Received June 2, 2022, accepted June 10, 2022, date of publication June 15, 2022, date of current version June 21, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3183224

FLSIR: Secure Image Retrieval Based on Federated Learning and Additive Secret Sharing

LEI ZHANG[1], RUIYAN XIA[2], WENSHENG TIAN[1], ZHAOKUN CHENG[1], ZHICHAO YAN[1], AND PANPAN TANG[1]
[1]Nanhu Laboratory, Jiaxing, Zhejiang 314000, China
[2]School of Information Science and Technology, ShanghaiTech University, Shanghai 201210, China

Corresponding author: Wensheng Tian (tws@nanhulab.ac.cn)

ABSTRACT With the rapid deployment of electronic imaging devices, plenty of high-quality images are in the general public's hands. These images can be profitable, such as providing retrieval services; however, it is difficult for the individual to profit without the support of the cloud platform. The straightforward idea is that image owners upload their images to the cloud; yet, it is infeasible as the cloud platforms are not fully-trusted. In previous works, in order to protect the privacy of image owners, many researchers consider the Secure content-based Image Retrieval (SIR) task, which enables cloud servers to provide retrieval services while not exposing the images from the owners. However, the existing schemes are often not friendly to users as it's assumed that the owners have no profit demand and are unwilling to provide extra computation resources. This work introduces federated learning into SIR, which ensures better retrieval accuracy and efficiency; the additive secret sharing technology is utilized to protect the image information, and a better secure comparison protocol is proposed for better efficiency. We believe that the users can enjoy a better secure retrieval service with our proposed scheme. The experiment results and security analysis demonstrate that our scheme provides a significant accuracy advantage while ensuring efficiency and security.

INDEX TERMS Secure image retrieval, federated learning, additive secret sharing, secure comparison protocol.

I. INTRODUCTION

The rapid development of semiconductors ensures that imaging equipment, such as mobile phones, becomes more and more cheap and popular. Plenty of high-quality images are owned by ordinary people from all walks of life. There is no doubt that these images are valuable and profitable. As an example, people may collect images similar to the landscape photos in their hands. In this case, the Content-Based Image Retrieval (CBIR) service is a reasonable revenue model; however, it is difficult for the individual to get enough users, and with the help of the cloud platforms become an inevitable choice.

The intuitive idea is that the image owners directly upload their images to the cloud server, and the authorized user uses the retrieval service by interacting with the server. Nevertheless, the cloud server itself is not entirely trusted. Thus, a retrieval scheme supporting encrypted images is essential for such tasks.

In the last decade, many researchers have paid attention to the Secure Image Retrieval (SIR) task with the perspective of privacy protection. More details will be discussed in Section II. A common problem in the existing schemes is insufficient retrieval accuracy. It is mainly due to the fact that the existing works always focus on image privacy and assume that the image owners only own low-performance devices and thus can not perform complex operations before the secure retrieval. It is not a reasonable assumption when the image owners want to profit from these images. In this work, we focus on the scenario where image owners jointly provide secure retrieval services for authorized users with the help of cloud platforms.

In this paper, we propose a scheme called FLSIR, which provides better secure retrieval services for the authorized users at the cost of a small amount of initial work from the image owners. The federated learning and additive secret sharing technology are utilized for better accuracy and

The associate editor coordinating the review of this manuscript and approving it for publication was Aysegul Ucar.

efficiency. In summary, this work mainly makes the following contributions:

1) We introduce federated learning into the SIR. To our knowledge, this is the first work to consider the effect of image labels on the SIR task. Compared to previous works, the simpler neural network model and better network parameters are obtained and thus bring better retrieval performance.
2) We utilize additive secret sharing to protect the image contents. A novel secure comparison protocol, which needs lower interaction rounds with little extra cost during the *offline phase* (i.e., before the retrieval), is designed and further accelerates the secure retrieval.
3) Two real-world image datasets and detailed experiment results demonstrate that our scheme achieves better performances in both accuracy and efficiency. Especially, our scheme ensures the information security of the feature extraction process, which is not supported by the previous works with similar time consumption.

We organize the paper as follows. In Section II, we review the existing works on SIR. Some preliminaries are described in Section III. The system model is introduced in Section IV. The detailed construction of our scheme is described in Sections V and VI. The security analysis and experiment results are presented in Sections VII and VIII. Finally, we make the conclusion in Section IX.

II. RELATED WORK

This section first introduces the mainstream SIR works in two categories. The difference between them is whether the image owner or the cloud server undertakes the feature extraction task. A surprising fact is that quite a few existing works in both categories support the SIR from multiple image owners. Thus, we describe them in detail and show how to combine them with supervised learning. Please note that we only focus on the encryption schemes which support the retrieval task; the other functional encryption technologies [1] are beyond the scope of this work.

A. FEATURE-ENCRYPTION BASED SCHEMES

In this category, the image owner only needs to encrypt the image with the standard AES tool, and the main challenges lie in the encryption of plaintext features. There are mainly two solutions: one uses the random transformation to change features equally and ensure the encrypted features are measurable; the other uses the typical cryptography tool to support the computation of encrypted features.

In the schemes based on random transformation, Lu *et al.* [2] firstly propose three bit-permutation or projection strategies. They [3] also introduce the order-preserving encryption to support the inverted index. However, it's difficult for these schemes to support complex distance measurement methods and image updates. Notice that the earth-move distance on local features is naturally a linear programming problem; Xia *et al.* [4] encrypts the features with a random orthogonal matrix and introduces the local sensitive hash; yet, their scheme needs the two-round interaction. Yuan *et al.* [5] introduce a scheme that supports the image owner and cloud server collaboratively building the encrypted indexes. Abduljabbar *et al.* [6], [7] further use better local features and more index construction schemes to improve efficiency and accuracy. In recent years, watermark technology [8] is also considered for tracking.

The schemes based on cryptography tools mainly use homomorphic encryption to protect the image features. Lu *et al.* [9] point out that the Somewhat Homomorphic Encryption (SHE) is unable to support secure retrieval without the interactions between the cloud server and authorized user under a reasonable security model. The Fully Homomorphic Encryption (FHE) can support the process without interactions; yet, the time and storage consumption is far beyond the schemes described above. The following works [10], [11] support such opinions, and we will discuss them in subsection II-C.

B. IMAGE-ENCRYPTION BASED SCHEMES

Many researchers also focus on the second category to reduce the overload on the image owner side, where the image owner only needs to encrypt the image before uploading. The existing works in the category can be classified into schemes based on encrypted statistical features and schemes based on the encrypted classical feature.

In the schemes based on statistical features, Ferreira *et al.* [12] firstly notice that the random permutation on the position and value information of image point will not influence the statistical feature of images. Yet, such features will be too weak to retrieve; thus, Xia *et al.* [13] further introduce intra-block permutation and get a better feature with the help of the Bag-Of-Word (BOW) model. In recent, Iida *et al.* [14] combined multiple statistical features from permutation-based encryption images. They further use JPEG compression to decrease the communication between the server and the image owner; however, it will lead to image distortion after the decryption by the users. To ensure compression and integrity, many researchers [15], [16] also propose the JPEG-format preserving encryption, the idea of such works is similar to [13]. In recent work, Xia *et al.* [17] further introduce the BOW model into such schemes.

As the statistical features always result in lower accuracy and limited range of use, many works try to extract classical features (e.g., SIFT [18]) from the encrypted images. Hsu *et al.* [19] first utilize the SHE to extract the encrypted SIFT feature; yet, their scheme requires plenty of interaction and has potential risk. It is difficult to avoid such problems if only outsourcing the images to one server. Thus, the following works focus on image retrieval from multiple servers. Wang [20] and Hu *et al.* [21] use two non-collusion servers and build a secure computation protocol between them with the help of SHE. Parallel computing technology such as SIMD is also utilized for better efficiency. Nevertheless, the actual time consumption is still hard to be accepted in the real world. For instance, the feature extraction process on a 500×500 size image will cost nearly four hours.

With the rapid development of Convolutional Neural Network (CNN), the features extracted from pre-trained CNN also show much better retrieval accuracy [22]. Thus, recent SIR works [23], [24] also try to extract them from encrypted images. It is equally the problem of securely interfacing [25] the CNN model. There are many works [26], [27] focusing on the general problem, and the critical difficulty of the task is the way of computing secure non-linear computation (e.g., secure comparison). Many multi-party secure computation technologies can be utilized, and recent work [27] shows that the scheme based on secret sharing has the best efficiency in the neural network scene.

Specific to the works in SIR, Liu *et al.* [23] utilize the lattice-based FHE scheme and introduce the divide-and-conquer evaluation protocol to extract the features from the pre-trained VGG16 network; yet, their scheme uses the feature from the fully-connection layer. Without the help of the corresponding training set, it is difficult to achieve the corresponding parameters in a pre-trained way. At the same time, their time consumption is still quite unbearable as the feature extraction for a small image is over 10 seconds. Pan *et al.* [28] use DenseNet to extract hash feature for retrieval; yet, they let the user undertake the feature extraction task, which is not practical. Ma *et al.* [29] use a simplified DenseNet and tune its parameters with the encrypted images; yet, both the accuracy and efficiency are still not satisfactory. Xia *et al.* [24] combine the secret sharing technology and some excellent results in CBIR. They also introduce some novel protocol that needs fewer interactions; in this case, the *online phase* feature extraction process significantly accelerates (e.g., 5 seconds). However, their protocols can not ensure the information security of features during the extraction process. To our knowledge, no existing scheme in SIR gets better accuracy than that with pre-trained CNN.

