Secure outsourcing SIFT: Efficient and Privacy-preserving Image Feature Extraction in the Encrypted Domain

Linzhi Jiang, *Student Member, IEEE,* Chuxiang Xu, *Member, IEEE,* Xiaofang Wang, *Student Member, IEEE,* Bo Luo, *Member, IEEE,* and Huaqun Wang

Abstract—Multimedia data needs huge storage space, and application of multimedia data needs powerful capability of computing. Cloud computing can help owner of multimedia data to deal with it. But, multimedia data on cloud may reveal privacy of data owner, such as sex, hobbies, address, looks, and so on. Data owner can encrypt multimedia data for confidentiality before uploading it to cloud. However, encrypted multimedia data makes its utilization difficult. In this paper, we firstly discover pre-existing schemes have problems of huge storage space, security and low efficiency due to their inefficient and insecure algorithms. Then, we propose an effective and practical privacy-preserving scale-invariant feature transform (SIFT) scheme for encrypted image. It uses leveled homomorphic encryption based on our new encoding schemes, our new homomorphic comparison, division and derivative encryption. Our new secure SIFT scheme can realize higher computing efficiency, greatly reduce communication cost and interactive times between user and server, and perform correct feature point detection, accurate feature point description and image matching. We evaluate security and efficiency of our new secure SIFT scheme, and compare our new secure SIFT scheme with other schemes in detail. The result shows that it is the closet to the original SIFT algorithm.

Index Terms—Scale-invariant feature transform, Feature extraction, Privacy-preserving, Leveled homomorphic encryption, Security

1 Introduction

With the development of cloud computing, server provides huge storage space and powerful computing capacity. For users and enterprises, server not only stores a variety of text files, but also stores a variety of multimedia files (images, voices and videos). Cloud service provides us with convenience, but server is not always credible. When malicious users access to server, data on server may expose privacy of users and enterprises [1], [11], such as sex of user, hobbies, home address and workplace, looks, salary, and so on. Privacy and security of cloud computing have become research hotspots. To preserve privacy and insure security for cloud computing, one of the best opinions is to encrypt data. Then, the encrypted data is uploaded to server by users. All operations of server are performed on encrypted data. Server can perform computing on encrypted data without decryption [2], [3]. Many schemes support privacy-preserving keyword search [4], [5], [6] on text data. Server does not know contents of queries and the returned results on queries. Privacy of users and queries is preserved.

- *Linzhi Jiang is with School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu, China, 611731. School of Finance and Mathematics, West AnHui University, Lu An, China, 237012. E-mail: linzjiang@hotmail.com.*
- *Chunxiang Xu is with the Center for Cyber Security, School of Computer Science and Engineering, University of Electronic Science and Technology of China.*
- *Xiaofang Wang is with School of Telecommunications Engineering, Xidian University, China.*
- *Bo Luo is with Department of Electrical Engineering and Computer Science, The University of Kansas, Lawrence, KS, 66049.*
- *Huaqun Wang is with Nanjing University of Posts and Telecommunications, China.*

Manuscript received April 19, 2005; revised August 26, 2015.

In addition to text data, there is a large amount of multimedia data stored on server. Multimedia data includes images, voices and videos, which may include much sensitive information about users. In social networking services (SNS), registered users may upload their images to server to share with their friends. Images may contain family members and residential environments. When users chat with their friends (or family members) by chatting software, voices and videos may be stored on server of service provider. According to images, malicious attackers can perform location analysis by tools (such as Google Maps). Any user may download your voices to analyze your voice features, and pretend to be you by super voice changer. Malicious users also can perform feature recognition through images and videos. Therefore, in order to preserve privacy, users can encrypt multimedia data before uploading it to server. But, encrypted multimedia data not only preserves privacy, but also makes its utilization difficult.

1.1 Related Work

Recently, people have extended privacy-preserving schemes from access and query on encrypted text [4], [5], [6] to secure multimedia data search [7], [8], [9], [10].

Hsu et al. [13], [14] firstly addressed problem of secure scale-invariant feature transform (SIFT) algorithm [12] in the encrypted domain. Amir et al. [15] pointed out the schemes of [13], [14] were computationally difficult to implement, and were insecure on preserving privacy of image. Qin et al. [19] constructed a privacy-preserving feature detection scheme with encrypted data comparison based on order-preserving encryption [22]. But, their scheme had high communication costs by multiple rounds of interaction between

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2017.2751476, IEEE Transactions on Dependable and Secure Computing

multiple servers. The scheme of [25] pointed out that the scheme of [19] could not preserve privacy of content on image. The scheme of [20] had high communication costs on Garbled circuit protocol [22]. Paillier [42] homomorphic encryption used in the scheme of [20] did not meet requirements of encrypted image processing. Wang et al. [23], [24] provided the scheme of privacy-preserving feature extraction of image with somewhat homomorphic encryption (SHE). Their scheme performed batch homomorphic evaluation on encrypted data with packaging technology named Single Instruction Multiple Data (SIMD). The scheme of [25] designed two novel secure interactive protocols (BSMP and BSCP) to compare multiple pairs of integers for privacy-preserving outsourcing feature extraction. Somewhat homomorphic encryption of [23], [24], [25] was inefficient, and interactive protocols of BSMP and BSCP had high communication costs between servers. At the same time, above schemes can not efficiently eliminate unstable key points and edge effect.

According to above analysis, the previous schemes mainly have the following shortcomings: (1) The encryption scheme (the schemes of [23], [24], [25]) for image is inefficient. (2) Privacy-preserving SIFT algorithm is insecure (the schemes of Hsu et al. [13], [14] and Qin et al. [19]). (3) Privacy-preserving feature detection schemes have high communication costs between user and server or servers (the schemes of Hsu et al. [13], [14] and the schemes of [19], [25]). (4) The above schemes can not effectively eliminate unstable key points and edge effect (the schemes of Hsu et al. [13], [14] and the scheme of [25]). Based on above shortcomings, we propose a new scheme of privacy-preserving feature extraction based on SIFT algorithm with leveled homomorphic encryption (LHE). Our new scheme has higher efficiency and lower storage costs than the schemes of [23], [24], [25], and has lower communication costs than the schemes of [13], [14], [19], [23], [24], [25]. By eliminating unstable key points and edge effect, our new scheme maintains consistency with the original SIFT algorithm [12]. At the same time, our new scheme realizes confidentiality of pixel and content of image.

1.2 Our Contribution

To realize above goals, we encrypt image by LHE based on NTRU [29], [33], SIMD and our new encoding schemes. We denote SIMD LHE based on our new encoding schemes and NTRU as $\widetilde{E}$. Our new encoding methods convert a non-integer into an integer, and support server to perform SIMD LHE on circuit with non-trivial finite fields of characteristic two. $\widetilde{E}$ gains an advantage on homomorphic evaluation speed over pre-existing fully homomorphic encryption (FHE). $\widetilde{E}$ has shorter time of key generation, encryption and decryption, and has smaller ciphertext size than SHE from RLWE [25]. User encrypts image, and uploads encrypted image to server. Then, server performs secure privacy-preserving SIFT algorithm. We provide a new non-interactive leveled homomorphic comparison algorithm (LHCA) to realize encrypted data comparison. With the help of LHCA, we give a new scheme to detect non-edge stable key points through algorithm of difference-of-Gaussian (DoG) in the encrypted domain. DoG has a relative strong response to edge effect on image. Once key points fall on the edge of image, these points become unstable.

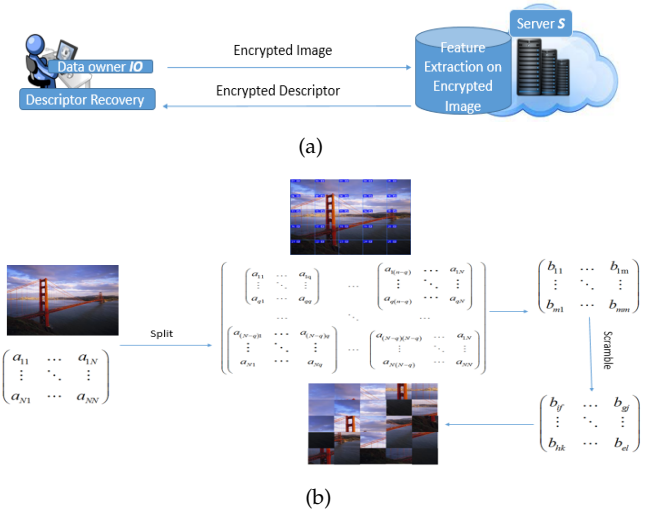


Fig. 1. (a) System model of our privacy-preserving SIFT algorithm. (b) Image split and scramble of our scheme.

