information

MDPI

Article

# A Content-Based Image Retrieval Scheme Using an Encrypted Difference Histogram in Cloud Computing

**Dandan Liu [1], Jian Shen [1,2], Zhihua Xia [1,2,*] and Xingming Sun [1,2]**

[1] School of Computer and Software, Nanjing University of Information Science & Technology, Nanjing 210044, China; jsrgldd0310@163.com (D.L.); s_shenjian@126.com (J.S.); sunnudt@163.com (X.S.)

[2] Jiangsu Engineering Center of Network Monitoring, Nanjing University of Information Science & Technology, Nanjing 210044, China

* Correspondence: xia_zhihua@163.com

Received: 7 June 2017; Accepted: 28 July 2017; Published: 7 August 2017

**Abstract:** Content-based image retrieval (CBIR) has been widely used in many applications. Large storage and computation overheads have made the outsourcing of CBIR services attractive. However, the privacy issues brought by outsourcing have become a big problem. In this paper, a secure CBIR scheme based on an encrypted difference histogram (EDH-CBIR) is proposed. Firstly, the image owner calculates the order or disorder difference matrices of RGB components and encrypts them by value replacement and position scrambling. The encrypted images are then uploaded to the cloud server who extracts encrypted difference histograms as image feature vectors. To search similar images, the query image is encrypted by the image users as the image owner does, and the query feature vector is extracted by the cloud server. The Euclidean distance between query feature vector and image feature vector is calculated to measure the similarity. The security analysis and experiments demonstrate the usability of the proposed scheme.

**Keywords:** difference histogram; searchable encryption scheme; content-based image retrieval

## 1. Introduction

The number of images generated by all kinds of devices has been greatly increasing in recent years. Accordingly, content-based image retrieval (CBIR) technology research has generated wide attention and made remarkable advances [1–5]. Images themselves are storage-consuming and the CBIR technologies are typically of high computation complexity. Thus, there is a motivation to outsource the CBIR services to the cloud server.

The public cloud storage services provide cheap storage space, are computationally convenient, and have multiple access modes. Although the cloud storage service has great advantages, it is worth pondering the privacy security problem it brings. The user defaults that the cloud service provider is untrustworthy [6–8]. The urgent need for privacy protection has attracted experts to study secure outsourced CBIR schemes.

To solve the privacy problems of outsourcing CBIR services, the existing secure CBIR schemes mainly take the following steps. Firstly, the user extracts features directly from the plaintext image, builds an index, and then encrypts the features, index, and images. After that, the encrypted feature, index, and image are uploaded to the cloud server. In these CBIR schemes, the cloud server only provides storage and retrieval services. The computation burden on the user side is still serious. Therefore, it is necessary to propose a secure CBIR scheme that can directly extract features from the ciphertext domain on the cloud server side.

**Contributions.** This paper proposes a secure CBIR scheme based on encrypted difference histograms (EDH-CBIR). The major contributions are enumerated as follows:

(1) A specially designed image encryption method is proposed to support the feature extraction directly from the ciphertext domain.
(2) In EDH-CBIR, users only need to complete the work of image encryption, the feature extraction and index establishment will be completed by cloud server, which will largely reduce the user's work.
(3) This paper takes the statistical characteristics of difference histogram into account, and considers two difference calculation methods. The retrieval accuracy and security in the two situations are tested and analyzed, respectively.

The rest sections as follows. The Section 2 describes the related work of the existing typical CBIR schemes, and the next section describes the system and security overview. Section 4 elaborates on the proposed scheme. Security analysis and experimental results are presented in Sections 5 and 6, respectively. Finally, the Section 7 gives the conclusion.

## 2. Related Work

Early searchable symmetric encryption (SSE) schemes [9–13] are mainly proposed to support the secure retrieval of the text. Lu et al., for the first time, proposed a ciphertext image retrieval scheme in 2009 [14]. The scheme extracts the local features from the whole image database, and uses the clustering method to generate the visual word (Visual Words). The Jaccard similarity of visual word sets is used to measure the similarity between images, and the image content is protected by Min-hash and order preserving. In the same year, Lu et al. also analyzed the characteristics of three kinds of feature protection methods [15]. The features that are encrypted with bit plane randomization and random unary coding can support Hamming distance calculation, and the features that are encrypted with the random projection algorithm can support L1 distance. In 2014, Lu et al. proposed an image retrieval algorithm based on homomorphic encryption, and compared the retrieval precision, efficiency and storage overhead with three previous proposed encryption algorithms [16]. The results show that the algorithm based on homomorphic encryption is time-consuming despite its high level of security. Xia et al. [17] proposed an image retrieval scheme based on the scale-invariance feature transform (SIFT) feature and the earth mover's distance (EMD). This scheme uses the SIFT algorithm to extract the image feature. The EMD algorithm is used to measure the frequency histogram of the distance. The EMD algorithm is essentially a linear programming problem. The authors used the linear transformation of the linear programming problem to protect the features. Cheng et al. [18] proposed an encrypted image retrieval scheme based on Markov process. The scheme encrypts encoding data of JPEG image by stream cipher, and extracts Markov features from the encrypted image data directly. The similarity between images is ultimately measured by the similarity of Markov features. In [6], Xia et al. represented the images with four MPEG descriptors, and protected the image features by secure k-nearest neighbour (kNN) algorithm. The proposed scheme used locality-sensitive hash to improve the search efficiency. Degang et al. [19] proposed a triple-bit quantization-based scheme. The scheme assigns a 3-bit to each dimension and applies the asymmetric distance algorithm to re-rank candidates. Although the aforementioned schemes address the privacy issues, the computational burden on users is quite enormous. In order to solve the problem, Bellafqira et al. [20,21] proposed two privacy-preserving feature extraction methods in the homomorphic encryption domain. However, the extracted features cannot be used for image similarity retrieval. Ferreira et al. [22] proposed an secure image encryption algorithm for image retrieval, and separately processed the image color information and texture information. This scheme encrypted texture information by random encryption algorithm, and protected color information by using deterministic encryption. The authors used the color histogram as an image feature to support image retrieval. The scheme proposed by Ferreira greatly reduces the computational burden of users. However, the color histogram ignores the texture information of the image. Inspired by [22], a new image retrieval scheme based on a difference histogram is proposed to improve the retrieval accuracy.