C. MULTI-SOURCE SIR SCHEMES AND SUPERVISED SIR

A surprising fact is that although plenty of works on SIR try to improve the accuracy and efficiency, most of them only consider the scene where only one image owner outsources his image. It is unreasonable as the user always hopes to get more comprehensive search results. In summary, the most challenging problem in Multi-Source SIR (MSSIR) is how to compare the feature extracted from different image owners. To our knowledge, four existing works provide some solutions for MSSIR. Shen *et al.* [10] and Zhang *et al.* [11] utilize the HE tools to encrypt the image features and ensure the distance evaluation can be executed in the same encryption key. The above two works are feature-encryption based schemes. In the image-encryption category, Ferreira *et al.* [12] consider the situation that multiple image owners serve users as a whole, the different keys are used to protect the single image owner, and the same value key ensures the retrieval of different owners. Based on the above three works, Gu *et al.* [30] detailed define the functions required by MSSIR. On the one hand, the user can get the retrieval results from all image owners who authorized him in the constant interaction rounds; on the other hand, the image owners can freely combine to provide the service as a whole. They use the property of *permutation group* to ensure security and demonstrate that the distance between different owners can be measured directly under the same number of permutation tables.

Multiple non-collusion servers are used in all the above MSSIR schemes, and they always assume that only one server undertakes the main computation. Instead, we want to show that the MSSIR can be supported naturally with two non-collusion servers, which undertake similar computation workloads using Additive Secret Sharing (ASS) technology. The ASS supports secure computations when the secret (e.g., image) is shared into multiple servers, and we will introduce the technology in subsection III-C. In this case, each server owns one share of the image from any image owner. In other words, servers jointly maintain a semi-plaintext environment. Thus, when the user wants to query from different image owners, the servers only need to check the authorization information and choose the valid shares to join in the retrieval.

However, even considering the research in plaintext, there is little work introducing supervised learning to retrieval tasks. Generally speaking, it is difficult to label a large number of images without the drive of economic benefits. The commercial products (e.g., Google Search) achieve the images through the web crawler; thus, it is challenging to introduce supervised learning into such scenes. As described in section I, SIR is a typical scene where the owner of images can be benefited from their high-quality images; therefore, it is a reasonable assumption to introduce supervised learning into SIR, and we focus on such scene in this work.

Let us analyze the scene in the models described above. In the feature-encryption based model, the owners can train the model locally and extract the features with the trained model. However, in this case, the model is different on the side of each image owner, and thus the uploading features are invalid for distance evaluation. Thus, the supervised SIR can only be considered in the image-encryption based scheme, and the cloud servers need to generate a consistent and better model, with the help of all the image owners, for secure retrieval.

The key problem here is ensuring supervised learning from multiple different image owners and leaking no information to the unauthorized entities (e.g., cloud servers). To our knowledge, there are two typical schemes to cope with this problem. The first one is called the secure training [27]: the data owner uploads his data and the corresponding label into the servers after the encryption, and then the servers collaboratively execute the secure training with plenty of cryptography tools. It is pretty friendly to the data owner; however, the best efficiency of such schemes still needs $100\times$ more time consuming than that in plaintext. Another way is called Federated learning (FL) [31]. The detailed information of the scheme will be introduced in subsection III-B. In summary, it could get better CNN parameters only by interacting with the parameter information rather than the original images.

In this work, we introduce FL and ASS into the SIR and propose the FLSIR scheme, which is also the first work considering supervised learning in SIR. The supervised learning makes the simpler CNN model usable, and we also introduce a better secure comparison protocol to accelerate the secure inference. The utilization of the above approaches makes our work gain much better accuracy and security with similar *online phase* time consumption.

III. PRELIMINARIES

A. IMAGE FEATURE EXTRACTED BY PRE-TRAINED CNN

Research on the image retrieval task has been conducted since the 1990s. In order to get better accuracy and robustness, plenty of global features and local features extraction schemes are proposed. With the development of CNN, Babenko *et al.* [32] notice that the feature extracted from pre-trained CNN shows great advantages over the previous hand-crafted features. Uijling *et al.* [33] further point out that the feature extracted from the convolution layer is better than that got from the fully connected layer. However, the feature vectors extracted from the convolution layer are in high dimensions, and the traditional aggregation method is ineffective. Babenko *et al.* [34] finally demonstrate that the *mean aggregation* and *max aggregation* is a suitable way of aggregating CNN features. The later optimization [22] is not essential for common images.

For instance, the server can use the pre-trained VGG16 as the feature extractor. When the server gets the color image size $M \times N$, it executes the inference process until the last *max-pooling* layer before the FC layer. Now it has gotten the vector sized of $512 \times \frac{M}{32} \times \frac{N}{32}$. If the server wants to extract *sum/max aggregation* feature; for each value in the 512-dim, it would compute the sum/max value of the vector size $\frac{M}{32} \times \frac{N}{32}$. After the above process, the required 512-dim feature vector is obtained.

Obviously, a simpler neural network brings less computation consumption, which is more important in the secure version. The previous SIR works always used pre-trained *VGG16* for feature extraction. In our work, with the help of supervised learning, a simpler network *VGG11* can be utilized, and better features will be obtained.

B. FEDERATED LEARNING

An overview of federated learning is beyond the scope of this paper, and the readers can refer [35]. In SIR, the horizontal federated learning pattern is involved. The workflow of a typical client-server setting horizontal federated learning can be briefly described as follow:

(i) Locally training. The client firstly gets the initial model and parameters from the server; then the private data is utilized for training, and the gradient can be obtained after the locally training. The gradient is then sent to the server under protection (e.g., encryption, disturbance, etc.).

(ii) Secure aggregation. The server gets all the gradient information from the clients. It computes the encrypted aggregation result (e.g., gradient averaging) depending on the detailed protection scheme. After aggregation, the encrypted parameters will be sent back to all clients.

(iii) Updating parameters. The clients update the local parameters as the decrypted aggregation results, then repeat the above steps until the model converges or the number of iterations reaches a fixed number.

Federated learning completes the training while ensuring that only the owner holds the data. However, recent works [36] also demonstrate that the gradients uploaded can also leak partial information about the original images. Thus, the protection scheme for the gradients is inevitable. To avoid such information leakage, the mainstream schemes utilize differential privacy [37] to add noise on gradients or threshold secret sharing [38] to ensure only the aggregated gradient information is made public. In this work, with the help of two cloud servers, a more straightforward strategy will be discussed in subsection VI-B.

C. ADDITIVE SECRET SHARING

Secret sharing is a typical SMC technology introduced by Shamir [39]. The secret is randomly shared into multiple shares, and only sufficient shares can recover the original secret. ASS is one of the simplest secret sharing technology, which can be seen as a (n, n)-threshold secret sharing technology. In detail, the secret x can be randomly split into n shares, where the sum of these shares will be x. Clearly, the lack of any share makes the recovered value totally random. In this work, we only set $n = 2$ as only two servers are used. Please note that the computations in the secret sharing technology are all operated on the finite ring. In the following, three different rings $\mathbb{Z}_L$, $\mathbb{Z}_p$, $\mathbb{Z}_2$ will be utilized for better efficiency, where $L = 2^{32}$ and $p = 67$. We use the $[\![x]\!]_i^s$ to represent the i-th ($i = 1/2$) share of x in the finite ring $\mathbb{Z}_s$ (e.g., $s = p$) and use the $[\![x]\!]_i$ to represent the i-th share of x in any potential finite ring. For simplicity, the following value with i in subscript means one corresponding share of the value, and the S_i means the cloud servers S_1 and S_2.

The ASS naturally supports the addition on the shares as $x + y = (x_1 + y_1) + (x_2 + y_2)$. *Beaver's triples* is widely used on the secure multiplication `SecMul`. The core idea is to utilize the multiplication on random numbers generated during the *offline phase* to substitute the multiplication on the true inputs. In detail, a pre-computed triple $\{a_i, b_i, c_i | ab = c\}$ is generated and sent to server S_i during the *offline phase*. During the *online phase*, each server computes $e_i = x_i - a_i$ and $f_i = y_i - b_i$. Then two server collaboratively recover e and f and S_i can compute $z_i = fa_i + eb_i + c_i + (i - 1)ef$. It is easy to notice that $z_1 + z_2 = xy$. [40] further propose the matrix multiplication `SecMatMul`, we here omit the process for simplicity as the key idea is similar to the above.

The `SecMul` can also be used to construct secret selection protocol `SS`. The detailed protocol is shown as algorithm 1.

Algorithm 1 Secret Selection Protocol SS

```
Require: $S_i$ has $[\![x]\!]_i$, $[\![y]\!]_i$, chosen bits $[\![b]\!]^2$.
Ensure: $S_i$ gets $res_i$, $res_i$ is share of $x$ if $b = 0$; otherwise, $res_i$ is $[\![y]\!]_i$.
    Offline Phase:
1:  $\mathcal{T}$ generates a random number $c$ corresponding its shares $[\![c]\!]_i$, $assc$ and sends it to $S_i$.
2:  $\mathcal{T}$ generates the randomness that the sub-protocol uses and sends them to $S_i$.
    Online Phase:
3:  $S_i$ computes $[\![e]\!]_i^2 = [\![b]\!]_i^2 \oplus [\![c]\!]_i^2$.
4:  $S_i$ collaboratively recover $e$.
5:  if $e = 1$ then
6:      $S_1$ computes $[\![c]\!]_1 = [\![1 - c]\!]_1$, $S_2$ computes $[\![c]\!]_2 = [\![-c]\!]_2$
7:  end if
8:  $S_i$ collaboratively computes $[\![res]\!]_i = SecMul([\![y]\!]_i - [\![x]\!]_i, [\![c]\!]_i) + [\![x]\!]_i$.
```

The random number c ensures that S_i does not know res_i is the share of x or y.

A much more difficult problem is computing the comparison (i.e., $x > y$) on additive shares. Since judging $x > y$ is equivalent to $x-y > 0$, the comparison problem can also be seen as judging whether a number is over 0. As we all know, the Most Significant Bit (MSB) of a number is 0 means that the number is positive; otherwise, it is negative. Thus, the critical problem is securely computing the MSB. To our knowledge, the latest secret sharing based work [27] needs $log_2 l + 4$ rounds with the help of three non-collusion servers, where l is the bit-length of the secret; in the subsection V-A, we will give a novel construction that only needs two non-collusion servers with $\lceil log_3(l + 1)\rceil + 3$ rounds.