We put forward our new schemes of leveled homomorphic division (LHD) and derivative algorithm (LHDA) on encrypted data. Combined our new LHD and LHDA with LHCA, we can eliminate unstable key points on the edge of image. Finally, we present a complete secure SIFT algorithm on encrypted image. Feature extraction and description of our new secure scheme preserve privacy of image in the encrypted domain. Our contribution mainly includes the following aspects:

1. We put forward encoding method for the fixed point real number and a new encoding method for LHE on the finite fields of characteristic two. Such encoding methods can convert a non-integer into an integer, and realize SIMD leveled homomorphic operation on multi-bit ciphertext with high computing efficiency.
2. We propose a new and effective LHCA for encrypted data. Server can realize homomorphic comparison on encrypted data. This insures LHCA is performed independently by server without user involved.
3. New LHD on encrypted data is presented. Our scheme can directly perform homomorphic division on encrypted data.
4. We for the first time present LHDA on encrypted data. It is the necessary operation for eliminating edge effect. In this way, server can eliminate unstable key points on the edge of image.
5. On the basis of our new LHCA, LHD and LHDA on encrypted data, we propose an effective and practical privacy-preserving SIFT algorithm in the encrypted domain.
6. We evaluate efficiency of our scheme, including feature points detection, accurate feature points description, image matching, and prove security of our scheme. At the same time, we compare our scheme with other schemes based on homomorphic encryption in detail.

2 Problem Statement

2.1 System Model

In our work, we mainly consider the following scenarios of outsourcing computing. We assume image owner (IO) has a large number of sensitive images (e.g. medicine images) that need to be outsourced to server (S) for storage and performing privacy-preserving SIFT algorithm. In order to preserve privacy, IO splits and scrambles image, then encrypts joint image and uploads encrypted joint image to S. S performs SIFT algorithm for detecting and describing image feature in the encrypted domain. After performing privacy-preserving SIFT algorithm in the encrypted domain, S will return encrypted descriptors of feature points to IO. Then, IO reverts to real encrypted descriptors of feature points on encrypted image. Fig. 1(a) shows system model of our privacy-preserving SIFT algorithm.

2.2 Security Model

The scheme of [19] has pointed out that outsourcing image feature extraction with only one server would result in leakage of image content. But, our scheme can be performed with only one server S, because we split and scramble image before uploading image to S. Another justification is that our new LHCA algorithm can independently perform comparison on encrypted data.

We consider privacy of image content including pixel values, key point locations and extracted features from image [19].

In our security model, we assume server S is "honest-but-curious". Namely, S correctly executes homomorphic evaluation, LHD, LHDA, LHCA and secure SIFT algorithm. But, S tries to learn additional information from encrypted data and all of operations performed by it. In our new scheme, IO uploads encrypted joint image (being split and scrambled) to S. S performs all operations on data in the encrypted domain. Therefore, S only obtains the local relationship of encrypted data for split and scrambled block of image. Consequently, privacy of image content can be preserved from S.

Homomorphic encryption is malleable. We assume the uploaded data will not be changed by a man-in-the-middle attack, and S will not modify uploaded data.

3 Preliminaries

In this section, we firstly give an overview to SIFT algorithm in this paper. Then, the basic LHE based on NTRU [33] is presented.

3.1 Scale-Invariant Feature Transform (SIFT) Algorithm

SIFT algorithm has been widely used in the fields of computer vision and pattern recognition. Lowe [39] showed that primate could achieve robust object recognition by feature detection under 2 seconds. In [40], a method of combining multiple images of a 3D object into a single model is presented. Such scheme had non-rigid changes, and improved robustness of object recognition. The method of extracting distinctive invariant feature from image was presented in [12]. Features are invariant to image scale and rotation. The robust matching can be achieved through a substantial range of affine distortion and changes in illumination, additional noise and 3D viewpoint. The main aspects of SIFT algorithm in Appendix A.

3.2 LFHE Based on NTRU (LHEBN)

Rivest et al. [1] published first multiplicatively homomorphic public-key cryptosystem called RSA. Paillier cryptosystem [42] only supports additively homomorphic encryption. Liu et al. [27] proposed a framework for efficient and privacy-preserving outsourced calculation of rational numbers. These homomorphic encryption systems only support additively or multiplicatively homomorphic evaluation. Gentry [3] provides a breakthrough for FHE based on ideal lattice. Security of FHE from LWE or RLWE relies on learning with errors problem (LWE) [43], [44] or ring learning with errors problem (RLWE) [44], [45]. A series of work proved that FHE from RLWE had higher efficiency for application than FHE from LWE. FHE from RLWE supports additively and multiplicatively homomorphic evaluation. But, above schemes are not suitable for division of encrypted data.

NTRU [29] is a ring-based public key cryptosystem. It has the following advantages: easily created short keys, high encryption (decryption) speed and low memory requirement. A modified NTRU [34], which can be used to construct the scheme of multi-key FHE [31], has the same hardness as the RLWE problem. Based on NTRU, a scalable implementation of FHE [32] is built. Based on the modified version of NTRU and multi-key FHE, a FHE scheme [33] was achieved. For the scheme of [33], ciphertext is one polynomial ring element. A comparison between homomorphic encryption scheme FV [38] and YASHE [33] is presented in [37]. In [30], authors proved which ring based somewhat homomorphic encryption schemes (BGV [46] and YASH [33]) is best. LHE based on NTRU has the following characteristics: easily created short keys, small ciphertext, high encryption (decryption) speed, efficient homomorphic evaluation and low memory requirement. We choose LHE based on NTRU as the basic encryption scheme. The detailed LHE based on NTRU in Appendix B.

4 SIFT Algorithm in the Encrypted Domain

In this section, we firstly review state-of-art about SIFT algorithm in the encrypted domain. Then, we analyze the key questions and operations of SIFT algorithm in the encrypted domain.

4.1 State-of-Art about Secure SIFT Algorithm

Feature extraction and description based on secure SIFT algorithm is firstly addressed in [13], [14]. But, comparison protocol of encrypted data in [13], [14] is not efficient and secure. To compare $E(M_1, \tau_1)$ and $E(M_2, \tau_2)$, server is required to compute the distance A_{k_1} by equation $(A_{k_1}, \zeta_{k_1}) = argumin(E(M_1, \tau_1)g^{Inc} - E(T_i, \tau_1))$. Assuming that 10 thresholds are chosen at random and recommended primes $p, \flat$ have 1000 bits, then there are about 2^{2000} elements in the plaintext space $\mathbf{Z}_N$. The distance to the next threshold is average $\frac{2^{2000}}{2 \cdot 10} (> 2^{1995})$. Taking IBM

Sequoia as an example, one step in the computation of equation $((A_{k_1}, \zeta_{k_1}) = argumin(E(M_1, \tau_1)g^{Inc} - E(T_i, \tau_1)))$ could be performed for more than $\frac{2^{1995}}{16.32 \cdot 10^{15}} (> 2^{1941})$ seconds. This is far beyond existing computing power of human. At the same time, using a large number of thresholds would increase communication costs between client and server and computing overheads in the setup phase. Therefore, if the encrypted values are large, comparison protocol is infeasible. If the scheme takes the encrypted values from a smaller domain, complexity of comparison protocol is reduced. But, the scheme is not secure. In [13], [14], pixel value is from 0 to 255. Such value is much smaller than the plaintext space of Paillier [42]. The schemes of [13], [14] take thresholds from the same subspace. The same encrypted thresholds for comparison protocol make a curious server to break security of comparison protocol.