## 3. System and Security Model

- System model.

  The proposed system includes three entities: image owner, cloud server and users. The specific tasks of the three entities and the communication between them are shown in Figure 1.

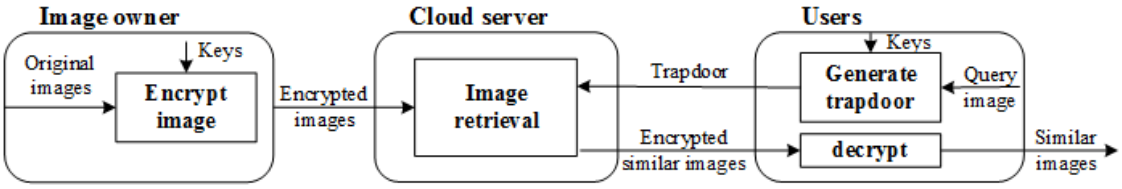



**Figure 1.** System model.

  *Image owner side*. The image owner holds an original image database $\mathbf{I} = \{I_i\}_{i=1}^{n}$, where $I_i$ is the $i$th image in image database and $n$ is the total image number in the image database. Firstly, the image owner generates the secret keys to encrypt the original image, and the encrypted image database can be represented as $\mathbf{C} = \{C_i\}_{i=1}^{n}$. After that, the image owner outsources the encrypted image database **C** to the cloud server.
  *Cloud server side*. After receiving the encrypted images, the cloud server extracts image features from the encrypted images and establishes the index. On receiving a search request from user, the cloud server extracts features from the trapdoor, and searches the most similar features in index. The $k$ images with the most similar features are returned to the user.
  *User side*. To search the wanted images, the users encrypt the query image as image owner does. The encrypted query image is uploaded as the trapdoor to the cloud server. User decrypts the similar images returned by the cloud server with secret keys.

- Security model.

  As a typical SSE scheme, the proposed scheme mainly considers semi-honest security model, i.e., honest-but-curious (HBC) security model.
  In the HBC model, the cloud server will complete the specified tasks, but may take interest in the content of the encrypted image by acquiring and analyzing historical search records. The image owner and the image users are trustworthy believed, meaning that the image owner and image users will not reveal any privacy information to the cloud server during the communication. Furthermore, if image $I_i$ and image $I_j$ return the same similar image set, it is not difficult to infer that the image $I_i$ is similar to image $I_j$. Hence, the information leakage caused by this way will not be discussed.

## 4. The Proposed Scheme

Section 3 presents the scheme with six main tasks: image encryption in the image owner side; feature extraction, index establishment and image retrieval in the cloud server side; and trapdoor generation and image decryption in the user side. The image user encrypts the query image as the image owner does and decrypts the encrypted similar images with the opposite operation of encryption. The tasks of the users are similar to those of the image owner, so we focus on image encryption, feature extraction and index establishment and image retrieval.

### *4.1. Image Encryption*

The algorithm is based on RGB color space, and the detailed process of encryption consists of two steps: difference matrix computation and difference matrix encryption.

- Difference matrix computation.

Difference histogram is used as the image feature. To extract the feature directly from the encrypted image, the image owner calculates the difference matrix of plaintext image. The difference matrix calculation is divided into the following three steps:

(1) *One-dimensional matrix*. Assuming that the image size in the image database is $M \times N$. We select the appropriate conversion method to convert the image pixel matrix into a one-dimensional array $Arrary$, in which the length of $Arrary$ is $imgsize$ and the $imgsize = M \times N$. Here, two conversion methods are mainly considered: orderly scanning and disorderly block scanning, and the schematic diagram is shown in Figure 2.

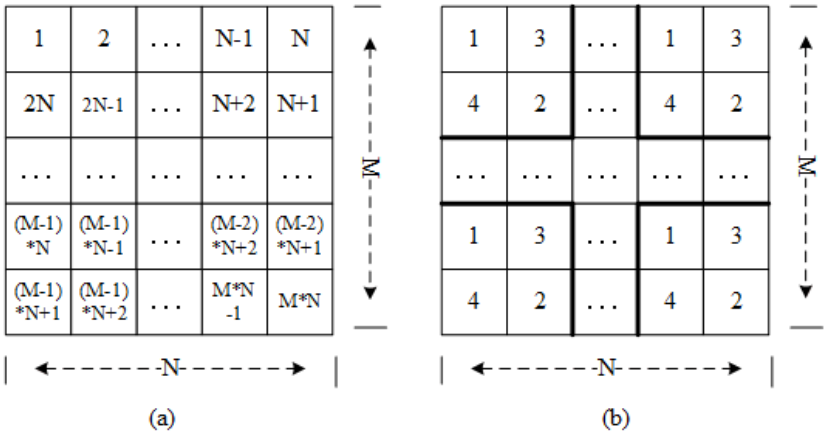



**Figure 2.** Schematic diagram of image pixel scanning, (**a**) orderly scanning, (**b**) disorderly scanning.

For the order scanning method, the pixel values are obtained by orderly scanning the pixel matrix. The scanned sequence is shown in Figure 2a. For the disorderly block scanning method, we firstly divide the image into blocks. Then, the pixel values is obtained by disorderly scanning the block pixel matrix. Note: we arrange image blocks by line priority here. The Figure 2b gives a disorder block scanning example with the block size of $2 \times 2$. The obtained pixels by two scanning methods are stored sequentially, so that the pixel matrix can be transformed into a one-dimensional array. According to the above scanning methods, we can obtain two kinds of arrays: order array and disorder array. The pixels in two kinds of array are represented as $Array(pixel) = \{pixel | 1 \leq pixel \leq imgsize\}$, and the associated pixel values in RGB components are represented as $\{Value(pixel)_*\}_{*\in\{r,g,b\}} \in [0, 255]$.