IV. PROBLEM DEFINITION

This work focuses on the scenario where multiple image owners sell their valuable images by providing the retrieval service with the help of the cloud servers. The critical problem here is to gain better retrieval results (e.g., accuracy, efficiency) while ensuring that the image can not be obtained in any situation besides the query result of the authorized user. The corresponding system and security model are shown as follows.

A. SYSTEM MODEL

Similar to [23], three types of entities are involved in the proposed system, i.e., the image owners, two cloud servers, and the authorized users, as shown in Fig. 1.

Image owner provides a large number of private high-value images. Each image belonging to the owner has a corresponding identity set $\mathcal{IID} = \{IID_i\}_{i=1}^n$, where n is the number of images the owner has. The image owners will label

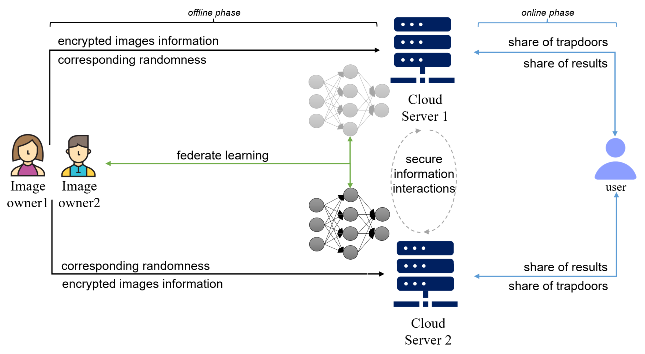


FIGURE 1. System model.

their images and join in the federated learning process to provide a better retrieval service for the user. Then, in order to totally outsource the retrieval task and ensure the images are not leaked, the owner uses ASS to protect its images and send the shares to the cloud servers.

Cloud server provides the storage service for image owners and retrieval service for authorized users. In order to ensure better retrieval service, the cloud servers firstly ensure the category of images that the platform supports and generate the public CNN model used in the federated learning process. The servers will store the encrypted images and corresponding features. When the servers get the encrypted query, the secure feature extraction and distance evaluation on vectors will be executed, and the shares of the retrieval results will be sent back to the user.

Authorized user can retrieve the images provided by the image owners and get the most similar ones with one round of interaction.

As discussed in subsection II-C, with the help of two non-collusion servers, the MSSIR can be naturally supported; thus, for simplicity, we assume that all the image owners authorize the user in the work.

B. SECURITY MODEL

Similar to previous works [12], [13], the honest-but-curious cloud servers are considered in the scheme, i.e., they follow the protocol but try to analyze the image content from the obtained information. As the servers are always well protected, we do not consider compromise attacks. As the cloud servers can be supported by different service providers (i.e., Amazon and Google), we assume the two cloud servers will not collude. As the collusion between cloud servers and any owner will not let the owner get extra information, such a scene is also not reasonable and beyond the scope of this work.

V. SECURE COMPUTATION IN THE CNN INFERENCE

The feature extraction, in another word, CNN inference, is the key process of our scheme. Thus, in this section, we propose a novel secure comparison protocol to accelerate the process and show the way to combine ASS and CNN inference.

A. SECURE COMPARISON

As described in subsection III-C, the secure comparison is equivalent to compute the MSB of the secret. Inspired by [27], the main steps of our secure comparison protocol contain three steps: *transform the share of arithmetic values to share of bits (A2B)*; *compute MSB*; *recover the MSB information*. In detail, the process is shown as the algorithms 2, 3, 5.

Step (1) to (5) shows the process of *A2B*. As the direct transformation from the value a to bits involves division operation, we avert such a difficult problem with the trusted party $\mathcal{T}$ (e.g., image owner). In the *offline phase*, $\mathcal{T}$ generates a random number x corresponding with its bits and distributes their shares. Then the servers compute and publicize the sum of a and x. The bits of $a + x$ can be obtained easily now. Although the bits of a are not achieved, we will show that the bits of $a + x$ are enough with the correction value (e.g., α, β, δ). To ensure the computation in a fixed ring (e.g., $\mathbb{Z}_{2^{32}}$), the above additions are actually modular additions. Thus, the actual value of $a + x$ may go over the scope of the ring (i.e., $a + x < a$), we will show that it is important information in the secure comparison. Following [27], the *wrap*2 function show as formula 1 is utilized to store the information.

$$wrap2(a_1, a_2, L) = \begin{cases} 0 & if \quad a_1 + a_2 < L \\ 1 & otherwise \end{cases} \tag{1}$$

The protocol `SecBitCmp` completes the comparison of two unsigned values by the share of their bits. As we all know, if one value x is less than the other one r, from the MSB to LSB (i.e., Least Significant Bit), there must exist a bit of x is 0, and the corresponding bit of r is 1; at the same time, the bits before the bit will be equal. In this case, the sum of (0 - 1), 1, and the difference of previous bits (e.g., 0 or 1) will be 0. Notably, the above situation is the only potential to get the result 0; in other words, if the 0 is achieved, it means that $x < r$. Consider that any value multiplied by 0 is 0, the above computation on each bit is executed, and the joint multiplication is computed and recovered after the masking with m.

In detail, the step (5) in `SecBitCmp` use the $1 - 2\beta$ (i.e., $(-1)^{\beta}$) to ensure the result of comparison is random and step (6) is actually computing the *xor* value (i.e., difference) of $x[t]$ and $r[t]$, t represents the bit position from MSB to LSB. Step (9) recovers the result, and step (10) corrects the influence brought $(-1)^{\beta}$. The main consumption of the protocol is happening in step (9) as multiple secure multiplications will be executed. With parallelization, it can be executed in $O(log_2 l)$ rounds of interactions by `SecMul`. In this work, to further decrease the interaction rounds, we introduce the protocol `SecThreeMul` which supports the secure multiplication of three secrets in one interaction.

Similar to `SecMul`, the `SecThreeMul` tries to transform the task to the *offline phase*. Based on formula 2, the share of xy, xz, and yz can be computed with the help of `SecMul` and the share of abc can be obtained by $\mathcal{T}$. The detailed process is shown in algorithm 4. Notably, steps (4) and (6) can be executed in parallel; thus, the `SecThreeMul` only needs one round of interaction. Although it is feasible to compute the joint multiplication of more numbers, the computation cost will increase, and the effect will be poor in the `SecBitCmp`. Consider that $l = 32$, two rounds of interaction are saved with the help of `SecThreeMul`.

Algorithm 2 Secure Value Comparison Protocol `SecValCmp`

```
Require: $S_i$ has $[\![a]\!]_i$.
Ensure: $S_i$ gets $[\![\theta]\!]_i^2$, where $\theta = 1$, then $a > L$; otherwise, $a < L$.
    Offline Phase:
1: $\mathcal{T}$ generates a random number $x \in \mathbb{Z}_L$ and the corresponding bits $[\![x[t]]\!]_i^2$, $t \in [0, l-1]$.
2: $\mathcal{T}$ computes $\alpha = wrap_2(x_1, x_2, L)$ and generates the share of $\alpha$.
3: $\mathcal{T}$ generates the randomness that the sub-protocol uses and sends them to $S_i$.
    Online Phase:
4: $S_i$ computes $r_i = a_i + x_i \ (mod\ L)$ and $\beta_i = wrap_2(a_i, x_i, L)$.
5: $S_i$ collaboratively recover $r$ to $S_1$ and $S_1$ computes $\delta = wrap_2(r_1, r_2, L)$.
6: $S_i$ collaboratively compute $[\![\eta]\!]_i^2$ = SecBitCmp($[\![x[t]]\!]_i^2$, $r + 1$]).
7: $S_1$ computes $\theta_1 = \beta_1 \oplus \delta \oplus \eta_1 \oplus \alpha_1$; $S_2$ computes $\theta_2 = \beta_2 \oplus \eta_2 \oplus \alpha_2$.
```

$$\begin{aligned}(x-a)(y-b)(z-c) &\\ = xyz + abc &+ (x-a)yz + (y-b)xz + (z-c)xy \\ &- x(y-b)(z-c) - y(x-a)(z-c) \\ &- z(x-a)(y-b)\end{aligned} \tag{2}$$

Now let us turn back to step (7) of `SecValCmp`, the following formula should be noticed.

$$\begin{cases} r = a + x - \eta \cdot L \\ r = r_1 + r_2 - \delta \cdot L \\ r_i = a_i + x_i - \beta_i \cdot L \\ x = x_1 + x_2 - \alpha \cdot L \end{cases} \tag{3}$$

As $a = a_1 + a_2 \ (mod\ L)$, $MSB(a)$ will be equal to $MSB(a_1) + MSB(a_2) + c$, where c means the carry bit from the previous bits. In detail, c means that in the finite ring $\mathbb{Z}_{L/2}$, if $a_1 + a_2$ is over the ring, then $c = 1$, else $c = 0$; in other words, c represents whether $2 \cdot a$ will be over the ring $\mathbb{Z}_L$. Thus, the relation between a_1, a_2 and L can be directly used to judge the MSB information of x; in other words, we need to get the information of θ where $a = a_1 + a_2 - \theta \cdot L$. Considering all the above formulas, it is easy to notice that $\theta = \beta \oplus \delta \oplus \eta \oplus \alpha$. The above process is shown in step (7) of `SecValCmp` and `SecCmp`.