The scheme of Qin et al. [19] can not protect locations of key points from revealing to server. In [25], basic encryption scheme has larger ciphertext size. After two times homomorphic multiplicatively operations on encrypted data, one-bit plaintext becomes 50KB ciphertext. Encryption, homomorphic evaluation and decryption on encrypted image require more times. The secure interactive protocols (BSMP and BSCP) are performed with several rounds of interaction between servers. Therefore, BSMP and BSCP increase computing overheads and communication costs.

Another allimportant limitation of above schemes is that robust and distinctiveness of original SIFT algorithm [12] are destroyed by their algorithms. For original SIFT algorithm [12], low-contrast key points and unstable edge response points require to be eliminated. Elimination of these key points can enhance stability of matching and improve anti-noise ability. Above three schemes can not effectively eliminate unstable key points and edge effect on encrypted image.

4.2 Key Operations for Privacy-Preserving Outsourcing SIFT Algorithm

In this section, we introduce key operations for our privacy-preserving outsourcing SIFT algorithm. We achieve more efficient secure SIFT algorithm in the encrypted domain.

4.2.1 Encryption for Image

IO splits image and scrambles sub-images. Then, IO uploads encrypted joint image I (being split and scrambled $I_1, I_2, \cdots, I_W$) to S. In order to improve efficiency of algorithm and support more kinds of function operations, we take full advantage of $\widetilde{E}$. $\widetilde{E}(\cdot)$ represents encryption of a plaintext message. It has smaller ciphertext size. $\widetilde{E}$ has shorter time for homomorphic evaluation.

4.2.2 Extremum Detection in Scale Space

Scale space of image on server is $\widetilde{E}(L(x, y, \sigma))$. DoG on server is $\widetilde{E}(D(x, y, \sigma))$. Server performs extremum detection with LHCA on encrypted data in DoG space.

4.2.3 Location of the Key Point

The above obtained extreme points are not true extreme points. Usually, interpolation is used to get close to the true extreme points by our new LHDA and LHD on encrypted data. Then, server eliminates points with low response to DoG and points on the edge of image.

4.2.4 Determination of the Feature Point Direction

Making use of LHD, server can obtain argument and amplitude. After computing gradient with LHD, server uses histogram to obtain statistic on gradient direction and amplitude of pixels in the neighbourhood of feature point. With histogram of gradient direction, the maximum value of histogram is found by LHCA. Then, server determines direction of feature point.

4.2.5 Feature Point Description

Description of Gaussian image gradient in the neighbourhood of feature points is feature point descriptor of SIFT algorithm. Server rotates image to the main direction, and generates a feature vector by homomorphic evaluation. Finally, server performs normalization of feature vector.

4.2.6 Efficiency and Security

IO takes advantage of $\widetilde{E}$ to improve efficiency for encryption and decryption. Server makes use of homomorphic evaluation, LHCA, LHD and LHDA on encrypted image. In such way, server can independently perform privacy-preserving SIFT algorithm excepting image encryption. $\widetilde{E}$ has smaller ciphertext size, higher computing efficiency and lower communication costs. At the same time, our new privacy-preserving SIFT algorithm enhances stability of image matching, and improves anti-noise ability. Our scheme protects privacy of the original image from adversary in the mean time.

5 Related Encryption Algorithm

In this section, we provide $\widetilde{E}$ based on our new encoding methods, SIMD and NTRU. SIMD technology can speed up operation of SIFT algorithm on encrypted image. Then, we provide LHCA, which helps server to detect key point and perform key point location. Finally, we introduce our new LHD and LHDA, which helps server to eliminate unstable key points on the edge of image.

5.1 Encoding the fixed point real number

$\widetilde{E}$ supports computing for integers and integral coefficient polynomials. In order to performing LHD and LHDA, we truncate raw data to a fixed point real number. This truncation is to meet the requirements of computational accuracy. Then, we encode it as integer.

Fixed point real number includes signed integer, unsigned integer and rational number. Rational number multiplies a certain value. It is expanded by a certain multiple. Therefore, it becomes signed integer (or unsigned integer) by scaled numeric formats. For LHD and LHDA, two integers of scaled numeric formats use same scaling factor. Scaled numeric formats has binary accuracy and insures accuracy of computing. Unsigned integer is converted into binary number. Its complement is itself. We convert signed integer into two's complement form, which eliminates substraction and guarantees correctness on computing. At the

same time, the sign bit takes part in operation as an effective part. Thereby, we simplify the operation rules, and improve computational accuracy.

5.2 SIMD LHE Based on Our New Encoding and NTRU

In [47], [48], authors proved that some previous FHE schemes can partition the plaintext space into a vector of plaintext slot by Chinese Remainder Theorem (CRT). Leveled homomorphic encryption based on NTRU works over a polynomial ring $R = Z[Y]/(\Phi_d(Y))$, where $\Phi_d(Y)$ is cyclotomic polynomial. Ciphertext is a polynomial, which can be viewed as a coefficient vector. $R_q = R/_qR$ denotes ciphertext space, $R_t = R/_tR$ $(1 < t < q)$ denotes the message space, and module q is an integer. The plaintext and ciphertext are all polynomials with integer coefficient module t.

In order to support homomorphic circuit evaluation, we encode data in the finite field $GF(2^\jmath)$, which has $2^\jmath$ elements. The filed $GF(2^\jmath)$ is comprised of the polynomials with degree $\jmath - 1$. Coefficients of these polynomials are over the field Z_2. These polynomials are denoted as $a_{\hbar-1}y^{\hbar-1}+\cdots+a_1y^1+a_0y^0$, where $a_i \in \{0,1\}$. A single integer can be represented as an entire polynomial in $GF(2^\jmath)$, and individual bit of integer is the coefficient of polynomial. Polynomial addition in $GF(2^\jmath)$ is just the corresponding coefficients addition modulo 2. The multiplication of polynomial may be not closed. If $\hbar > 1$, degree of the product on polynomials may exceed $\hbar - 1$. Resorting to modular, we can insure multiplication is closed.

$\Phi_d(Y)(\in Z[Y])$ is cyclotomic polynomial, and d is a positive integer. We define $R = Z[Y]/(\Phi_d(Y))$, where $d = 2^k$ and $n = 2^k - 1$. R is a polynomial ring. The plaintext space is defined as $R_{2^p} = R/_{2^p}R$ $(1 < 2^p < q)$. The ciphertext space is defined as $R_q = R/_qR$, where modulo q is an integer. $\Phi_d(Y)$ has primitive n-th root of unity $\bmod 2^p$. $\Phi_d(Y)(\in F_{2^p}[Y])$ can be decomposed into as $\Phi_d(Y) = \Pi_i^{\Im}F_i(y)$, where $F_i(y)$ is irreducible and has the same degree ζ. For some d, we can get $\Im = \frac{\psi(d)}{\zeta}$, which is the number of slot. Then, we have isomorphism: $R_{2^p} = Z_{2^p}[Y]/(\Phi_d(Y)) \cong R_{2^p}[Y]/(F_1(y)) \otimes \cdots \otimes R_{2^p}[Y]/(F_\Im(y)) \cong F_{2^{p\,d}} \otimes \cdots \otimes F_{2^{p\,d}}$. Consequently, we can batch $\Im$ independent plaintexts $(M_0, \cdots, M_{\Im-1})$ into the unique element in $R_{2^p} = Z_{2^p}[Y]/(\Phi_d(Y))$. Thus, we have $\Im$ independent plaintext slots in a single ciphertext. SIMD technology performs the same function on $\Im$ inputs by just one time. Namely, it packs the palintext (Bit-slice) into slots and just performs the function once. c_1 is the ciphertext of plaintexts $(M_0, \cdots, M_{\Im-1})$, and c_2 is the ciphertext of plaintext $(M'_0, \cdots, M'_{\Im-1})$, where $c_1 = \widetilde{E}(M_0, \cdots, M_{\Im-1})$ and $c_2 = \widetilde{E}(M'_0, \cdots, M'_{\Im-1})$. We can perform SIMD homomorphic evaluation: $c_1+c_2 = \widetilde{E}(M_0+M'_0, \cdots, M_{\Im-1}+M'_{\Im-1})$, $c_1 \cdot c_2 = \widetilde{E}(M_0 \cdot M'_0, \cdots, M_{\Im-1} \cdot M'_{\Im-1})$.