(2) *Difference value calculation*. We acquire one-dimensional difference arrays $DiffArray$ by subtracting adjacent value in the $Array$. The $DiffArray$ is represented as $DiffArray(pixel) = \{pixel | 1 \leq pixel \leq imgsize\}$. The formula of $DiffValue$ computation as follows:

$$\{DiffValue(pixel)_*\}_{*\in\{r,g,b\}} = \{Value(pixel+1)_*\}_{*\in\{r,g,b\}} - \{Value(pixel)_*\}_{*\in\{r,g,b\}} \quad (1)$$

(3) *Difference matrix acquisition*. The difference matrix $I_{DM}$ can be gained by inverse conversion of the difference array $DiffArray$. The transformation method is the inverse operation of the initial conversion method. The difference position of difference matrix can be represented as $p = \{(x, y) | 1 \leq x \leq M, 1 \leq y \leq N\}$, and the corresponding difference value can be represented $pv$.

- Difference matrix encryption.

After difference matrix computation, we obtain two types of difference matrices: order difference matrix (ODM) and disorder difference matrix (DDM). The difference value in the difference

matrix shows the changing trend of the pixel, and the difference position shows the roughness of difference matrix . To prevent the leakage of privacy, we encrypt the difference matrix by value replacement and position scrambling for ODM and DDM. Since the encryption methods are the same, we simply abbreviate as the differential matrix (DM) encryption.

– Value replacement. The image owner firstly generates three random permutations $key_{vr}$, $key_{vg}$, $key_{vb}$ of the range $[\{val_{min*}\}_{*\in\{r,g,b\}}, ..., \{val_{max*}\}_{*\in\{r,g,b\}}]$ by a pseudo-random permutation generator, where the parameter $val_{min}$ is the minimum difference value and $val_{max}$ is the maximum difference value in the image database. After that, image owner replaces the original difference value by the value in the random sequence. Denote $pv_r$, $pv_g$, and $pv_b$ as the three components of difference values $pv$ in $I_{DM}$, and $pv'_r$, $pv'_g$, $pv'_b$ are the corresponding encryption results. For $\forall pv \in I_{DM}$, do:

$$\begin{aligned} pv'_r &\leftarrow key_{vr}[pv_r], \\ pv'_g &\leftarrow key_{vg}[pv_g], \\ pv'_b &\leftarrow key_{vb}[pv_b]. \end{aligned} \tag{2}$$

A simple example is given to visualize the difference value replacement method. A sequence example is shown in Table 1. We give an original difference matrix in Figure 3a, and replace the original difference values with the values in the random sequence (Table 1), the results are shown in Figure 3b. Note: simple instance does not consider color space and it is just used as instantiated objects.

**Table 1.** The sequence example.

| **Original difference value** | $val_{min}$ | ... | 0 | 1 | 2 | 3 | 4 | 5 | ... | $val_{max}$ |
|---|---|---|---|---|---|---|---|---|---|---|
| **Random sequence** | $val_{max}$ | ... | 3 | 5 | 4 | 0 | 1 | 2 | ... | $val_{min}$ |

| | | | |
|---|---|---|---|
| 0 | 1 | 5 | 2 |
| 1 | 4 | 2 | 1 |
| 1 | 3 | 2 | 5 |
| 4 | 3 | 0 | 1 |

(a)

| | | | |
|---|---|---|---|
| 3 | 5 | 2 | 4 |
| 5 | 1 | 4 | 5 |
| 5 | 0 | 4 | 2 |
| 1 | 0 | 3 | 5 |

(b)

**Figure 3.** The sample of value replacement, (**a**) an example of the original difference matrix, (**b**) difference matrix after value replacement.

– Position scrambling. $I'_{DM}$ is the encryption results of value replacement. The image owner generates three random permutations $key_{pr_h}$, $key_{pg_h}$, $key_{pb_h}$ of the rang $[1, ..., M]$ and three random permutations $key_{pr_w}$, $key_{pg_w}$, $key_{pb_w}$ of the rang $[1, ..., N]$ by a pseudo-random permutation generator. Denote $p_r$, $p_g$, $p_b$ as the three components of difference value position $p$. For $\forall p = \{(x, y)|1 \le x \le M, 1 \le y \le N\} \in I'_{DM}$, do:

$$\begin{aligned} (x,y)_r &\leftarrow (key_{pr_h}[x], key_{pr_w}[y]), x \in [1, ..., M], y \in [1, ..., N], \\ (x,y)_g &\leftarrow (key_{pg_h}[x], key_{pr_w}[y]), x \in [1, ..., M], y \in [1, ..., N], \\ (x,y)_b &\leftarrow (key_{pb_h}[x], key_{pr_w}[y]), x \in [1, ..., M], y \in [1, ..., N]. \end{aligned} \tag{3}$$

Give an example: we assume that a difference in the R component of the difference matrix in position $(1,1)$, i.e., $(x,y)_r = (1,1)$. The first value in the random sequence $key_{pr_h}$ is assumed to be 92, and the first value in the random sequence $key_{pr_w}$ is assumed to be 88, i.e., $key_{pr_h}[1] = 92, key_{pr_w}[1] = 88$ . According to the Formula (3), $(x,y)_r \leftarrow (key_{pr_h}[1], key_{pr_w}[1]) = (92,88)$, i.e., the position of first difference position $(1,1)$ becomes $(92,88)$. Operating on all pixel locations, we can get the encrypted image $C$.

