionality reduction and all the binary encoded hash features, rather than the CNN features of the entire database.

In Fig. 2(d), we display the image retrieval time in **Search** algorithm, returning the top 10 most similar images. Compared with linear retrieval scheme TBIR [16], our retrieval time is significantly reduced. By using the clustering algorithm to implement sub-linear retrieval, it is more suitable for real-time processing. Compared with the FGIR [10] and DVREI [4] schemes based on hierarchical index, our scheme takes less time during retrieval. This is attributed to our combination of clustering algorithm and combined features, which reduces the computational complexity of retrieval and improves the retrieval speed.

Table II tests the storage burden of indexes and trapdoors for privacy-preserving image retrieval for 1000 images in different schemes. Compared with other schemes, our scheme saves storage space as shown in Table II and achieves better retrieval accuracy as shown in Fig. 2(a), which indicates that the proposed scheme is suitable for efficient retrieval.

IV. CONCLUSIONS

In this paper, we propose a privacy-preserving image retrieval scheme using combined features in cloud computing. We first construct the combined features by using CNN features and deep hash features. Then, a two-layer index structure is constructed using the clustering algorithm, and an encrypted searchable index is generated using differential privacy and LWE-based algorithms. Both security and performance analysis demonstrated that our proposed scheme could guarantee the security of outsourced data and achieve efficient search in cloud.

ACKNOWLEDGMENT

This work was supported in part by the National Key R&D Program of China under Grant 2022YFB3103500, in part by the National Natural Science Foundation of China under Grants 62302195, and in part by the GuangDong Basic and Applied Basic Research Foundation under Grant 2022A1515011960.

References

[1] 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," *Information Sciences*, vol. 387, pp. 195–204, 2017.

[2] L. B. Furstenau, Y. P. R. Rodrigues, M. K. Sott, *et al.*, "Internet of things: Conceptual network structure, main challenges and future directions," *Digital Communications and Networks*, vol. 9, no. 3, pp. 677–687, 2023.

[3] J. Liang, P. Li, Z. Liu, and H. He, "Privacy-preserving clustering-based image retrieval in cloud-assisted internet of things," *IEEE INTERNET OF THINGS JOURNAL*, vol. 12, no. 12, pp. 20 484–20 497, Jun. 2025.

[4] Y. Li, J. Ma, Y. Miao, *et al.*, "Dvrei: Dynamic verifiable retrieval over encrypted images," *IEEE Transactions on Computers*, vol. 71, no. 8, pp. 1755–1769, 2022.

[5] X. Li, W. Lei, W. Tang, Y. Wang, X. Yang, and X. Liao, "Segmented hash-based privacy-preserving image retrieval scheme in cloud-assisted iot," *IEEE Internet of Things Journal*, 2024.

[6] B. Ferreira, J. Rodrigues, J. Leitão, and H. Domingos, "Privacy-preserving content-based image retrieval in the cloud," in *2015 IEEE 34th Symposium on Reliable Distributed Systems (SRDS)*, 2015, pp. 11–20.

[7] M. Kitayama and H. Kiya, "Hog feature extraction from encrypted images for privacy-preserving machine learning," in *2019 IEEE International Conference on Consumer Electronics - Asia (ICCE-Asia)*, 2019, pp. 80–82.

[8] Z. Xia, L. Wang, J. Tang, N. N. Xiong, and J. Weng, "A privacy-preserving image retrieval scheme using secure local binary pattern in cloud computing," *IEEE Transactions on Network Science and Engineering*, vol. 8, no. 1, pp. 318–330, 2021.

[9] Z. Xia, L. Jiang, D. Liu, L. Lu, and B. Jeon, "Boew: A content-based image retrieval scheme using bag-of-encrypted-words in cloud computing," *IEEE Transactions on Services Computing*, vol. 15, no. 1, pp. 202–214, 2022.

[10] J. Liang, L. Wang, and P. Li, "Fine-grained privacy-preserving image retrieval in cloud environment," in *2024 Asia Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, IEEE, 2024, pp. 1–6.

[11] Q. Feng, P. Li, Z. Lu, *et al.*, "Dhan: Encrypted jpeg image retrieval via dct histograms-based attention networks," *Applied Soft Computing*, vol. 133, p. 109 935, 2023.

[12] J.-S. Li, I.-H. Liu, C.-J. Tsai, Z.-Y. Su, C.-F. Li, and C.-G. Liu, "Secure content-based image retrieval in the cloud with key confidentiality," *IEEE Access*, vol. 8, pp. 114 940–114 952, 2020.

[13] M. Shen, G. Cheng, L. Zhu, X. Du, and J. Hu, "Content-based multi-source encrypted image retrieval in clouds with privacy preservation," *Future Generation Computer Systems*, vol. 109, pp. 621–632, 2020.

[14] M. Li, M. Zhang, Q. Wang, *et al.*, "Instantcryptogram: Secure image retrieval service," in *IEEE INFOCOM 2018 - IEEE Conference on Computer Communications*, 2018, pp. 2222–2230.

[15] J. Yuan, S. Yu, and L. Guo, "Seisa: Secure and efficient encrypted image search with access control," in *2015 IEEE Conference on Computer Communications (INFOCOM)*, 2015, pp. 2083–2091.

[16] L. Song, Y. Miao, J. Weng, K.-K. R. Choo, X. Liu, and R. H. Deng, "Privacy-preserving threshold-based image retrieval in cloud-assisted internet of things," *IEEE Internet of Things Journal*, vol. 9, no. 15, pp. 13 598–13 611, 2022.

[17] W. Lu, A. Swaminathan, A. L. Varna, and M. Wu, "Enabling search over encrypted multimedia databases," in *Media Forensics and Security*, E. J. D. III, J. Dittmann, N. D. Memon, and P. W. Wong, Eds., International Society for Optics and Photonics, vol. 7254, SPIE, 2009, p. 725 418.

[18] Z. Brakerski, C. Gentry, and S. Halevi, "Packed ciphertexts in lwe-based homomorphic encryption," in *Public-Key Cryptography–PKC 2013: 16th International Conference on Practice and Theory in Public-Key Cryptography, Nara, Japan, February 26–March 1, 2013. Proceedings 16*, Springer, 2013, pp. 1–13.

[19] I. Radosavovic, R. P. Kosaraju, R. Girshick, K. He, and P. Dollár, "Designing network design spaces," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 10 428–10 436.

[20] Q. Tong, Y. Miao, L. Chen, *et al.*, "Vfirm: Verifiable fine-grained encrypted image retrieval in multi-owner multi-user settings," *IEEE Transactions on Services Computing*, vol. 15, no. 6, pp. 3606–3619, 2021.

[21] 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 *Proceedings of the 2024 International Conference on Multimedia Retrieval*, 2024, pp. 488–496.

[22] C. Dwork, A. Roth, *et al.*, "The algorithmic foundations of differential privacy," *Foundations and Trends® in Theoretical Computer Science*, vol. 9, no. 3–4, pp. 211–407, 2014.

[23] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *Theory of Cryptography: Third Theory of Cryptography Conference, TCC 2006, New York, NY, USA, March 4-7, 2006. Proceedings 3*, Springer, 2006, pp. 265–284.

[24] N. I. of Standards and T. (NIST), *Recommendation for key management - part 1: General (revision 5)*, https://csrc.nist.gov/publications/detail/sp/800-57-part-1/rev-5/final, 2022.