5.3 LHCA on Encrypted Data

One key step for SIFT algorithm is extremum detection, which refers to comparison computation on data. Privacy-preserving SIFT algorithm performs all operations in the encrypted domain. In order to perform extremum detection in the encrypted domain, the first problem we need to solve is comparison computation on encrypted data.

According to $\widetilde{E}$, let M_1 and M_2 be two plaintexts. $\widetilde{E}(M_i)$ denotes encryption of M_i, where $i = 1, 2$. We can get $c_1 = \widetilde{E}(M_1) = [\lfloor q/t \rfloor [M_1]_t + e_1 + hs_1]_q (\in R)$ and $c_2 = \widetilde{E}(M_2) = [\lfloor q/t \rfloor [M_2]_t + e_2 + hs_2]_q (\in R)$, where t is 2^p. Model of comparison on encrypted data is described as following: IO uploads the encrypted data $\widetilde{E}(M_i)$ to server, server obtains $M_1 > M_2$ or $M_1 < M_2$.

Comparison Algorithm on Encrypted Data:
Input: IO encrypts two messages M_1 and M_2, and uploads ciphertext to server.
Output: Server S outputs $M_2 > M_1$ or $M_2 < M_1$.
(1) IO encrypts two messages M_1 and M_2 in the following method. $c_1 = \widetilde{E}(M_1) = [\lfloor q/t \rfloor [M_1]_t + e_1 + hs_1]_q (\in R_q)$, $c_2 = \widetilde{E}(M_2) = [\lfloor q/t \rfloor [M_2]_t + e_2 + hs_2]_q (\in R_q)$ and $c_3 = \widetilde{E}(M_1) = [\lfloor q/t \rfloor [M_1]_t + e_2 + hs_2]_q (\in R_q)$ or $\widetilde{c}_3 = \widetilde{E}(M_2) = [\lfloor q/t \rfloor [M_2]_t + e_1 + hs_1]_q (\in R_q)$, where t is 2^p. After encryption of data, IO uploads ciphertexts to server.
(2) Server computes $c_2 - c_3$ or $\widetilde{c}_3 - c_1$. If $c_2 - c_3 > 0$ or $\widetilde{c}_3 - c_1 > 0$, server can obtain $M_2 > M_1$; if $c_2 - c_3 < 0$ or $\widetilde{c}_3 - c_1 < 0$, server can obtain $M_2 < M_1$.
(3) Server outputs $M_2 > M_1$ or $M_2 < M_1$.

5.4 LHD on Encrypted Data

For plaintext message m $(m \in R_{2^p})$, we can denote it as polynomial $(m(x))$ by above encoding schemes. $\widetilde{E}(m(x))$ denotes encryption of $m(x)$. We make use of Fast Fourier Transform (FFT) algorithm to obtain LHD for our scheme.

Encrypted FFT Algorithm ($\widetilde{E}FFT_{\omega,\Re}(E_h(m(x)))$):
Input: $< \widetilde{E}_h(m_0(x), \widetilde{E}_h(m_1(x), ..., \widetilde{E}_h(m_{\Re-1}(x) >$, where $m(x) = \Sigma_{i=0}^{\Re-1} m_i x^i$. Let $\Re = 2^k \in N(k \geq 1)$ be the power of a primitive $\Re - th$ root of unity $\omega \in R$. Let h be the public key for $\widetilde{E}$.
Output: $< \widetilde{E}_h(m(1)), \widetilde{E}_h(m(\omega)), ..., \widetilde{E}_h(m(\omega^{\Re-1})) >$.
1. If $\Re = 2$, return

$$\widetilde{E}_h(m_0(x)) + \widetilde{E}_h(m_1(x)) \cdot \omega.$$

2. For $0 \leq j \leq \Re/2$, compute

$$\widetilde{E}_h(r_{0,j}) = \widetilde{E}_h(m_j(x)) + \widetilde{E}_h(m_{j+\Re}/2(x)).$$

3. For $0 \leq j \leq \Re/2$, compute

$$\widetilde{E}_h(r_{1,j}) = \widetilde{E}_h(m_j(x)) - \widetilde{E}_h(m_{j+\Re}/2(x)).$$

4. For $0 \leq j \leq \Re/2$, compute

$$\widetilde{E}_h(r^*_{1,j}) = \widetilde{E}_h(r_{1,j}) \cdot \omega^j.$$

5. Let

$$r_0 =< r_{0,0}, r_{0,1}, ..., r_{0,\Re/2-1} >,$$
$$r^*_1 =< r^*_{1,0}, r^*_{1,1}, ..., r^*_{1,\Re/2-1} > .$$

6. Compute two encrypted vectors

$$V_0 =< v_{0,0}, v_{0,1}, ..., v_{0,\Re/2-1} >,$$
$$V^*_1 =< v^*_{1,0}, v^*_{1,1}, ..., v^*_{1,\Re/2-1} >,$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2017.2751476, IEEE Transactions on Dependable and Secure Computing

$V_0 \leftarrow \widetilde{E} - FFT_{\omega^2,\Re/2}(\widetilde{E}_h(v_0)), V_1 \leftarrow \widetilde{E}FFT_{\omega^2,Re/2}(\widetilde{E}_h(v_1^*))$. 7. Compute

$$\widetilde{E}_h(r(x)) = \widetilde{E}_h(m(x)) - \widetilde{E}PolyMult_{\widetilde{\omega},n'}(\widetilde{E}_h(n(x)), g(x)).$$

8. Output $\widetilde{E}_h(r(x))$.

Two integers $\widetilde{x}$ and $\widetilde{y}$ $(\widetilde{x}, \widetilde{y} > 0)$ are ε-bit integers. For $\widetilde{E}$, $\widetilde{E}(\widetilde{x})$ denotes encryption of $\widetilde{x}$, and $\widetilde{E}(\widetilde{y})$ denotes encryption of $\widetilde{y}$. By the above-mentioned new encoding methods, we compute $\widetilde{y}' = \widetilde{y} \cdot 10^{j+1}$. Therefore, $u' = 10^{j+1}(1 - (\frac{\widetilde{y}}{2^j}))$. $\widetilde{E}(10^{j+1} \cdot 2^{-(j+1)}(1 + u + u^2 + ... + u^\varepsilon)) = \widetilde{E}(5^j(1 + u + u^2 + ... + u^\varepsilon))$. According to the properties of additively and multiplicatively homomorphic encryption, we achieve $\widetilde{E}(5^{j+1}(1+u+u^2+...+u^\varepsilon)) = \widetilde{E}(5^{j+1})(\Sigma_{i=0}^{\varepsilon}\widetilde{E}(u^i))$. Based on CRT, we can compute $\widetilde{E}(u^\varepsilon)$. $\widetilde{E}(5^{j+1})(\Sigma_{i=0}^{\varepsilon}\widetilde{E}(u^i))$ is to be getted. We obtain $\widetilde{E}(10^{j+1} \cdot \frac{1}{\widetilde{y}})$. At last, $\widetilde{E}(\frac{1}{\widetilde{y}})$ can be computed by $\widetilde{E}PolyDiv$. We can compute $\widetilde{E}(\widetilde{x} \cdot \frac{1}{\widetilde{y}})$, which is the encryption of $(\widetilde{x} \cdot \frac{1}{\widetilde{y}})$.

7. Return

$$< v_{0,0}, v_{1,0}, v_{0,1}, v_{1,1}, ..., v_{0,(\Re-1)}, v_{1,(\Re-1)} > .$$

Encrypted Interpolation on Powers of $\Re$-th Root of Unity ($\widetilde{E}.Interpru$):
Input: Let $\Re = 2^k \in N(k \geq 1)$ be the power of a primitive $\Re - th$ root of unity $\omega \in R$. Let h be the public key for $\widetilde{E}$. $\widetilde{E}_h(v)$ denotes encryption of v, where $\widetilde{E}_h(v) =< \widetilde{E}_h(v_0), \cdots, \widetilde{E}_h(v_{\Re-1}) >$ and $\omega \in R^{\Re}$.
Output: The encrypted polynomial $\widetilde{E}_h(m(x))$, where $m(\omega^i) = v_i (0 \leq i \leq \Re)$.
1.Compute $\omega^{-1}, \omega^{-2}, \cdots, \omega^{-(\Re-1)}$.
2.Compute and return $1/\Re \cdot \widetilde{E}FFT_{\omega^{-1},\Re}(\widetilde{E}_h(v))$.