### *4.2. Feature Extraction and Index Construction*

The proposed scheme reduces the user's computational burden by transferring feature extraction and indexing tasks to the cloud server. When the image owner uploads the encrypted image database to the cloud server, the cloud server extracts histograms directly from encrypted images as image features.

The difference value in the R, G, B components range from $[\{val_{min*}\}_{*\in\{r,g,b\}}, ..., \{val_{max*}\}_{*\in\{r,g,b\}}]$. The frequency of each difference value is calculated as the difference histogram. The cloud server extracts the difference histograms $\{\mathbf{h}_{i*}\}_{*\in\{r,g,b\}} = \{h_{i1}, h_{i2}, ..., h_{i\tau}, ..., h_{i\{val_{sum*}\}}\}_{*\in\{r,g,b\}}$ of the three components directly from the encrypted image and combines them into a feature vector, as shown below: $\mathbf{f}_i = \{\mathbf{h}_{ir}, \mathbf{h}_{ig}, \mathbf{h}_{ib}\} = \{f_{i1}, f_{i2}, ..., f_{i\tau}, ..., f_{i\{Val_{sum}\}}\} = \{f_{i\tau}\}_{\tau=1}^{val_{sum}}$, where $i \in \{1, ..., n\}$, $Val_{sum} = val_{sum_r} + val_{sum_g} + val_{sum_b}$, and $\{Val_{sum*}\}_{*\in\{r,g,b\}}$ can be represented as:

$$\{val_{sum*}\}_{*\in\{r,g,b\}} = \{val_{max*}\}_{*\in\{r,g,b\}} - \{val_{min*}\}_{*\in\{r,g,b\}}. \tag{4}$$

In this way, the ciphertext image can be mapped into ciphertext feature vectors. Based on the relationship between encrypted images and feature vectors, a linear index is established as shown in Table 2.

**Table 2.** The index.

| Image Identity | Feature Vector |
|---|---|
| $C_1$ | $\mathbf{f}_1 = \{f_{11}, f_{12}, ..., f_{1\tau}, ..., f_{1\{Val_{sum}\}}\}$ |
| ... | ... |
| $C_i$ | $\mathbf{f}_i = \{f_{i1}, f_{i2}, ..., f_{i\tau}, ..., f_{i\{Val_{sum}\}}\}$ |
| ... | ... |
| $C_n$ | $\mathbf{f}_n = \{f_{n1}, f_{n2}, ..., f_{n\tau}, ..., f_{n\{Val_{sum}\}}\}$ |

### *4.3. Image Retrieval*

The user encrypts the query image with the mentioned encryption method as trapdoor **TD**. After accepting the trapdoor **TD** from users, the cloud server extracts the feature vector from **TD**, denoted as $\mathbf{f}_q = \{f_{q1}, f_{q2}, ..., f_{q\tau}, ..., f_{q\{Val_{sum}\}}\}$. In search of the most similar images, the cloud server retrieves the index, i.e., the cloud server matches the $\mathbf{f}_q$ with all the feature vectors in the index. Euclidean distance is used to measure similarity. Cloud server calculates the Euclidean distance $d(\mathbf{f}_q, \mathbf{f}_i)$ between $\mathbf{f}_q$ and all $\mathbf{f}_i, i \in \{1, ..., n\}$. The $d(\mathbf{f}_q, \mathbf{f}_i)$ is used as the similarities of images in database to the query image. The $d(\mathbf{f}_q, \mathbf{f}_i)$ is calculated as:

$$d(\mathbf{f}_q, \mathbf{f}_i) = \sqrt{\sum_{\tau=1}^{Val_{sum}} (f_{q\tau} - f_{i\tau})^2} \tag{5}$$

By computing all the $d(\mathbf{f}_q, \mathbf{f}_i)$, we get the distance between the query feature and all the features. Similar vectors have smaller distances. Therefore, all the distances are sorted in ascending order, and a

similar image database is made up of the $k$ images with the most smallest distance. Then, the similar image database is returned to the query user. Thus, the cloud server has completed the process of image retrieval.

## 5. Security Analysis

Honest-but-curious (HBC) cloud server is considered as the security model. We analyze the security of the proposed scheme in the ciphertext-only attack (COA) model and known background attack (KBA) model.

### *5.1. Security under COA Model*

Our security proofs follow the paradigm in secure multi-party computations [23]. Interaction between cloud server and users is defined as a real experiment, and the HBC cloud server is defined as an attacker $A$. We build an ideal experiment, the simulator $S$ is used to simulate all the possible attacks by cloud servers. If the difference between the real experiment and the ideal experiment is subtle, the proposed scheme proves security.

**Theorem 1.** *The scheme proposed is secure against HBC probabilistic polynomial time adversaries. The security strength is used to measure security of the proposed scheme.*

- **Security of the encrypted image.** Simulator $S$ simulates a image set $\mathbf{I}^S$. The simulator $S$ knows the image number and the image size of the image database, so it can simulate a hypothetical image database $\mathbf{I}^S$ similar to real image database $\mathbf{I}$. EDH-CBIR contains the encrypted order difference histogram-based CBIR scheme (EODH-CBIR) and the encrypted disorder difference histogram-based CBIR scheme (EDDH-CBIR). The security of the two schemes is analyzed.