Algorithm 3 Secure Bit Comparison Protocol `SecBitCmp`

```
Require: $S_i$ has $[\![x[t]]\!]_i^p$, $r$.
Ensure: $S_i$ gets $\eta_i$.
    Offline Phase:
 1: $\mathcal{T}$ generates a random bit $\beta$, and computes the shares $[\![\beta]\!]^2$, $[\![\beta]\!]^p$.
 2: $\mathcal{T}$ generates a random number $m \in \mathbb{Z}_p^*$ and computes its shares $[\![m]\!]^p$.
 3: $\mathcal{T}$ generates the randomness that the sub-protocol uses and sends them to $S_i$.
    Online Phase:
 4: for $t = \{l-1, l-2, \ldots, 0\}$ do
 5:     $S_i$ computes $u[t] = \mathtt{SecMul}([\![1 - 2 \cdot \beta]\!]_i^p, [\![x[t] - r[t]]\!]_i^p)$.
 6:     $S_i$ computes $[\![w[t]]\!]_i^p = [\![x[t]]\!]_i^p - 2 \cdot r[t] \cdot [\![x[t]]\!]_i^p \ (mod\ p)$, $S_1$ further computes $[\![w[t]]\!]_1^p = [\![w[t]]\!]_1^p + r[t] \ (mod\ p)$.
 7:     $S_i$ computes $[\![c[t]]\!]_i^p = [\![u[t]]\!]_i^p + \sum_{k=t+1}^{l} [\![w[k]]\!]_i^p \ (mod\ p)$, then $S_1$ computes $[\![c[t]]\!]_i^p = 1 + [\![c[t]]\!]_i^p \ (mod\ p)$.
 8: end for
 9: $S_i$ computes $d = m_i \cdot \prod_{t=0}^{l-1} c[t] \ (mod\ p)$ with SecMul and SecThreeMul.
10: if $d = 0$, then $S_1$ computes $[\![\eta]\!]_1^2 = [\![\beta]\!]_1$; otherwise, $S_1$ computes $[\![\eta]\!]_1^2 = \beta_1 \oplus 1$; $S_2$ computes $[\![\eta]\!]_2^2 = [\![\beta]\!]_2$.
```

B. SECURE CNN INFERENCE BASED ON ADDITIVE SECRET SHARING

A typical CNN model mainly contains four different types of layers. The process of secure computation on such layers is shown as follows.

(i) FC and Conv Layer. The FC and Conv layer involves multiplication and addition. For instance, the FC layer, with the weight matrix W, bias vector b, and input neuron vector x, can get the output vector $y = Wx + b$. If the matrix W and b are public, each server only needs to compute the matrix multiplication locally as $y = W(x_1 + x_2) + b_1 + b_2$.

(ii) Activation Layer. The activation layer provides the non-linear computation for the model by activation function (e.g., *sigmoid*). In this work, we focus on the *ReLU* function, which is shown as formula 4.

$$ReLU(x) = \begin{cases} x & (x > 0) \\ 0 & else \end{cases} \tag{4}$$

To securely compute the *ReLU* layer, the `SecCmp` and `SS` are combined as shown in algorithm 6. The servers first get the information about whether the value is over 0; then, the servers use such information to select the bigger one secretly. After the computation, the servers get the wanted result; however, they still know nothing about the original secret.

(iii) Pooling Layer. The pooling layer downsamples the neurons and simplifies the model parameters. Two widely used schemes are called *mean-pooling* and *max-pooling*.

Algorithm 4 Secure Three Number Multiplication Protocol `SecThreeMul`

```
Require: $S_i$ has $[\![x]\!]_i$, $[\![y]\!]_i$, $[\![z]\!]_i$.
Ensure: $S_i$ gets $[\![xyz]\!]_i$.
    Offline Phase:
 1: $\mathcal{T}$ generates random numbers $a, b, c \in \mathbb{F}$, then computes $d = abc$.
 2: $\mathcal{T}$ randomly splits $(a, b, c, d)$ into $n$ additive shares ($[\![a]\!]_i$, $[\![b]\!]_i$, $[\![c]\!]_i$, $[\![d]\!]_i$) and sends them to $S_i$.
 3: $\mathcal{T}$ generates the randomness that the sub-protocol uses and sends them to $S_i$.
    Online Phase:
 4: $S_i$ collaboratively computes $[\![xy]\!]_i = \mathtt{SecMul}(x_i, y_i)$, $[\![xz]\!]_i = \mathtt{SecMul}(x_i, z_i)$, and $[\![yz]\!]_i = \mathtt{SecMul}(y_i, z_i)$.
 5: $S_i$ computes $[\![e]\!]_i = [\![x]\!]_i - [\![a]\!]_i$, $[\![f]\!]_i = [\![y]\!]_i - [\![b]\!]_i$, and $[\![g]\!]_i = [\![z]\!]_i - [\![c]\!]_i$.
 6: $S_i$ collaboratively recover $e, f$, and $g$.
 7: $S_i$ computes $[\![xyz]\!]_i = [\![d]\!]_i - [\![x]\!]_i fg - [\![y]\!]_i eg - [\![z]\!]_i ef + e[\![yz]\!]_i + f[\![xz]\!]_i + g[\![xy]\!]_i$.
 8: $S_1$ computes $[\![xyz]\!]_1 = [\![xyz]\!]_1 + efg$.
```

Algorithm 5 Secure Comparison Protocol `SecCmp`

```
Require: $S_i$ has $[\![x]\!]_i$, $[\![y]\!]_i$.
Ensure: $S_i$ gets shares of 1 if $x > y$; otherwise, $S_i$ gets shares of 0.
    Offline Phase:
 1: $\mathcal{T}$ generates the randomness that the sub-protocol uses and sends them to $S_i$.
    Online Phase:
 2: $S_i$ collaboratively compute $b_i = \mathtt{SecValCmp}(2a_i, L)$.
 3: $S_i$ computes $res_i = MSB(a_i) \oplus b_i$.
```

As the size of the pooling block is always known to both servers, the *mean-pooling* only involves multiplication with plaintext. The *max-pooling* scheme needs to get the largest value in the block, which will involve `SecCmp` protocol. In this work, we use the *VGG11* model, and 2×2 *max-pooling* block will be utilized, and the computation process on each block is shown as algorithm 7.

In this work, all the tensors and model parameters in the secure computation are in the ring $\mathbb{Z}_L$, and 13 bits of precision are considered. Please note that the remaining integer bits are sufficient to include the number of bits that appear in the following computation.

VI. FLSIR BASED ON TWO SERVERS

In this section, we detail the process of our FLSIR scheme. In summary, three main modules are involved: *servers initialize the platform*; *image owners attend federated learning and outsource images*; and *authorized users perform retrieval.*

A. SERVERS INITIALIZE THE PLATFORM

During the initialization process, the servers need to generate the CNN model utilized in the further

```
Algorithm 6 Secure ReLU Protocol SecReLU
Require: $S_i$ has $[\![x]\!]_i$.
Ensure: $S_i$ gets the share of $x$ if $x > 0$; otherwise, $S_i$ gets the share of 0.
    Offline Phase:
1: $\mathcal{T}$ generates the randomness that the sub-protocol uses and sends them to $S_i$.
    Online Phase:
2: $S_i$ collaboratively compute $\eta_i = SecCmp([\![x]\!]_i, 0)$.
3: $S_i$ collaboratively compute $res_i = SS(\eta_i, [\![x]\!]_i, 0)$.
```

```
Algorithm 7 Secure Max-Pooling Protocol SecMaxPool
Require: $S_i$ has $[\![x_1]\!]_i$, $[\![x_2]\!]_i$, $[\![x_3]\!]_i$, $[\![x_4]\!]_i$.
Ensure: $S_i$ gets $res_i$, where $res_i$ is the share of $max(x_1, x_2, x_3, x_4)$.
    Offline Phase:
1: $\mathcal{T}$ generates the randomness that the sub-protocol uses and sends them to $S_i$.
    Online Phase:
2: $S_i$ collaboratively compute $[\![\eta_1]\!]_i = SecCmp([\![x_1]\!]_i, [\![x_2]\!]_i)$; $S_i$ collaboratively compute $[\![\eta_2]\!]_i = SecCmp([\![x_3]\!]_i, [\![x_4]\!]_i)$
3: $S_i$ collaboratively compute $[\![x_{12}]\!]_i = SS([\![\eta_1]\!]_i, [\![x_1]\!]_i, [\![x_2]\!]_i)$; $S_i$ collaboratively compute $[\![x_{34}]\!]_i = SS([\![\eta_2]\!]_i, [\![x_3]\!]_i, [\![x_4]\!]_i)$
4: $S_i$ collaboratively compute $[\![\eta_3]\!]_i = SecCmp([\![x_{12}]\!]_i, [\![x_{34}]\!]_i)$;
5: $S_i$ collaboratively compute $[\![res]\!]_i = SS([\![\eta_3]\!]_i, [\![x_{12}]\!]_i, [\![x_{34}]\!]_i)$
```

retrieval. In detail, the following three steps are needed.

(i) Determine business scope. In order to facilitate the image owners to label their images, the servers first determine the number of image types that the platform operates (e.g., n). For instance, if the platform only considers three types of images, human, bird, and flower, then n is set as 3.

(ii) Ensure model structure. Then the servers reach an agreement on the CNN model, which is utilized for feature extraction. The typical *VGG11* model is utilized as an example in this work. Notably, the number of nodes of the last FC layer is reset as n.

(iii) Initialize the parameters. In order to accelerate model training, the pre-trained parameters in the convolution layer will be used. As the last layer has been reset, the corresponding parameters can not be used directly. In this work, the initial parameters in the fully connected layer will be randomly generated with normal distribution. The CNN model and its parameters will be public for each image owner. Notably, the two servers will own the same model and parameters.

B. IMAGE OWNERS ATTEND FEDERATED LEARNING AND OUTSOURCE IMAGES

In order to completely outsource retrieval tasks (i.e., no attendance during the *online phase*) and improve users' retrieval experience, the owners need to complete the task of *outsource images* and *train a better CNN model*.

To *outsource images*, image owners need to carry out the following steps.

(i) Generate encrypted images and tensors. The image owners split their images into two image shares by ASS in the ring $\mathbb{Z}_{2^8}$. Then, the image owners use the public normalization parameters to transform each image to the input tensor and further split the tensors into two tensor shares by ASS in the ring $\mathbb{Z}_{2^{32}}$.

(ii) Upload the encrypted information. The image owners send one of their image shares and input shares to the server. As one kind of hyper-parameters, the servers know the ID information of these shares.

After getting the shared input tensors, the servers will execute the *feature extraction process*. As described in Section V-B, the convolution layer, relu layer, and max-pooling layer can be executed with `SecMatMul`, `SecReLU`, and `SecMaxPool` protocols, while the input and output of these layers are all additive shares of true information. Thus, the servers can collaboratively extract the shares of image features based on the input tensors and the current CNN model; then, following the results in CBIR, the mean-pooling aggregation scheme will be executed, and each server owns the share of the true aggregated feature vector.