Encrypted Polynomial Multiplication Algorithm $(\widetilde{E}PolyMult_{\omega,\Re})(\widetilde{E}_h(m(x)), g(x))$:
Input: Encrypted polynomial $\widetilde{E}(m(x)) =< \widetilde{E}_h\ (m_0)(x), \widetilde{E}_h\ (m_1)(x), \cdots, \widetilde{E}_h(m_{d_1})(x) >$. $g(x) = \Sigma_{i=0}^{d_2} g_i x^i$ is plaintext polynomial. ω is a primitive $\Re - th$ root of unity, where $\Re = 2^k$ and $d_1 + d_2 < \Re$. Let h be the public key for $\widetilde{E}$.
Output: Encryption of the product polynomial $z(x) = m(x)\ g(x)$.
1. Compute

$$< \widetilde{E}_h(m(1)), \cdots, \widetilde{E}_h(m(\omega^{\Re-1})) > \leftarrow \widetilde{E}FFT_{\omega,\Re}(m(x)).$$

2. Compute $< (g(1), ..., (g(\omega^{\Re-1})) > \leftarrow FFT_{\omega,\Re}(g(x))$.
3. For $0 \leq i \leq \Re$, compute $\widetilde{E}_h(z(i)) = z(i) \cdot \widetilde{E}_h(m(i)))$.
4. Let $v_{z(x)} \leftarrow < z(1), \cdots, z(\Re - 1) >$.
5. Compute and return $\widetilde{E}Interporu_{\omega,\Re}(\widetilde{E}_h(v_{z(x)}))$.

Encrypted Polynomial Division Algorithm $(\widetilde{E}PolyDiv_{\widetilde{\omega},n'})(\widetilde{E}_h(m(x)), n(x))$:
Input: Encrypted polynomial $\widetilde{E}(m(x)) =< \widetilde{E}_h\ (m_0)(x), \widetilde{E}_h\ (m_1)(x), \cdots, \widetilde{E}_h(m_{d_1})(x) >$. $g(x) = \Sigma_{i=0}^{\widetilde{m}} g_i x^i$ is the plaintext polynomial. $\widetilde{\omega}$ is a primitive $n' - th$ root of unity, where $n' = 2^k$ and $n' > 2\Re - \widetilde{m} + 1$. h be the public key for $\widetilde{E}$.
Output: Encryption of the remainder polynomial $r(x)$, where $m(x) = g(x)n(x) + r(x)$.
1. Compute $\widetilde{\omega}^2, \cdots, \widetilde{\omega}^{\Re-1}$.
2. Let $m'(x) = rev_{\Re}(m(x))$. We obtain $\widetilde{E}_h(m'(x))$ by reversing the order of coefficient on $\widetilde{E}_h(m(x))$.
3. Compute the polynomial

$$g'(x) = rev_{\Re-\widetilde{m}}(n(x))^{-1} \bmod x^{\Re-\widetilde{m}+1}.$$

4. Compute

$$\widetilde{E}_h(g_1(x)) = \widetilde{E}PolyMult_{\widetilde{\omega},n'}(n'(x)), \widetilde{E}_h(m'(x)).$$

5. Compute

$$\widetilde{E}_h(g_2(x)) = \widetilde{E}_h(g_1(x)) \bmod x^{\Re-\widetilde{m}+1}.$$

This operation can be performed by additively homomorphic evaluation.
6. Compute

$$\widetilde{E}_h(g(x)) = rev_{\Re-\widetilde{m}}(\widetilde{E}_h(g_2(x))).$$

5.5 LHDA on Encrypted Data

After finding a peak value candidate, a detailed fit to the nearby data would be performed for location, eliminating edge response and peak value management. Taylor expansion of the scale-space function ($D(x, y, \sigma)$) is used for these operations. Taylor expansion refers to derivative operation on encrypted data. Because homomorphic encryption do not support derivative operation on encrypted data, we provide an approximate computation based on LHD.

For the original SIFT algorithm [12], when derivative $\frac{\partial}{\partial x} D(x, y, \sigma)$ exists, it can be replaced by $\frac{D(x+h,y,\sigma)-f(x,y,\sigma)}{h}$. Namely, $\frac{\partial}{\partial x} D(x, y, \sigma) \approx \frac{D(x+h,y,\sigma)-D(x,y,\sigma)}{h}$. In the encrypted domain, we use $\frac{\widetilde{E}(D(x+h,y,\sigma))-\widetilde{E}(D(x,y,\sigma))}{\widetilde{E}(h)}$ instead of $\widetilde{E}(\frac{\partial}{\partial x} D(x, y, \sigma))$. Namely, with the help of LHD scheme, $\widetilde{E}(\frac{\partial}{\partial x} D(x, y, \sigma)) \approx \frac{\widetilde{E}(D(x+h,y,\sigma))-\widetilde{E}(D(x,y,\sigma))}{\widetilde{E}(h)}$, where h is very small number. Based on our encoding scheme, h is converted into an integer, and x, y and σ are all expanded by the same multiple. Using the same method, we can perform other approximate derivative computation. Our experiment results of feature point detection show that approximate derivative computation is reasonable.

6 Privacy-preserving SIFT Algorithm

In this section, we present the detailed privacy-preserving SIFT algorithm based on $\widetilde{E}$, LHCA, LHD and LHDA.

6.1 Encryption for Image

Let $I_{original}(x, y)$ be the original image, $I(x, y)$ be the joint image I with split and scrambled sub-images $I_1, I_2, \cdots, I_W$. Scramble can be used to change image data positions [51]. As showed in Fig.1 (b), we make use of partitioned matrix and elementary transformation of matrix to easily preform image split and scramble. Let two-dimensional Gaussian kernel function be $G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2}$. In the encrypted domain, IO pre-computes $G(x, y, \sigma)$ in plaintext domain for (x, y) and different σ, and encodes it with the help of above encoding methods. Finally, IO encrypts encoding result of $G(x, y, \sigma)$ and uploads encrypted data to server S. Gauss scale-space is $\widetilde{E}(L(x, y, \sigma)) =$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2017.2751476, IEEE Transactions on Dependable and Secure Computing

$\widetilde{E}(G(x,y,\sigma)\otimes I(x,y))$. On the basis of convolution operation, $\widetilde{E}(L(x,y,\sigma)) = \widetilde{E}(G(x,y,\sigma)\otimes I(x,y)) = \widetilde{E}(\sum_{i,j} I(x-i,y-j)G(x,y,\sigma)) = \sum_{i,j}\widetilde{E}(I(x-i,y-j)G(x,y,\sigma)) = \sum_{i,j}\widetilde{E}(I(x-i,y-j))\widetilde{E}(G(x,y,\sigma))$. This is the method of generating output image. The output pixel is the weighted sum of the input neighborhood pixels. The computing steps are following: (1) Rotate the kernel $180°$ around center; (2) Slide the kernel, so that its center is on the (i,j) pixel of input image $\widetilde{E}(I(x,y))$; (3) Using above formula, we get the (i,j) pixel of output image; (4) Continuously perform above operations until all the pixels of output image are obtained. When we filter image by convolution, parts of the kernel are located at the outside of edges. Such case is called edge effect. To solve such case, we can take advantage of strategies of filling in constant and edge pixels duplication. DoG has the response image $D(x,y,\sigma)$ by substraction of two images in adjacent gauss scale-space. Namely, $\widetilde{E}(D(x,y,\sigma)) = \widetilde{E}((G(x,y,k\sigma)-(G(x,y,\sigma))\otimes I(x,y)) = \widetilde{E}(L(x,y,k\sigma)) - \widetilde{E}(L(x,y,\sigma))$. The detailed steps of constructing $D(x,y,\sigma)$ are as follows: (1) Gaussian kernel of different scale factors is convolved with image $\widetilde{E}(I(x,y))$ to obtain different scale space of image. This group of images is the first layer in pyramid image; (2) Down-sampling with a twice pixel distance for a twice scale image in the first layer of image to obtain the first image of the second layer in pyramid image. Then, Gaussian kernel with different scale-factor is used to perform convolution with such image. Finally, server S gets a set of images of the second layer in pyramid image; (3) Down-sampling with a twice pixel distance from a twice scale image in the second layer of pyramid image to obtain the first image of the third layer in pyramid image. Then, Gaussian kernel with different scale-factor is used to perform convolution with such image. Finally, server S gets a set of images of the third layer in pyramid image. A set of images in each layer of pyramid image are obtained by the same method; (4) Gaussian difference image is obtained by subtraction of adjacent Gaussian image on each layer.