  – *EODH-CBIR.* To simulate an image in EODH-CBIR, the simulator $S$ needs to solve a permutation to get the order difference matrix, and needs to solve $val_{sum_r}! + val_{sum_g}! + val_{sum_b}!$ permutations for value replacement, and $3 * imgsize!$ for pixel scrambling of three components. *Sec* is defined as the security strength and the *Sec* of order difference as $Sec_{od}$, which can be expressed as:

$$\begin{aligned} Sec_{od} = & log_2(val_{sum_r}!) + log_2(val_{sum_g}!) \\ & + log_2(val_{sum_b}!) + 3 * log_2(imgsize!)bit \end{aligned} \tag{6}$$

  – *EDDH-CBIR.* To simulate an image in EDDH-CBIR, the simulator $S$ needs to solve 3*$imgsize!$ permutations to get disorder difference matrix , $val_{sum_r}! + val_{sum_g}! + val_{sum_b}!$ permutations for value replacement, and $3 * imgsize!$ for pixels scrambling of three components. We define the security strength of disorder difference as $Sec_{dd}$, which can be expressed as:

$$\begin{aligned} Sec_{dd} = & log_2(val_{sum_r}!) + log_2(val_{sum_g}!) \\ & + log_2(val_{sum_b}!) + 6 * log_2(imgsize!)bit \end{aligned} \tag{7}$$

- **Security of the image feature.** The proposed scheme extracts the difference histogram from the encrypted image directly as the image feature. Simulator $S$ simulates an image set $\mathbf{I}^S$. The simulator $S$ can extract simulated features of $\mathbf{I}^S$. The security strength of the image feature is mainly determined by the difference value displacement. Therefore, the security strength of order difference image features can be represented as $log_2(val_{sum_r}!) + log_2(val_{sum_g}!) + log_2(val_{sum_b}!)bit$, and the security strength of disorder difference image features can be represented as $3 * log_2(imgsize!) + log_2(val_{sum_r}!) + log_2(val_{sum_g}!) + log_2(val_{sum_b}!)bit$.

- **Security of the trapdoor.** Simulator $S$ simulates a query image $\mathbf{I}^S{}_q$. The simulator $S$ knows the size of the query image, so it can simulate a query image $\mathbf{I}^S{}_q$ with the same pixel number as real query image $\mathbf{I}_q$. The user encrypts the query image as the image owner dose, so it has the same security strength as image encryption. Specific analysis is no longer expounded.

### *5.2. Security under the KBP Model*

In addition to the previously mentioned information leakage, the statistical characteristics of plaintext images may be inferred by the ciphertext images. The pixel values of each color component have a range of $[0, 255]$, and the theoretical difference values have a range of $[-255, 255]$, i.e., simulator $S$ needs to solve 500! sequences for color permutation encryption. However, some difference values will not occur in an image, and the number of resolved sequences is reduced. Taking the Lena standard gray image as an example, we calculate the gray order difference matrix (GODM) and gray disorder difference matrix (GDDM) according to Section 4.1. The difference distributions of GODM and GDDM are shown in Figure 4, Figure 5, respectively. The difference values of GODM have a range of $[-150, 150]$, and the difference values of GDDM have a range of $[-200, 200]$. Compared with theoretical sequences, the number of resolved sequences is greatly reduced.

Figure 4 presents as the Laplasse distribution, i.e., the values are centered around 0. Under this distribution, the attacker $S$ can easily judge the original value corresponding to the occur frequency. Hence, the color value replacement encryption algorithm will be weakened. Figure 5 smoothes this distribution and improves security to some extent.

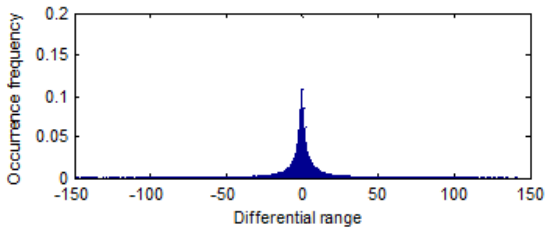



**Figure 4.** The order difference value distribution of the Lena.jpg.

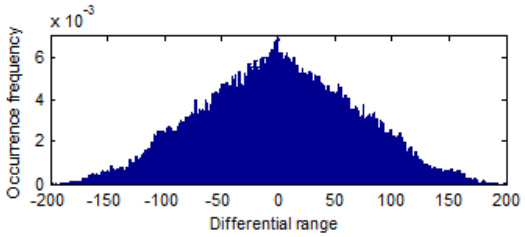



**Figure 5.** The disorder difference value distribution of the Lena.jpg.

## 6. Experimental Results

This section shows the experimental results of encryption effectiveness and retrieval accuracy. The scheme is implemented with MatLab 2014a. The simulation is conducted on a computer with Intel Core CPU 2.50 GHZ and 16 G memory. All the experiments in this paper are based on the INRIA Holidays database [24]. The image database contains 1491 images in 500 classes. The first images of each category are grouped into a query images set.

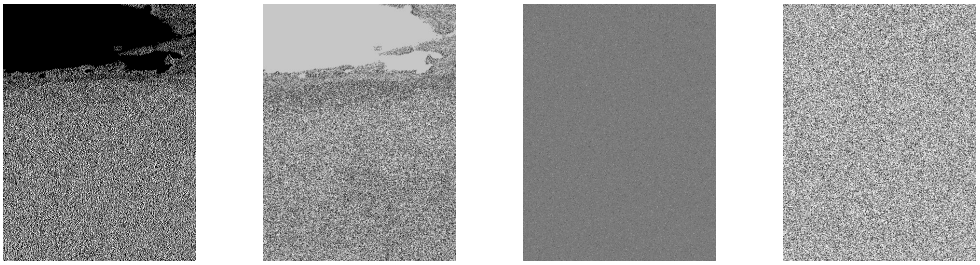
### *6.1. Effectiveness of Image Encryption*

Figure 6 shows the R, G, B components of the first image in the INRIA Holidays database.