To *train a better CNN model*, image owners need to execute the following steps with the help of cloud servers.

(i) Download the public model. The image owner downloads the public model published by the servers at the current time. Then following the decision of image types by the platform, the image owner labels its images with one-hot encoding. For instance, the platform ensures that the three neurons in the last layer represent *human*, *bird*, and *flower*; then, the image owner will encode a *bird* image as (0, 1, 0).

(ii) Execute the federated learning. With the labeled images, the image owners can execute the federated learning as described in subsection III-B. The detailed experiment setting will be given in subsection VIII-B1. The critical problem here is how to avoid information leakage from the gradients.

With the help of two servers, the problem can be solved simply and consistently. Before uploading the parameters, each image owner uses ASS to split the true gradient value into two shares. Each server only gets one share of the gradients from the image owners. As discussed in subsection III-B, the gradient aggregation only involves plaintext multiplication and addition; the servers can aggregate locally and reveal the final aggregated gradients to get the new aggregated model. In this case, the gradient information from a single image or image owner can be protected.

C. AUTHORIZED USERS PERFORM RETRIEVAL

With the help of two cloud servers, the authorized user can securely search the most similar images provided by the image owners. In detail, four steps will be involved during the secure retrieval.

(i) Upload trapdoor. The authorized user needs to transform its query image into the tensor with pre-determined normalization. Similarly, the tensor is shared with ASS, and each server gets one share.

(ii) Feature extraction. Similar to *the feature extraction process* described in subsection VI-B, the encrypted aggregated feature is extracted with the trained *VGG11*.

(iii) Similarity evaluation. The servers firstly need to standardize each vector. In detail, the servers firstly compute the sum of each feature vector and recover the actual sum. Then, each vector divides the corresponding sum to ensure the sum of each vector is 1, then the `SecMul` can be utilized for computing the squared euclidean distances. The problem has become how to get the size relation of the distances. As the number of database vectors keeps huge, the scheme in *max-pooling* will not be a good choice. In this case, note that the distance information is far from the original image content; thus, the information security of distances is unnecessary.

In this case, we use the scheme proposed in [24]. In detail, for the distance set x_i, the size relationship of $kx_i + b$ is equivalent to that of x_i for any positive k as long as x and $kx+b$ are all in the $\mathbb{Z}_L$. We further notice that the squared euclidean distance only involves a narrow range of chosen $\mathbb{Z}_L$. Thus, we randomly choose $2^{-10} < k < 2^{10}$ and $-100 < b < 100$. In detail, the k and b can be generated and shared by $\mathcal{T}$, and the servers can compute and reveal $kx + b$ easily with the help of `SecMul`. In a sense, it is a trade-off between space and time.

(iv) Get Results. Now, the servers have gotten the *IID* the most similar images corresponding. The servers can further send the share of such images to the authorized user. The user can quickly recover the wanted images by adding the two shares.

D. UPDATE OPERATION

The above subsections show all the processes, from servers initializing the platform to the users obtaining the wanted similar images. In this subsection, we further give the scheme during the update.

1) IMAGE UPDATE

When the image owner wants to update a few images, the owner only needs to generate and upload the corresponding shares of images and tensors. Then the servers will compute the feature of these new images for further retrieval. When the number of these images is large, the image owner can initiate a new FL process based on these images. When the owner wants to delete some images, the servers only need to delete the corresponding images and features.

2) USER UPDATE

A new joint image owner can follow the process shown in subsection VI-B. Since the servers will store the share of his uploaded gradients, the servers only need to delete his images and update the aggregated model with the negative gradients when the owner exits the system.

3) MODEL UPDATE

The aggregated model will continue changing with the change of image database and image owner; yet, at any time, the features in the database and the query feature should be extracted by the same model. Thus, a reasonable choice is to update the model at the fixed time intervals (e.g., two times in a month). When the model is not updated, the newly uploaded image also needs to extract features according to the old model; When the model is updated, the old image also needs to re-extract features according to the new model. The details of the update are beyond the scope of this paper, which we believe is easy to determine in practice.

VII. SECURITY ANALYSIS

Similar to almost all previous works in SIR, we focus on the security of image content and image features from owners or users. In detail, we need to prove that the cloud servers or other image owners cannot get the image information, and the cloud servers cannot get the information about the user's query image. In this section, we first prove the security of our protocols, then further analyze the security risk during the *offline phase* and the *online phase*.

A. SECURITY OF PROTOCOLS

The security of our protocols is proven under the typical real-ideal world simulation paradigm [41]. In real-world execution, the protocols involve the interactions between the cloud servers. In the ideal environment, the servers directly send the share of inputs to the trusted party and compute the corresponding function. To prove the security, it suffices to show that the view of the corrupted party in the real-world (i.e., adversary $\mathcal{A}$) is simulatable by the corrupted party in the ideal world (i.e., simulator $\mathcal{S}$) given its input and output. In the following, we will show that our protocol can ensure the information security of the inputs. In order to simplify the proofs, the following Lemmas are taken.

Lemma 1 [42]: A protocol is perfectly simulatable if all its sub-protocols are perfectly simulatable.

Lemma 2 [42]: If a random element r is uniformly distributed on $\mathbb{Z}_n$ and independent from any variable $x \in \mathbb{Z}_n$, then $r \pm x$ is also uniformly random and independent from x.

Lemma 3 [27]: The protocols `SecMul`, `SecMatMul` *and* `SS` *are secure in the honest-but-curious model.*

Theorem 1: The protocol `SecThreeMul` *is secure in the honest-but-curious model.*

Proof: For S_1, except those brought by `SecMul`, the $view_1$ of interaction during executing protocol is that (e_2, f_2, g_2). Take e_2, which is computed by $x_2 - a_2$ as the instance, as a_2 is totally random in $\mathbb{Z}_n$, e_2 is also totally random according to Lemma. 2. The situation for f_2 and g_2 is similar. Besides, the output $[\![xyz]\!]_1$ of S_1 will also be totally random as each term in step (7) contains an additive share. Thus, both $view_1$ and $output_1$ are simulatable by the simulator $\mathcal{S}$ and the view of $\mathcal{S}$ and $\mathcal{A}$ will be computationally indistinguishable. The situation for S_2 is basically symmetric, and we omit the analysis for simplicity. $\square$

Theorem 2: The protocol `SecBitCmp` *is secure in the honest-but-curious model.*

Proof: For S_1, except those brought by `SecMul` and `SecThreeMul`, the $view_1$ of interaction during executing protocol will be d, which is the production of all $c[i]$ and random number m. Notably, the number m undertakes the role of the blind factor, and it is difficult to infer any $c[t]$ without m. As $[\![\beta]\!]_1^2$ is totally random, the $output_1$ is also random and simulatable. The situation for S_2 is basically symmetric, and we omit the analysis for simplicity. $\square$

Theorem 3: The protocol `SecValCmp` *is secure in the honest-but-curious model.*

Proof: For S_1, except those brought by `SecBitCmp`, the $view_1$ of interaction during executing protocol will be r, which is the result of $a + x$ $(mod\ L)$, as x is totally random in $\mathbb{Z}_L$, the r is also totally random and simulatable. The situation for S_2 is basically symmetric, and we omit the analysis for simplicity. $\square$

Theorem 4: The protocol `SecCmp`, `SecReLU`, *and* `SecMaxPool` *are secure in the honest-but-curious model.*

Proof. Except those brought by `SecValCmp` and `SS`, the above protocols have no interaction view; thus, according to Lemma 1, the above protocols are simulatable. $\square$

B. SECURITY ANALYSIS DURING THE OFFLINE PHASE

As described in subsection VI-B, the *offline phase* mainly involves *secure information uploading*, *federated learning*, and *secure feature extraction* process.

In the *secure information uploading* process, the shares of image contents and corresponding tensors are sent. As all computation is executed on the image owner side, in this case, each server can get nothing due to the property of ASS. Similar, the computation in the *secure feature extraction* phase, which is actually the CNN inference, can be entirely composed of the protocols proven secure in the subsection VII-A; in other words, any information on the image contents and features, except some hyper-parameters like the image size, will not be leaked to the servers or the other image owners in the above two phases.

The *federated learning* process involves many image owners and cloud servers. As the training process is computed locally, we mainly analyze the potential risk on the aggregated gradients. As the gradients in our scheme are encrypted with the ASS before the uploading, each server can not get any information except the aggregated model in each round. Thus, the critical problem is the information leakage brought by the aggregated model. Although some recent works [36] show that the images can be inferred from the gradients; however, two important assumptions are needed: the first one is that the number of images for training should be very small (e.g., 8), and the second one is that the non-linear layer model uses should be simple (e.g., the square function). However, in our system model, the image owner will always train all his images, which will be a large number. Meantime, the model uses a more compact non-linear layer, which makes the inferring process significantly difficult. Besides, the inferring process can not recover an image with the same quality (it always has plenty of noise points). As we all know, a small ruin on the image will seriously influence its value; thus, such inferring process is unprofitable. Therefore, although ensuring the perfect security in federated learning is still an open problem, the security risk in our FLSIR process is acceptable.

C. SECURITY ANALYSIS DURING THE ONLINE PHASE

As described in subsection VI-C, the *online phase* mainly involves the *secure feature extraction* and *secure similarity evaluation* process. The *secure feature extraction* process is the same as that in the *offline phase*, which is secure; thus, we focus on the *secure similarity evaluation* process, especially the secure sort process. In `SecSort`, the value of $kx+b$ will be public, where x is the square euclidean distance between query image and images stored in the servers; thus, we mainly analyze the information leakage by it.