6.2 Extremum Detection in Scale Space

$\widetilde{E}(D(x,y,\sigma))$ is scale space of image in the encrypted domain. Extremum detection can be performed by server with LHCA on encrypted data.

Extreme points detection of (DoG) scale space can be performed by comparing each sampling point with its adjacent points. Then, sever can judge whether sampling point is larger or smaller than its adjacent points in its image space and scale space. For any one of the detected points, it should be compared with eight adjacent points of same scale. There are total twenty-six $(8+3\times3\times2=26)$ points corresponding to the upper and lower adjacent scales. This method insures that extreme points are detected in both scale space and two-dimensional image space. The searching process of extreme points starts from the second layer of each group. The second layer is taken as current layer. Taking one cube for each point of the second layer in DoG image. Upper and lower layers are the first layer and the third layer. Then, server performs extreme points search by LHCA. When search on the second layer is completed, server uses the third layer as current layer to perform similar search, until all extreme points are found.

In the encrypted domain, server can compare $D(x,y,\sigma)$ with $D(x,y+1,\sigma)$ by LHCA. $\widetilde{E}(D(x,y,\sigma)) = \sum_{i,j}\widetilde{E}(I(x-i,y-j))\widetilde{E}(G(x,y,\sigma))$, $\widetilde{E}(D(x,y+1,\sigma)) = \sum_{i,j}\widetilde{E}(I(x-i,y+1-j))\widetilde{E}(G(x,y,\sigma))$. According to LHCA for encrypted data, server S gets $D(x,y+1,\sigma) > D(x,y,\sigma)$ or $D(x,y+1,\sigma) < D(x,y,\sigma)$. Therefore, server can perform extreme points detection in the encrypted domain.

6.3 Location of the Key Point

Usually, we can obtain extremum in continuous space by interpolation with our LHDA and LHD.

In order to improve stability of the key points, we make use of Taylor function of DoG in scale space:

$\widetilde{E}(D(X)) = \widetilde{E}(D) + \widetilde{E}(\frac{\partial D^T}{\partial X}) + \widetilde{E}(\frac{1}{2}X^T)\widetilde{E}(\frac{\partial^2 D}{\partial X^2})\widetilde{E}(X)$, where $X=(x,y,\sigma)^T$. The offset of extreme point is $\widetilde{E}(\hat{X}) = \widetilde{E}(-\frac{\partial^2 D^{-1}}{\partial X^2})\widetilde{E}(\frac{\partial D}{\partial X})$. For encrypted image, we get

$$\widetilde{E}(\frac{\partial}{\partial x}D(x,y,\sigma)) \approx \widetilde{E}(\frac{D(x+h,y,\sigma)-D(x,y,\sigma)}{h}) = \frac{\widetilde{E}(D(x+h,y,\sigma)-\widetilde{E}(D(x,y,\sigma))}{\widetilde{E}(h)},$$

$$\widetilde{E}(\frac{\partial^2}{\partial x^2}D(x,y,\sigma)) \approx \frac{\widetilde{E}(D(x+2h,y,\sigma))-\widetilde{E}(D(x,y,\sigma))}{\widetilde{E}(h^2)},$$

$$\widetilde{E}(\frac{\partial^2}{\partial x\partial y}D(x,y,\sigma)) \approx \frac{\widetilde{E}(D(x+h,y+h,\sigma))+\widetilde{E}(D(x,y,\sigma))}{\widetilde{E}(h^2)} - \frac{\widetilde{E}(D(x,y+h,\sigma))}{\widetilde{E}(h^2)} - \frac{\widetilde{E}(D(x+h,y,\sigma)}{\widetilde{E}(h^2)}.$$

From

$$\begin{pmatrix} \frac{\partial D}{\partial x} \\ \frac{\partial D}{\partial y} \\ \frac{\partial D}{\partial \sigma} \end{pmatrix} = \frac{\partial D^T}{\partial X}, \begin{pmatrix} \frac{\partial^2 D^{-1}}{\partial x^2} & \frac{\partial^2 D^{-1}}{\partial x\partial y} & \frac{\partial^2 D^{-1}}{\partial x\partial \sigma} \\ \frac{\partial^2 D^{-1}}{\partial y\partial x} & \frac{\partial^2 D^{-1}}{\partial y^2} & \frac{\partial^2 D^{-1}}{\partial y\partial \sigma} \\ \frac{\partial^2 D^{-1}}{\partial \sigma\partial x} & \frac{\partial^2 D^{-1}}{\partial \sigma\partial y} & \frac{\partial^2 D^{-1}}{\partial \sigma^2} \end{pmatrix} = \frac{\partial^2 D^{-1}}{\partial X^2},$$

we can obtain the offset of extreme point $\widetilde{E}(\hat{X}) = \widetilde{E}(-\frac{\partial^2 D^{-1}}{\partial X^2})\widetilde{E}(\frac{\partial D}{\partial X})$. The final location of extreme point is $\widetilde{E}(X+\hat{X})$.

To eliminate unstable points on the edge of image, which has lower response to DoG, we can utilize Hessian matrix. Response peak value of a flat DoG has a large principle curvature across the edge, but has a small principle curvature on the vertical edge. The principal curvature can be found by 2×2 Hessian matrix H:

$$H(x,y) = \begin{pmatrix} D_{xx}(x,y) & D_{xy}(x,y) \\ D_{xy}(x,y) & D_{yy}(x,y) \end{pmatrix}.$$

Let $\alpha=\lambda_{max}$ be the maximum eigenvalue and $\beta=\lambda_{min}$ be the minimal eigenvalue. We can get: $Tr(H) = D_{xx}+D_{yy} = \alpha+\beta$, $Det(H) = D_{xx}D_{yy}-(D_{xy})^2 = \alpha\cdot\beta$, where $Tr(H)$ represents trace of matrix H, and $Det(H)$ represents determinant of matrix H. Let $\gamma=\frac{\alpha}{\beta}$ denote ratio of the largest eigenvalue to the smallest eigenvalue. We have

$\frac{Tr(H)^2}{Det(H)} = \frac{(\alpha+\beta)^2}{\alpha\cdot\beta} = \frac{(\gamma+1)^2}{\gamma}$, where γ can be a pre-set value. According to LHDA, LHD and LHCA, server can compute $\widetilde{E}(H(x,y))$ by $\widetilde{E}(D_{xx}(x,y))$, $\widetilde{E}(D_{xy}(x,y))$, $\widetilde{E}(D_{yy}(x,y))$, $\widetilde{E}(\frac{Tr(H)^2}{Det(H)})$. At the same time, server can compare $\frac{Tr(H)^2}{Det(H)}$ with γ. Hence, server can eliminate unstable key points on the edge.

6.4 Determination of Feature Point Direction

In the encrypted domain, we utilize LHD and LHCA to compute feature point direction. Let $arctan\{\frac{Diff_y}{Diff_x}\}$ be equal to $\wp$. According to $\widetilde{E}$ and LHD, server can compute $\widetilde{E}(\frac{Diff_y}{Diff_x}) \doteq \frac{\widetilde{E}(Diff_y)}{\widetilde{E}(Diff_x)} = \widetilde{E}(tan\wp)$. We divide the rang of $0^\circ \sim 360^\circ$ into 36 parts, where value of each part is 10°. We select $\wp_i$ to be $10^\circ, 20^\circ, 30^\circ, \cdots, 350^\circ$ and 360° as the thresholds. Based on LHCA, server can compare $\widetilde{E}(\frac{Diff_y}{Diff_x})$ with $\widetilde{E}(\wp_i)$ to obtain feature point direction, where $\widetilde{E}(\wp_i)$ can be pre-computed by image owner IO. Gradient argument can be computed by $\sharp(x,y) = \sqrt{(Diff_x)^2 + (Diff_y)^2}$ for SIFT algorithm. In encrypted domain, server can compute it by $\widetilde{E}(\sharp(x,y)) = \|\widetilde{E}(Diff_x)\| + \|\widetilde{E}(Diff_y)\|$.