(**a**) (**b**) (**c**) (**d**)

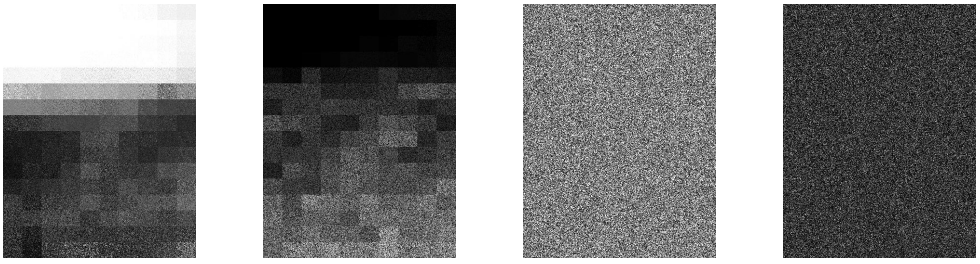
**Figure 6.** The first image in the INRIA Holidays database and the related R, G, B components. (**a**) Original image; (**b**) R component; (**c**) G component; (**d**) B component.

The proposed scheme calculates the difference matrix of three components and encrypts them. Taking the R component as an example, we show difference value replacement, position scrambling and overlay effects of order difference and disorder difference in Figures 7 and 8, respectively. Merge encryption results of three RGB components and the final encryption image are shown in Figure 9.



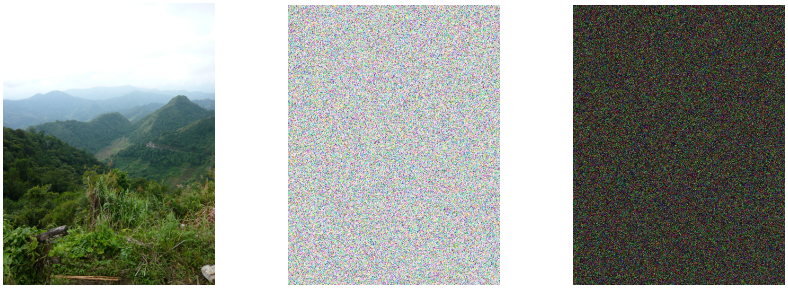
(**a**) (**b**) (**c**) (**d**)

**Figure 7.** The procedure of order differential encryption. (**a**) Order difference of R; (**b**) Value replacement; (**c**) Position permutation; (**d**) Encrypted R.



(**a**) (**b**) (**c**) (**d**)

**Figure 8.** The procedure of disorder differential encryption. (**a**) Block disorder difference; (**b**) Value replacement; (**c**) Position permutation; (**d**) Encrypted R.



(**a**) (**b**) (**c**)

**Figure 9.** The encrypted results. (**a**) Original image; (**b**) Encrypted result of order difference; (**c**) Encrypted result of disorder difference.

### *6.2. Retrieval Accuracy*

In our experiments, mean average precision (mAP) is used to measure the retrieval accuracy.

On analysis of the encrypted disorder difference histogram-based CBIR scheme (EDDH-CBIR), the size of image block is probably an important parameter affecting retrieval precision. The experimental results of different block sizes are shown in Table 3.

**Table 3.** The mean average precision (mAP) (%) of different parameters for the encrypted disorder difference histogram-based content-based image retrieval scheme (EDDH-CBIR).

| Block size | $50 \times 50$ | $100 \times 100$ | $200 \times 200$ | $500 \times 500$ |
|---|---|---|---|---|
| mAP (%) | 50.761 | 51.436 | 49.075 | 48.013 |

The proposed EDH-CBIR is divided into two sub-schemes: EODH-CBIR and EDDH-CBIR. Some other contrast experiments are carried out to compare the accuracy of the proposed method. The contrast experiments in the ciphertext domain contain the encrypted color histogram-based CBIR scheme (ECH-CBIR) and the global disorder difference histogram-based CBIR scheme (GDDH-CBIR). The contrast experiments in the plaintext domain contain the order difference histogram-based CBIR scheme (ODH-CBIR) and the disorder difference histogram-based CBIR scheme (DDH-CBIR). The mAPs of all the mentioned schemes are shown in Table 4. Experimental results show that the EDH-CBIR is indeed advantageous, and can obtain the comparable accuracy in the plaintext domain.

**Table 4.** The retrieval accuracies of different schemes. EODH-CBIR: encrypted order difference histogram-based CBIR scheme; ECH-CBIR: encrypted color histogram-based CBIR scheme; GDDH-CBIR: global disorder difference histogram-based CBIR scheme; ODH-CBIR: order difference histogram-based CBIR scheme; DDH-CBIR: disorder difference histogram-based CBIR scheme.

| **Schemes** | **EODH-CBIR** | **EDDH-CBIR** | **ECH-CBIR** | **GDDH-CBIR** | **ODH-CBIR** | **DDH-CBIR** |
|---|---|---|---|---|---|---|
| mAP (%) | 49.923 | 51.436 | 47.865 | 45.787 | 49.923 | 51.436 |

### *6.3. Efficiency*

Efficiency is a significant measurement standard, and it includes the time consumptions of image encryption, index construction, and image searching. For comparison, this section considers the contrast experiments in the ciphertext domain.