From the view of servers, the masked distances between the database images and the query image will be public. In this case, the similarity relation and the ratio between the distance will be leaked. Yet, we believe it is an acceptable trade-off between efficiency and security based on the following two reasons. On the one hand, it is impossible to infer the x (i.e., accurate distance) from $kx + b$ as both k and b are randomly generated. It is also impossible to infer the image content from x as the servers know nothing about both query and images in the database. On the other hand, the leakage problem is avoidable if the strategy on *max-pooling* is considered. In other words, if the user insists on the perfect security, it is easy for the system to switch to the corresponding scheme. Similarly, the standardized operation on the vectors can also achieve perfect security with the secure division protocol [27].

In summary, although the perfect security is given up for acceptable efficiency in our scheme, we believe the adversary (i.e., cloud server) can not obtain the original images or their CNN features at a considerable cost; in other words, our FLSIR is a feasible strategy in practical. Notably, both the federated learning and sort protocol have perfect-security alternatives, and it is easy to combine them with our system model.

VIII. EXPERIMENT RESULTS

In this section, we give the experiment results of our scheme and corresponding comparison results. We first give the comparison on protocols and further use two real-world datasets to evaluate the actual time consumption and accuracy results.

A. PERFORMANCE OF PROTOCOLS

As our scheme's time efficiency partly benefits from the novel protocols designed, this subsection focuses on the efficiency comparison of the proposed protocol `SecCmp`, `SecReLU`, and `SecMaxPool`. Notably, the primary consumption of the

TABLE 1. Comparison of the protocol complexities (here l means bit-length of shares in $\mathbb{Z}_L$, and l_p means bit-length of shares in $\mathbb{Z}_p$).

	SecCmp		SecReLU		SecMaxPool	
	Rounds	Comm(bits)	Rounds	Comm(bits)	Rounds	Comm(bits)
Huang [26]	$l+1$	$10l-2$	$l+1$	$10l-4$	$l+1$	$60l-24$
Falcon [27]	$log_2 l+4$	$(6l+1)l_p+l$	$log_2 l+5$	$(6l+1)l_p+4l$	$3log_2 l+15$	$(18l+3)l_p+12l$
Ours	$\lceil log_3(l+1)\rceil+3$	$(11l-6)l_p+l$	$\lceil log_3(l+1)\rceil+4$	$(11l-6)l_p+5l$	$\lceil log_3(l+1)\rceil+8$	$(33l-18)l_p+15l$

schemes based on secure multi-party comparison lies in the communication (i.e., interaction rounds and communication size). The detailed communication comparison results are shown in Table. 1.

With the help of SecThreeMul, the SecCmp of our scheme needs lower interaction rounds. With a better SecCmp protocol, the SecReLU and SecMaxPool protocol also get the better efficiency. Notably, compared to [26], our scheme can further protect the position information in the *ReLU* and *max-pooling* layer; in other words, the servers can not distinguish which neuron is positive or maximum. Compared to [27], our scheme only needs two, rather than three, non-collusion servers, which makes our work more practical.

Although the protocols in previous SIR works [24] need the lower communication rounds (e.g., 2); however, on the one hand, their scheme can not ensure the information security of the values during the CNN inference; in other words, the range of value will be leaked in each comparison; on the other hand, similar to [26], the position information will be directly leaked. In the following, we will further show that the advantage of the network model can make our scheme get similar efficiency.

B. CONSUMPTION IN THE OFFLINE PHASE

The consumption during the *offline phase* involves the *federated learning*, *information uploading*, and *feature extraction* process. To better show the corresponding consumption, two real-world image datasets (e.g., *Corel1k*, *Corel10k*) are utilized. In *Corel1k*, 1,000 images sized 384 × 256 or 256 × 384 is contained. The images are distributed in 10 categories, and each category includes 100 images. The *Corel10k* has 10,000 images sized 126 × 187 or 187 × 126 uniformly distributed in 100 categories. The number of image owners is set as 2, 5, or 10. The images are randomly and evenly distributed in the hands of all the owners. For instance, when using the *Corel1k* and 5 image owners as the experiment setting, each owner randomly owns 200 images in the *Corel1k*. The image owners' side experiments are executed on the machines with Intel Core i7-11800H CPU @ 2.3GHZ and 64 GB memory. The servers' side experiments are executed on the machine with Core i7-9700 CPU @ 3GHZ and 128GB memory. All the machines are in the same LAN environment.

1) CONSUMPTION IN THE FEDERATED LEARNING

As shown in subsection VI-B, the process of federated learning includes *training locally*, *share gradients*, *uploading gradients*, *aggregating model*, and *updating model*.

TABLE 2. Time consumption of federated learning process (s).

	Corel1k			*Corel10k*		
number of image owner	2	5	10	2	5	10
training locally	1023	610	373	4327	3030	1763
encrypt gradients	0.206	0.206	0.206	0.231	0.231	0.231
aggregate model	0.242	0.688	1.013	0.271	0.817	1.364

TABLE 3. Size of the modified *VGG11* model (KB).

	Corel1k	Corel10k
Model Size	503161	504601

The uploading and updating consumption are strongly dependent on the network environment; thus, we here only give the size of the model. The pre-trained *VGG11* model is utilized, and the number of neurons is reset as 10 and 100 for two datasets. Together, the parameters in the fully connected layer are re-generated with normal distribution. The CNN training setting are summarized as follow: $learning\_rate = 0.01$, $weight\_decay = 5E-4$, $batch\_size = 64$. The training images will be first resized as 256 × 256, then 224 × 224 size of the image will be randomly sampled. The *mean* and *std* normalize parameters are fixed as (0.4914, 0.4822, 0.44665) and (0.2023, 0.1994, 0.2010).

To better show the actual consumption, the following federated learning setting is used: 1) each image owner trains the model locally 10 epochs; 2) the owners send the shared gradients and get the new aggregated model; 3) repeat the above process 10 rounds. The detailed time consumption (single round) in the two datasets is shown in the Table. 2. Due to the consumption of federated learning synchronization, the actual training time will partly depend on the network environment and slightly more than the time shown in Table. 2.

2) CONSUMPTION OF UPLOADING INFORMATION

In order to outsource the secure retrieval task, the image content (i.e., RGB value) will be uploaded to the servers. An example of the original image, shared images and the decrypted image is shown in Fig. 2. Besides, the image should be transformed into the tensor with reasonable normalization (same as that in federated learning) to utilize the CNN model to extract the feature. As the transformation always involves division, the image owners are more suitable to execute the operation. Notably, in this step, the image is resized as 256 × 256, and only the centre 224 × 224 size image is transformed into the tensor. Similar to the model, the storage consumption of the above information is given in Table. 4. Notably, as the image was shared in a much smaller ring $\mathbb{Z}_{2^8}$, the expansion of the encrypted image is very small. In detail,

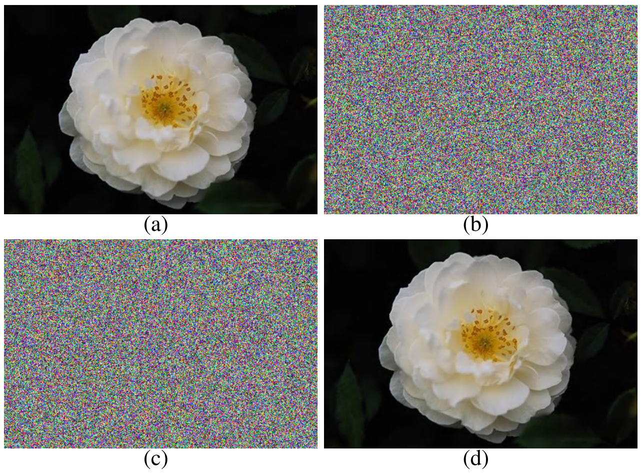


FIGURE 2. An example of the original image (a), shared images (b, c), and the decrypted image(d).

TABLE 4. Size of the outsourcing information(MB).

	Corel1k			*Corel10k*		
number of image owners	2	5	10	2	5	10
image contents	282	112.8	56.4	685	274	137
image tensors	574.3	229.7	114.9	5743.2	2297.3	1148.6

similar to previous statistical-feature based schemes [12], the average size expansion rate of *Corel1k dataset* is 873%, and that of *Corel10k dataset* is 646.2%.

3) CONSUMPTION OF FEATURE EXTRACTION PROCESS

In this work, the secure extraction process executes the CNN inference process from the first layer to the last *max-pooling* layer of *VGG11*. The detailed time consumption and corresponding comparison are shown in Table 5.

Although the schemes based on statistical features have better efficiency in the feature extraction process, we will show that their accuracy is difficult to compare with CNN-based schemes. Compared to [24], which uses *VGG16*, our scheme uses a simpler model. In detail, our scheme only needs to compute 8, rather than 11, times of *ReLU*. Thus, the efficiency of this work is similar, even if [24] uses the more efficient but not information-security protocol. Besides, as our scheme resizes the images into 256 × 256 before feature extraction, the efficiency will significantly improve while facing high-resolution images, which are more common nowadays.

C. CONSUMPTION IN THE ONLINE PHASE

As shown in subsection VI-C, the secure retrieval (i.e., *offline phase*) involves *Upload trapdoor*, *Feature extraction* and *Distance evaluation*. Similar to [24], we assume the 50 most similar images will be sent back by servers. The detailed time consumption on two real datasets is shown in Tables 6 and 7. It is easy to note that, with much less decryption cost, both [24] and this work achieve much less time consumption in the *online phase*. The main time consumption in statistical-feature based schemes is the *decryption*

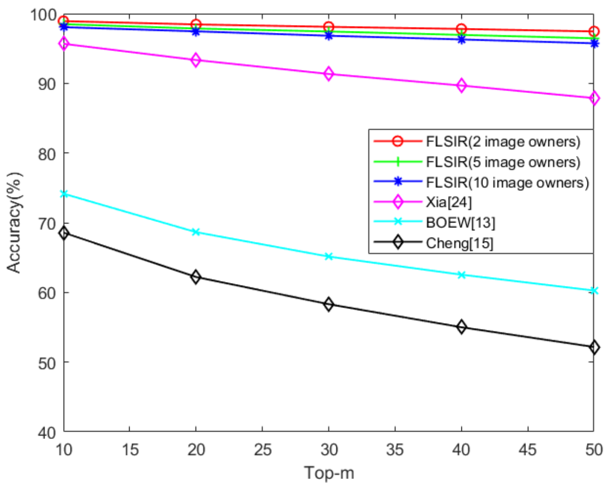


FIGURE 3. Accuracy comparison in *Corel1k* dataset.