After finished gradient computation of Gaussian image in the neighbourhood of feature point, gradient direction and argument of pixels in the neighbourhood of feature point can be computed by histogram. The horizontal axis of gradient direction histogram is gradient direction angel, and the vertical axis is gradient magnitude accumulation value corresponding to gradient direction angel (with Gaussian weight). Gradient direction histogram is in the range of $0^\circ \sim 360^\circ$, which is divided into 36 columns. Value of each column is 10°. For histogram in the range of $0^\circ \sim 10^\circ$, result is $\Sigma_{0^\circ\sim 10^\circ}\widetilde{E}(\sharp(x,y)\widetilde{E}(\omega_{x,y})$, where $\widetilde{E}(\omega_{x,y}) = \widetilde{E}(e^{\frac{-(x^2+y^2)}{2(1.5\sigma)^2}})$. In the encrypted domain, IO pre-computes $e^{\frac{-(x^2+y^2)}{2(1.5\sigma)^2}}$ in plaintext domain for (x,y) and different σ. Then, IO encodes it with the help of above encoding methods. Finally, IO encrypts encoded $e^{\frac{-(x^2+y^2)}{2(1.5\sigma)^2}}$ and uploads encrypted data to server. Server can get 8 histograms. Finally, server applies LHCA to compare any two histogram of different angle range, and obtains the peak value of histogram. Peak value of histogram represents the main direction of image gradient in the neighbourhood of feature point. That is the main direction of feature point.

6.5 Key Point Description

Descriptor of feature point makes use of gradient information of eight directions. Gradient information is computed over the 4×4 window in scale space of feature point. Descriptor is represented as $4 \times 4 \times 8$ $(= 128)$ dimensional vector. The steps are as follows:

(1) Rotate axis to the direction of feature point to insure rotation invariance. The gradient location and direction of $3\sigma 5 \times 3\sigma 5 \times 2$ image in the neighbourhood of feature point are rotated with a direction angel θ. Namely, the x-axis of the original encrypted image are rotated to the main direction of feature point.

(2) Determine the image area used to compute descriptor. Feature descriptor is related to scale of feature point. Therefore, gradient should be computed on Gaussian image corresponding to feature point. The neighbourhood of feature point is divided into 4×4 sub-regions. Each sub-region acts as seed point, which has eight direction. Each sub-region has three sub-pixels, and is assigned a rectangular region with a length of three for sampling.

(3) Compute histogram of gradient in sampling area. The rotated region is divided into 4×4 sub-regions. Each region interval contains 3σ pixels. Then, gradient histogram of eight direction is computed in sub-region. By computing the cumulative value in gradient direction, a seed point is formed. At the same time, histogram of gradient direction in each sub-region is divided into eight direction interval from 0° to 360°. Each interval is 45°. That is, each seed point has gradient intensity information of eight direction interval. There are 4×4 sub-regions. Therefore, there are $4 \times 4 \times 8$ (=128) data, which forms 128 dimensional feature vector.

6.6 Encrypted Image Matching by SIFT Algorithm

We use Euclidean distance of feature vectors as similarity measure to perform feature points matching for two images.

Server takes one of feature points in the first image, and find the first two feature points in the second image. Which one of the first two feature points in the second image is closer to the selected feature point in the first image can be performed by Euclidean distance. Euclidean distance of encrypted image can be computed by additively homomorphic evaluation. Comparison of Euclidean distance in the encrypted domain can be performed by LHCA. For the first two feature points in the second image, if Euclidean distance of the first selected feature point with the selected feature point in the first image is less than Euclidean distance of the second selected feature point with the selected feature point in the first image, server computes quotient of these two Euclidean distances. If quotient is less than some proportional threshold ($Ratio$), pair of matching points is accepted. Such quotient in encrypted domain can be computed via LHD. Comparison of proportional threshold and quotient in the encrypted domain can be computed via LHCA. The recommended $Ratio$ is as follows: (1) $Ratio$ is 0.4 for matching required high matching accuracy; (2) $Ratio$ is 0.6 for matching required more matching points; (2) For the general case, $Ratio$ is 0.5.

7 Efficiency and Security Analysis for Our System

In this section, we firstly provide efficiency analysis of $\widetilde{E}$. Secondly, we provide efficiency analysis for LHD, LHCA and LHDA. Finally, we analyze security of $\widetilde{E}$, LHCA, LHD and LHDA based on the RLWE assumption.

7.1 Efficiency Analysis of SIMD LHE Based on Encodings and NTRU

Based on encoding methods and SIMD technology, we can packet many plaintext into a ciphertext by plaintext slot. The plaintext plots can be constructed via algebraic structure.

$\Phi_d(X)$ is the cyclotomic polynomial, where $d = 2^k$ and $n = 2^k - 1$. The plaintext space is defined as $R_{2^p} = R/_{2^p}R$ $(1 < 2^p < q)$. The ciphertext space is defined as $R_q = R/_qR$, modulo q is an integer. $\Phi_d(X)$ can be decomposed

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2017.2751476, IEEE Transactions on Dependable and Secure Computing

into as $\Phi_d(X) = \Pi_i^{\Im} F_i(x)$, where $F_i(x)$ is irreducible. $\Im = \frac{\psi(d)}{\zeta}$ is the number of slot, where ζ is degree of polynomial $F_i(x)$. $Z_{2^p}[X]/(\Phi_d(X)) \cong R_{2^p}[x]/(F_1(x)) \otimes \cdots \otimes R_{2^p}[x]/(F_{\Im}(x)) \cong F_{2^{pd}} \otimes \cdots \otimes F_{2^{pd}}$. $\Im$ independent plaintexts $(M_0, \cdots, M_{\Im-1})$ are batched into the unique element $R_{2^p} = Z_{2^p}[X]/(\Phi_d(X))$. SIMD operation makes use of $\Im$-Add and $\Im$-Mult via following operation: Let I is the index set of plaintexts $(I_0, I_1, \cdots, I_{\Im-1})$, such that $I_i = 1$ if $i \in I$. cI results in the new ciphertext, in which the plaintext element in the slots corresponding to I is contained. If server has two ciphertext c_1 and c_2, server can pack c_1 and c_2 into the $c = c_1\bar{I}_i + c_2\widetilde{I}_j$, where $\bar{I}$ and $\widetilde{I}$ are two index sets. In the same plaintext slots, we can perform additively and multiplicatively SIMD homomorphic evaluation. A function f with $\Im$ different inputs can be computed by homomorphic evaluation with one SIMD operation. We provide comparison of $\widetilde{E}$ with other schemes in Fig. 2.

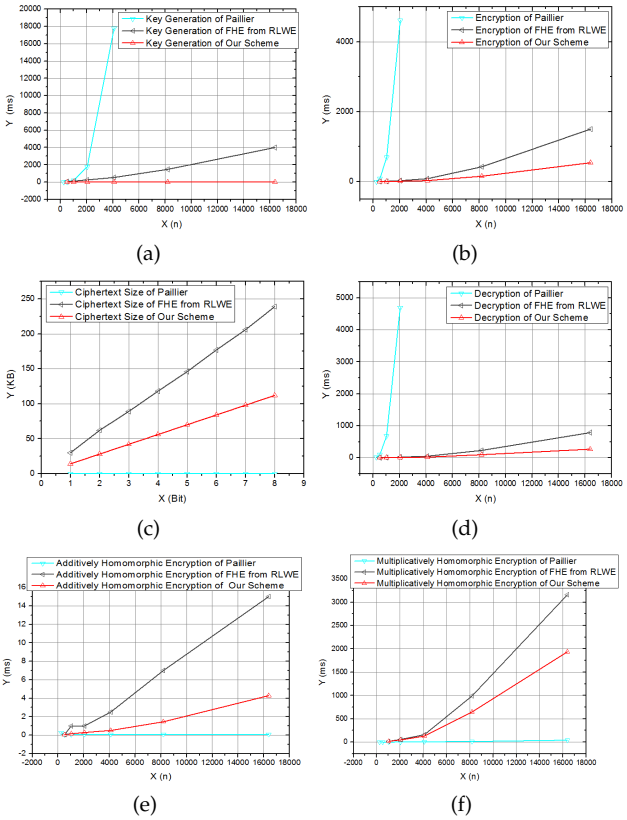


Fig. 2. Comparison of time on key generation, encryption, ciphertext size, decryption and additively and multiplicatively homomorphic evaluation about Paillier, SHE from RLWE and our scheme.