- *The time consumption of image encryption.* The encryption process of ECH-CBIR includes value replacement and position scrambling. The encryption processes of EODH-CBIR and GDDH-CBIR

include the difference matrix calculation, difference value replacement, and pixel scrambling. EDDH-CBIR includes the block difference matrix calculation, the difference value replacement, and pixel permutation. The time consumptions of image encryptions of all above schemes are shown in Figure 10.

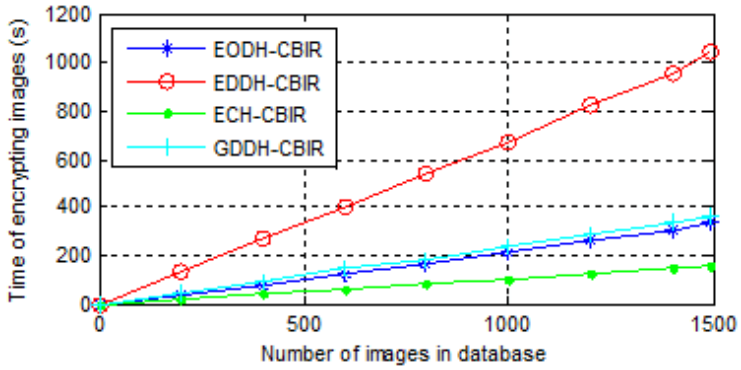



**Figure 10.** The time consumption of image encryption.

- *The time consumption of index construction.* A linear index is built for all the schemes so as to observe them more intuitively. Time consumption actually includes feature extraction and indexing, and results of three scheme are shown in Figure 11.

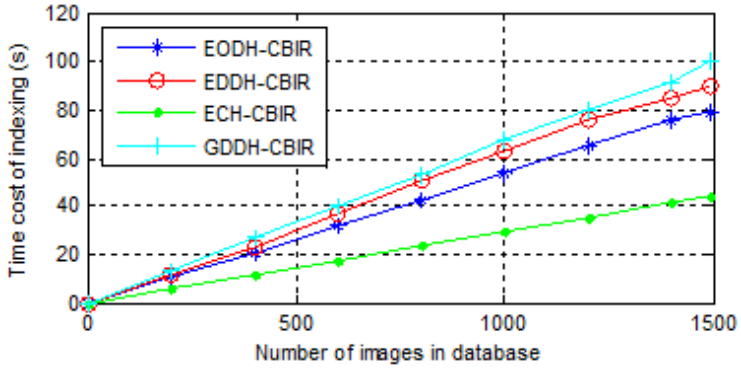



**Figure 11.** The time consumption of index construction.

- *The time consumption of image retrieval.* When the cloud server receives the user's trapdoor, it searches the index for the $k$ most similar images. The index designed in this paper is a linear one, so the retrieval time is only related to the length of feature vectors. The time consumption of mentioned schemes are shown in Figure 12.

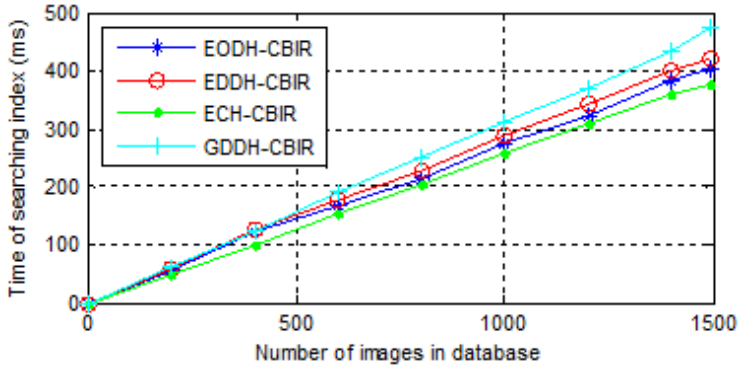



**Figure 12.** The time consumption of image retrieval.

## 7. Conclusions

In this paper, a secure CBIR scheme is proposed by using encrypted difference features. The scheme encrypts the image by difference matrix calculation, difference value replacement, and difference position scrambling. On the basis of this scheme, we compare it with the ECH-CBIR scheme, the GODH-CBIR scheme, the ODH-CBIR scheme and the DDH-CBIR scheme, and the experiments show that our encrypted difference histogram feature has advantages. However, both the EDDH-CBIR and the EODH-CBIR scheme have the problem of security risks under the KBP model. Future work will focus on more efficient encryption methods to improve the security of the EDH-CBIR scheme.

**Acknowledgments:** This work is supported by the NSFC (61672294, 61601236, U1536206, 61502242, 61572258, U1405254, 61373133, 61373132, 61232016), BK20150925, the Six peak talent project of Jiangsu Province (R2016L13), NRF-2016R1D1A1B03933294, CICAEET, and the PAPD fund. Zhihua Xia is supported by the BK21+ program from the Ministry of Education of Korea.