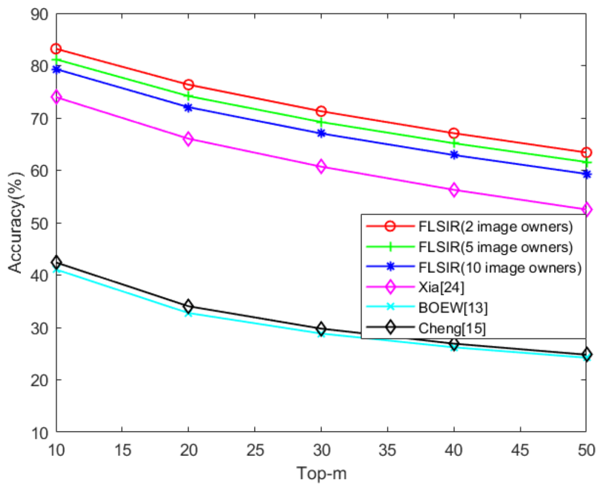


FIGURE 4. Accuracy comparison in *Corel10k* dataset.

step undertaken by the user; such a problem still exists in recent work [17], and the advantages of our scheme will be sharper when more similar encrypted images are returned. As the protocol in this work tries to get perfect security, the time consumption during *feature extraction* is slightly more than [24]. We believe that it is a meaningful trade-off between security and efficiency as the privacy demand from the user is unpredictable. At the same time, with the introduction of *federated learning*, it is feasible to introduce the simpler CNN model and gain lower time consumption in our system model.

D. RETRIEVAL PRECISION

In this work, the retrieval accuracy is defined as $P_m = m'/m$, where m' is the number of the true similar images in all m returned results. The accuracy comparison results in the two datasets are shown in Fig. 3 and 4. It is easy to note that the accuracy of the CNN-based scheme is much better than the others (more than 30%). Benefiting from the federated learning process, our work further gains a significant accuracy improvement compared to [24].

TABLE 5. Time consumption of feature extraction and aggregation.

	BOEW [13]		Cheng [15]		Xia [24]		Ours	
	Corel1k	*Corel10k*	*Corel1k*	*Corel10k*	*Corel1k*	*Corel10k*	*Corel1k*	*Corel10k*
Feature extraction	190.99s	1495.07s	109.79s	426.33s	4015.61s	10365.71s	4136.19	25695.81
Feature aggregation	620.81s	860.73s	-	-	0.028s	0.279s	0.028s	0.279s

TABLE 6. Secure retrieval consumption in *Corel1k*.

	BOEW [13]	Cheng [15]	Xia [24]			Ours		
			Offline	*Online*		*Offline*	*Online*	
	Runtime	Runtime	Runtime	Runtime	Comm Size	Runtime	Runtime	Comm Size
Trapdoor generation	0.28s	0.08s	-	0.024s	-	-	0.027s	-
Feature extraction	0.19s	0.11s	5.106s	4.01s	959.76MB	9.73s	4.136s	2955.35MB
Feature aggregation	0.01s	-	-	<1ms	-	-	<1ms	-
Distance Comparison	0.003s	54.35s	3.17ms	0.199s	141.06KB	3.17ms	0.199s	141.06KB
Decryption	14.09s	3.92s	-	0.85s	-	-	0.85s	-
Total	14.575s	58.52s	5.109s	5.206s	959.93MB	9.733s	5.212s	2955.49MB

TABLE 7. Secure retrieval consumption in *Corel10k*.

	BOEW [13]	Cheng [15]	Xia [24]			Ours		
			Offline	*Online*		*Offline*	*Online*	
	Runtime	Runtime	Runtime	Runtime	Comm Size	Runtime	Runtime	Comm Size
Trapdoor generation	0.112s	0.04s	-	0.011s	-	-	0.013s	-
Feature extraction	0.149s	0.07s	1.182s	1.05s	226.83MB	9.73s	4.136s	2955.35MB
Feature aggregation	0.01s	-	-	<1ms	-	-	<1ms	-
Distance Comparison	0.03s	32.05s	30.87ms	0.261s	1406.68KB	30.87ms	0.261s	1406.68KB
Decryption	5.59s	2.03s	-	0.422s	-	-	0.422s	-
Total	5.991s	34.19s	1.213s	1.876s	228.24MB	9.761s	4.832s	2956.72MB

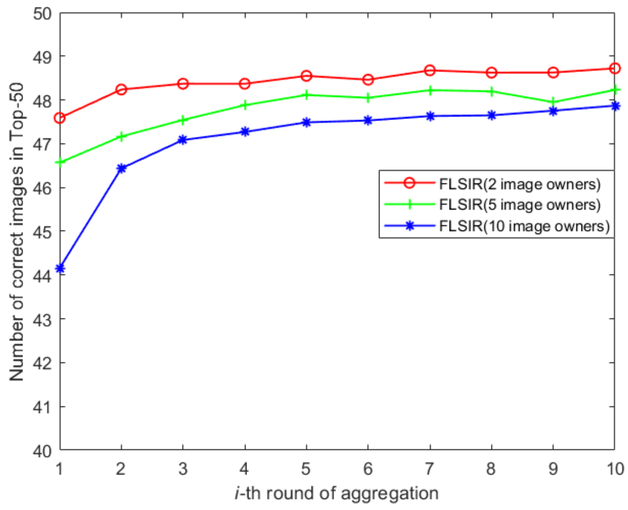


FIGURE 5. *Corel1k* Top-50 accuracy of aggregation model in each round.

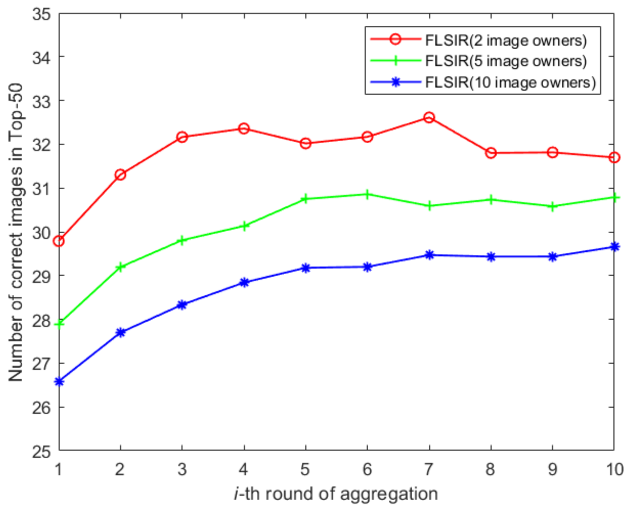


FIGURE 6. *Corel10k* Top-50 accuracy of aggregation model in each round.

To better show the effect of federated learning, we further give the accuracy at each round of aggregation as shown in Fig. 5 and 6. Notably, With only a small number of aggregation rounds, the accuracy will be significantly improved; it also implies that with appropriate settings, the consumption of image owners during the *offline phase* will be relatively low.

To our knowledge, the work gains the existing best accuracy results in the SIR. We believe that without the training process, it will be difficult for future works to make further progress.

IX. CONCLUSION

In this work, we propose a novel scheme called FLSIR, which firstly introduces supervised learning into SIR. We use federated learning to let the servers undertaken by image owners and cloud servers collaboratively train a better feature extractor while preserving the privacy of images. We also design a novel secure comparison protocol that costs fewer interaction rounds during the online phase. To sum up, compared to previous works, with some extra *offline* work, the accuracy, efficiency, and security are significantly improved; especially, compared to one of the existing best works [24], about 10%

Top-50 accuracy increment is gained. We believe that our novel system model will make secure image retrieval more practical. In the future work, we will focus on faster secure computation protocols. Furthermore, the various potential problems of federated learning (e.g., data imbalance) in the SIR scene will be carefully considered.

ACKNOWLEDGMENT

The authors would like to thank Qi Gu for his helpful discussions, experimental theories, and data support.

REFERENCES

[1] Y. Zhang, R. Zhao, X. Xiao, R. Lan, Z. Liu, and X. Zhang, ''HF-TPE: High-fidelity thumbnail-preserving encryption,'' *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 3, pp. 947–961, Mar. 2022.

[2] W. Lu, A. L. Varna, A. Swaminathan, and M. Wu, ''Secure image retrieval through feature protection,'' in *Proc. ICASSP*, Taipei, Taiwan, Apr. 2009, pp. 1533–1536.

[3] W. Lu, A. Swaminathan, A. L. Varna, and M. Wu, ''Enabling search over encrypted multimedia databases,'' *Proc. SPIE*, vol. 7254, pp. 404–414, Feb. 2009.

[4] Z. Xia, Y. Zhu, X. Sun, Z. Qin, and K. Ren, ''Towards privacy-preserving content-based image retrieval in cloud computing,'' *IEEE Trans. Cloud Comput.*, vol. 6, no. 1, pp. 276–286, Jan./Mar. 2018.

[5] J. Yuan, S. Yu, and L. Guo, ''SEISA: Secure and efficient encrypted image search with access control,'' in *Proc. IIEEE INFOCOM*, Hong Kong Apr. 2015, pp. 2083–2091.

[6] Z. A. Abduljabbar, H. Jin, A. Ibrahim, Z. A. Hussien, M. A. Hussain, S. H. Abbdal, and D. Zou, ''Privacy-preserving image retrieval in IoT-cloud,'' in *Proc. IEEE Trustcom/BigDataSE/ISPA*, Aug. 2016, pp. 799–806.

[7] Z. A. Abduljabbar, A. Ibrahim, M. A. Hussain, Z. A. Hussien, M. A. A. Sibahee, and S. Lu, ''EEIRI: Efficient encrypted image retrieval in IoT-cloud,'' *KSII Trans. Internet Inf. Syst.*, vol. 13, no. 11, pp. 5692–5716, Nov. 2019.