**Author Contributions:** Z.X. conceived and designed the experiments; D.L. performed the experiments; X.S. analyzed the data; J.S. contributed reagents/materials/analysis tools; D.L. wrote the paper.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Smeulders, A.W.M.; Worring, M.; Santini, S.; Gupta, A.; Jain, R. Content-based image retrieval at the end of the early. *IEEE Trans. Pattern Anal. Mach. Intel.* **2000**, *22*, 1349–1380.
2. Yong, R.; Huang, T.S.; Chang, S.F. Image Retrieval: Current Techniques, Promising Directions, and Open Issues. *J. Vis. Commun. & Image Represent.* **1999**, *10*, 39–62.
3. Rui, Y.; Huang, T.S.; Ortega, M.; Mehrotra, S. Relevance feedback: A power tool for interactive content-based image retrieval. *IEEE Trans. Circuits and Syst. Video Technol.* **1998**, *8*, 644–655.
4. Liu, Y.; Zhang, D.; Lu, G.; Ma, W.Y. A survey of content-based image retrieval with high-level semantics. *Pattern Recognit.* **2007**, *40*, 262–282.
5. Akgül, C.B.; Rubin, D.L.; Napel, S.; Beaulieu, C.F.; Greenspan, H.; Acar, B. Content-Based Image Retrieval in Radiology: Current Status and Future Directions. *J. Digit. Imaging* **2011**, *24*, 208–222.
6. Xia, Z.; Wang, X.; Sun, X.; Wang, Q. A Secure and Dynamic Multi-keyword Ranked Search Scheme over Encrypted Cloud Data. *IEEE Trans. on Parallel & Distrib. Syst.* **2016**, *27*, 340–352.
7. Dong, B.; Liu, R.; Wang, W.H. PraDa: Privacy-preserving Data-Deduplication-as-a-Service. In Proceeding of the 23rd ACM International Conference on Information and Knowledge Management, Shanghai, China, 3–7 November 2014; pp. 1559–1568.
8. Lindell, Y.; Pinkas, B. Secure Multiparty Computation for Privacy-Preserving Data Mining. *J. Priv. Confid.* **2013**, *25*, 761–766.
9. Curtmola, R.; Garay, J.; Kamara, S.; Ostrovsky, R. Searchable symmetric encryption: improved definitions and efficient constructions. In Proceedings of the ACM Conference on Computer and Communications Security, Alexandria, VA, USA, 30 October–3 November 2006; pp. 79–88.
10. Kuzu, M.; Islam, M.S.; Kantarcioglu, M. Efficient Similarity Search over Encrypted Data. In Proceedings of the IEEE International Conference on Data Engineering, Arlington, VA, USA, 1–5 April 2012; pp. 1156–1167.
11. Hahn, F.; Kerschbaum, F. Searchable Encryption with Secure and Efficient Updates. In Proceedings of the ACM Sigsac Conference on Computer and Communications Security, Scottsdale, AZ, USA, 3–7 November 2014; pp. 310–320.
12. Yuan, X.; Wang, X.; Wang, C.; Squicciarini, A.; Ren, K. Enabling Privacy-Preserving Image-Centric Social Discovery. In Proceedings of the IEEE International Conference on Distributed Computing Systems, Hsinchu, Taiwan, 30 June–3 July 2014; pp. 198–207.
13. Weng, L.; Amsaleg, L.; Morton, A.; Marchand-Maillet, S. A Privacy-Preserving Framework for Large-Scale Content-Based Information Retrieval. *IEEE Trans. Inf. Forensics & Secur.* **2014**, *10*, 152–167.
14. Lu, W.; Swaminathan, A.; Varna, A.L.; Wu, M. Enabling search over encrypted multimedia databases. In *Media Forensics and Security I*; SPIE: Bellingham, WA, USA, 2009; p. 725418.

15. Lu, W.; Varna, A.L.; Swaminathan, A.; Wu, M. Secure image retrieval through feature protection. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, Taipei, Taiwan, 19–24 April 2009; pp. 1533–1536.
16. Lu, W.; Varna, A.L.; Wu, M. Confidentiality-Preserving Image Search: A Comparative Study Between Homomorphic Encryption and Distance-Preserving Randomization. *IEEE Access* **2014**, *2*, 125–141.
17. Xia, Z.; Zhu, Y.; Sun, X.; Qin, Z.; Ren, K. Towards Privacy-preserving Content-based Image Retrieval in Cloud Computing. *IEEE Trans. Cloud Comput.* **2015**, doi:10.1109/TCC.2015.2491933.
18. Cheng, H.; Zhang, X.; Yu, J.; Li, F. Markov Process Based Retrieval for Encrypted JPEG Images. In Proceedings of the International Conference on Availability, Reliability and Security, Toulouse, France, 24–28 August 2015; pp. 417–421.
19. Xu, D.; Xie, H.; Yan, C. Triple-Bit Quantization with Asymmetric Distance for Image Content Security. *Mach. Vis. Appl.* **2017**, *28*, 1–9.
20. Bellafqira, R.; Coatrieux, G.; Bouslimi, D.; Quellec, G. Content-based image retrieval in homomorphic encryption domain. In Proceeding of the International Conference of the IEEE Engineering in Medicine and Biology Society, Milan, Italy, 25–28 August 2015; pp. 2944–2947.
21. Bellafqira, R.; Coatrieux, G.; Bouslimi, D.; Quellec, G.; Bellafqira, R.; Coatrieux, G.; Bouslimi, D.; Quellec, G.; Quellec, G.; Bellafqira, R. An end to end secure CBIR over encrypted medical database. In Proceeding of the International Conference of the IEEE Engineering in Medicine & Biology Society, Oralndo, FL, USA, 16–20 August 2016; p. 2537.
22. Ferreira, B.; Rodrigues, J.; Leitao, J.; Domingos, H. Practical Privacy-Preserving Content-Based Retrieval in Cloud Image Repositories. *IEEE Trans. Cloud Comput.* **2017**, *PP*, 1.
23. Canetti, R. Universally composable security: a new paradigm for cryptographic protocols. In Proceedings of the IEEE Symposium on Foundations of Computer Science, Las Vegas, NV, USA, 8–11 October 2001; p. 136.
24. Jegou, H.; Douze, M.; Schmid, C. Hamming Embedding and Weak Geometric Consistency for Large Scale Image Search. In Proceedings of the European Conference on Computer Vision, Marseille, France, 12–18 October 2008; pp. 304–317.