[8] J. Anju and R. Shreelekshmi, ''A faster secure content-based image retrieval using clustering for cloud,'' *Expert Syst. Appl.*, vol. 189, Mar. 2022, Art. no. 116070.

[9] W. Lu, A. L. Varna, and M. Wu, ''Confidentiality-preserving image search: A comparative study between homomorphic encryption and distance-preserving randomization,'' *IEEE Access*, vol. 2, pp. 125–141, 2014.

[10] M. Shen, G. Cheng, L. Zhu, X. Du, and J. Hu, ''Content-based multi-source encrypted image retrieval in clouds with privacy preservation,'' *Future Gener. Comput. Syst.*, vol. 109, pp. 621–632, Aug. 2020.

[11] L. Zhang, T. Jung, K. Liu, X.-Y. Li, X. Ding, J. Gu, and Y. Liu, ''PIC: Enable large-scale privacy preserving content-based image search on cloud,'' *IEEE Trans. Parallel Distrib. Syst.*, vol. 28, no. 11, pp. 3258–3271, Nov. 2017.

[12] B. Ferreira, J. Rodrigues, J. Leitão, and H. Domingos, ''Practical privacy-preserving content-based retrieval in cloud image repositories,'' *IEEE Trans. Cloud Comput.*, vol. 7, no. 3, pp. 784–798, Jul./Sep. 2019.

[13] 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 Trans. Services Comput.*, vol. 15, no. 1, pp. 202–214, Jan. 2022.

[14] K. Iida and H. Kiya, ''Privacy-preserving content-based image retrieval using compressible encrypted images,'' *IEEE Access*, vol. 8, pp. 200038–200050, 2020.

[15] H. Cheng, X. P. Zhang, J. Yu, and Y. Zhang, ''Encrypted JPEG image retrieval using block-wise feature comparison,'' *J. Vis. Commun. Image Represent.*, vol. 40, pp. 111–117, Oct. 2016.

[16] H. Liang, X. Zhang, and H. Cheng, ''Huffman-code based retrieval for encrypted JPEG images,'' *J. Vis. Commun. Image Represent.*, vol. 61, pp. 149–156, May 2019.

[17] Z. Xia, Q. Ji, Q. Gu, C. Yuan, and F. Xiao, ''A format-compatible searchable encryption scheme for JPEG images using bag-of-words,'' *ACM Trans. Multimedia Comput., Commun., Appl.*, vol. 18, no. 3, pp. 1–18, Aug. 2022, doi: 10.1145/3492705.

[18] D. G. Lowe, ''Object recognition from local scale-invariant features,'' in *Proc. ICCV*, vol. 2, Corfu, Greece, 1999, pp. 1150–1157.

[19] C. Y. Hsu, C. S. Lu, and S. C. Pei, ''Image feature extraction in encrypted domain with privacy-preserving SIFT,'' *IEEE Trans. Image Process.*, vol. 21, no. 11, pp. 4593–4607, Nov. 2012.

[20] Q. Wang, S. Hu, K. Ren, J. Wang, Z. Wang, and M. Du, ''Catch me in the dark: Effective privacy-preserving outsourcing of feature extractions over image data,'' in *Proc. IEEE INFOCOM*, San Francisco, CA, USA, Apr. 2016, pp. 1–9.

[21] S. Hu, Q. Wang, J. Wang, Z. Qin, and K. Ren, ''Securing SIFT: Privacy-preserving outsourcing computation of feature extractions over encrypted image data,'' *IEEE Trans. Image Process.*, vol. 25, no. 7, pp. 3411–3425, Jul. 2016.

[22] X.-S. Wei, J.-H. Luo, J. Wu, and Z.-H. Zhou, ''Selective convolutional descriptor aggregation for fine-grained image retrieval,'' *IEEE Trans. Image Process.*, vol. 26, no. 6, pp. 2868–2881, Jun. 2017.

[23] F. Liu, Y. Wang, F.-C. Wang, Y.-Z. Zhang, and J. Lin, ''Intelligent and secure content-based image retrieval for mobile users,'' *IEEE Access*, vol. 7, pp. 119209–119222, 2019.

[24] Z. Xia, Q. Gu, L. Xiong, W. Zhou, and J. Weng, ''Privacy-preserving image retrieval based on additive secret sharing,'' 2020, *arXiv:2009.06893*.

[25] C. Juvekar, V. Vaikuntanathan, and A. Chandrakasan, ''GAZELLE: A low latency framework for secure neural network inference,'' in *Proc. USENIX Secur. Symp.*, Baltimore, MD, USA, 2018, pp. 1651–1669.

[26] K. Huang, X. Liu, S. Fu, D. Guo, and M. Xu, ''A lightweight privacy-preserving CNN feature extraction framework for mobile sensing,'' *IEEE Trans. Depend. Sec. Comput.*, vol. 18, no. 3, pp. 1441–1455, Jun. 2021.

[27] S. Wagh, S. Tople, F. Benhamouda, E. Kushilevitz, P. Mittal, and T. Rabin, ''Falcon: Honest-majority maliciously secure framework for private deep learning,'' *Proc. Privacy Enhancing Technol.*, vol. 2021, no. 1, pp. 188–208, Jan. 2021.

[28] W. Pan, M. Wang, J. Qin, and Z. Zhou, ''Improved CNN-based hashing for encrypted image retrieval,'' *Secur. Commun. Netw.*, vol. 2021, Feb. 2021, Art. no. 5556634, doi: 10.1155/2021/5556634.

[29] W. Ma, T. Zhou, J. Qin, X. Xiang, Y. Tan, and Z. Cai, ''A privacy-preserving content-based image retrieval method based on deep learning in cloud computing,'' *Expert Syst. Appl.*, vol. 203, Oct. 2022, Art. no. 117508.

[30] Q. Gu, Z. Xia, and X. Sun, ''MSPPIR: Multi-source privacy-preserving image retrieval in cloud computing,'' *Future Gener. Comput. Syst.*, vol. 134, pp. 78–92, Sep. 2022.

[31] L. Li, Y. Fan, M. Tse, and K.-Y. Lin, ''A review of applications in federated learning,'' *Comput. Ind. Eng.*, vol. 149, Nov. 2020, Art. no. 106854.

[32] A. Babenko, A. Slesarev, A. Chigorin, and V. Lempitsky, ''Neural codes for image retrieval,'' in *Proc. ECCV*, Zurich, Switzerland, 2014, pp. 584–599.

[33] J. R. Uijlings, K. E. Van De Sande, T. Gevers, and A. W. Smeulders, ''Selective search for object recognition,'' *Int. J. Comput. Vis.*, vol. 104, no. 2, pp. 154–171, 2013.

[34] A. B. Yandex and V. Lempitsky, ''Aggregating local deep features for image retrieval,'' in *Proc. ICCV*, Santiago, Chile, Dec. 2015, pp. 1269–1277.

[35] P. Kairouz *et al.*, ''Advances and open problems in federated learning,'' *Found. Trends Mach. Learn.*, vol. 14, nos. 1–2, pp. 1–210, Jun. 2021.

[36] L. Zhu and S. Han, ''Deep leakage from gradients,'' in *Federated Learning*. Cham, Switzerland: Springer, 2020, pp. 17–31.

[37] K. Wei, J. Li, M. Ding, C. Ma, H. H. Yang, F. Farokhi, S. Jin, Q. S. T. Quek, and H. V. Poor, ''Federated learning with differential privacy: Algorithms and performance analysis,'' *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 3454–3469, 2020.

[38] K. Mandal, G. Gong, and C. Liu, ''Nike-based fast privacy-preserving highdimensional data aggregation for mobile devices,'' Univ. Waterloo, Waterloo, ON, Canada, Tech. Rep. CACR 2018-10, 2018, pp. 142–149.

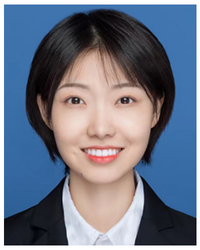
[39] A. Shamir, ''How to share a secret,'' *Commun. ACM*, vol. 22, no. 11, pp. 612–613, Nov. 1979.

[40] S. Wagh, D. Gupta, and N. Chandran, ''SecureNN: Efficient and private neural network training,'' *IACR Cryptol. ePrint Arch.*, vol. 2018, p. 442, Feb. 2018.

[41] R. Canetti, ''Universally composable security: A new paradigm for cryptographic protocols,'' in *Proc. FOCS*, Las Vegas, NV, USA, 2001, pp. 136–145.

[42] D. Bogdanov, S. Laur, and J. Willemson, ''Sharemind: A framework for fast privacy-preserving computations,'' in *Proc. ESORICS*, Málaga, Spain, 2008, pp. 192–206.

LEI ZHANG received the Ph.D. degree in computer science from the National ICT of Australia (NICTA), in 2009. His research interests include cloud computing, edge computing, federated learning, and secure multiparty computation. He has participated and coordinated several national/international projects on cloud and cloud computing.



ZHAOKUN CHENG received the master's degree in management science and engineering from Central South University, China, in 2021. She is currently a Research Assistant with the Nanhu Laboratory, China. Her research interests include privacy-preserving technologies, cross-efficiency, productivity measurement and improvement, social network analysis, and data analysis.

RUIYAN XIA is currently pursuing the bachelor's degree in computer science and technology with ShanghaiTech University. Her current research interests include cybersecurity, privacy-preserving technologies, federated learning, and big data.

ZHICHAO YAN received the master's degree in electronic and communication engineering from Hangzhou Dianzi University, China, in 2017. He is currently a Researcher with the Nanhu Laboratory, China. His research interests include big data security, privacy preserving computing, and cloud computing.

WENSHENG TIAN is currently a Researcher with the Nanhu laboratory, China. His current research interests include machine learning, security, and privacy.

PANPAN TANG received the Ph.D. degree in cartography and geographic information system from the Aerospace Information Research Institute, Chinese Academy of Sciences, Beijing, China, in 2014. He is currently an Associate Professor with the Nanhu Laboratory, Jiaxing, Zhejiang, China. His research interests include deep learning in remote sensing image classification and privacy-preserving AI.