In Fig. 2, X(n) represents module of Paillier [42] and degree of $x^n + 1$, and X(bit) represents the plaintext size. Y(ms) represents the computing times, and Y(KB) represents ciphertext size. Paillier encryption of Hsu et al. [14] has a ciphertext expansion of two. Namely, a plaintext of $|N|$ bits is expanded to a ciphertext of $|N^2|$ bits. SHE based on RLWE used in Hu et al. [25] has two ring elements in ciphertext. $\widetilde{E}$ has one ring element in ciphertext. We compare ciphertext size of $\widetilde{E}$ with SHE from RLWE with same parameter. According to Fig. 2, $\widetilde{E}$ has smaller ciphertext, faster key generation, faster encryption (decryption) and faster homomorphic evaluation than SHE based on RLWE used in Hu et al. [25].

7.2 Efficiency Analysis of LHD

LHD algorithm on encrypted data mainly includes powering and addition operations. Powering operation contains FFT algorithm, $\widetilde{E}.Interpru$, encrypted polynomial multiplication algorithm and encrypted polynomial division algorithm. FFT Algorithm requires $\Re \log \Re$ times of homomorphic addition/subtraction and $\frac{\Re}{2} \log \Re$ times of homomorphic multiplication. $\widetilde{E}.Interpru$ has the same result. Encrypted polynomial multiplication algorithm invokes FFT algorithm and $\widetilde{E}.Interpru$ at a time. Encrypted polynomial multiplication algorithm is invoked two times by encrypted polynomial division algorithm. Step 5 and 7 of encrypted polynomial division algorithm require $\widetilde{m}$ and $(\Re - \widetilde{m})$ times of homomorphic addition/substraction. Hence, we can easily obtain following result.

Theorem 1 (Correctness of LHD). *Let* $q, t = 2^p (1 < t < q)$, $\Delta = \lfloor q/2 \rfloor$, $\delta = sup\{\|g \cdot h\|_\infty / (\|g\|_\infty \|h\|_\infty) : g, h \in R\}$, B_{key} *is* B*-bounded key sampling distribution,*$\eta_2(q)$ *means the reduction of* q *into* $[0, 2)$. $\boldsymbol{v}_1$ *is the inherent noise of* $\widetilde{E}(5^j)$. $\boldsymbol{v}_2$ *is the inherent noise of* $\widetilde{E}(y)$. $\boldsymbol{v}_3$ *is the inherent noise of* $\widetilde{E}(10^{j+1})$. $\boldsymbol{V}(> 0)$ *is the inherent noise bound, namely* $\|\boldsymbol{v}_i\|_\infty \leq \boldsymbol{V} < \Delta/2 (i \in 1, 2, 3)$. $\|\boldsymbol{V}_u\| < \boldsymbol{V} + \frac{1}{2}(\delta 2(3 + 2\delta B_{key})(2\boldsymbol{V}) + \delta \boldsymbol{V} + 2\delta \eta_2(q)(5 + 2\delta + B_{key}) + (2\delta B_{key})^2 + 4\delta^2 \ell_{2,q} B_{error} B_{key})$. *The scheme LHD can correctly compute* $\boldsymbol{n}$ *times multiplications that are arranged in a binary tree of* $\boldsymbol{n}$ *levels of multiplication, if* $[((4\boldsymbol{n} \log \boldsymbol{n} + \frac{\boldsymbol{n}}{2} \log \boldsymbol{n} \delta (3 + 2\delta B_{key}) + \delta + \boldsymbol{n}))^{\boldsymbol{n}} \|\boldsymbol{V}_u\|_\infty + \boldsymbol{n}((4\boldsymbol{n} \log \boldsymbol{n} + \frac{\boldsymbol{n}}{2} \log \boldsymbol{n} \delta (3 + 2\delta B_{key}) + \delta + \boldsymbol{n}))^{\boldsymbol{n}-1} (\boldsymbol{n} + 1) \eta_2(q) + \boldsymbol{n} \log \boldsymbol{n} \delta \eta_2(q) [((5 + 2\delta + B_{key}) + (\delta 2 B_{key})^2) + 1) + \delta^2 \ell_{2,q} 2 B_{error} B_{key}]] < \triangle/2$.

7.3 Efficiency Analysis of LHDA

LHDA includes additively homomorphic evaluation and LHD.

Homomorphic first-order partial derivative $\frac{\partial}{\partial x} D(x, y, \sigma)$ needs one time of additively homomorphic evaluation and one time of LHD on encrypted data. Second-order partial derivative $\frac{\partial^2}{\partial x \partial y} D(x, y, \sigma)$ needs three times of additively homomorphic evaluation and three times of LHD on encrypted data.

7.4 Efficiency Analysis of LHCA

According to HCA on encrypted data, IO encrypts data and uploads encrypted data to server. $\widetilde{E}$ has smaller size of ciphertext and faster homomorphic evaluation on encrypted data. IO computes $c_1 = \widetilde{E}(M_1) = [\lfloor q/t \rfloor [M_1]_t + e_1 + hs_1]_q (\in R_q)$, $c_2 = \widetilde{E}(M_2) = [\lfloor q/t \rfloor [M_2]_t + e_2 + hs_2]_q (\in R_q)$ and $c_3 = \widetilde{E}(M_1) = [\lfloor q/t \rfloor [M_1]_t + e_2 + hs_2]_q (\in R_q)$ or $\widetilde{c}_3 = \widetilde{E}(M_2) = [\lfloor q/t \rfloor [M_2]_t + e_1 + hs_1]_q (\in R_q)$, where t is 2^p. Such three ciphertexts is still small with fast computing speed. With just one time additively homomorphic evaluation, server can judge $M_2 > M_1$ or $M_2 < M_1$ by $c_2 - c_3 > 0$ or $\widetilde{c}_3 - c_1 > 0$, or $\widetilde{c}_3 - c_1 < 0$.

7.5 Security of LHE, LHD, LHDA and LHCA

The key dependent message security is IND-CPA security. The security of $\widetilde{E}$, LHD and LHDA is based on IND-CPA [45] and RLWE assumption [43].

$RLWE_{d,q,\chi_{error}}$ is hard problem. Therefore, $\widetilde{E}$ is IND-CPA security. Our new LHD and LHDA are IND-CPA security under RLWE assumption. For the subfield attack on the scheme of YASH [36], the subfield attack is subexponential in the security parameter λ for YASH [33], if a) L (depth of a circuit evaluated) is sufficiently big to enable Fully homomorphic encryption, and b) n is chosen to be minimal such that a lattice attack on the full field does not succeed. The condition a) requires the scheme of LHE can execute the bootstrapping procedure, which has to make an additional assumption. But our LHE scheme does not support such additional assumption based on selected parameters. Therefore, our LHE scheme are immune to such attack.

For the security of LHCA, we mainly consider attacks on the known plaintext attack (KPA) and the ciphertext only attack (COA). Server S is honest-but-curious. S may be able to derive the parameters s_1 and s_2 or e_1 and e_2, and obtain M_1 and M_2.

Definition 1 ($\alpha - BDD$ Problem [16]). *Let a lattice ς and a vector ϑ (within distance $\alpha \cdot \lambda_1(\varsigma)$), $\alpha - BDD$ problem is to find a lattice point $\varrho \in \varsigma$ within distance $\alpha \cdot \lambda_1(\varsigma)$ from the target.*

Theorem 2 (Security of LHCA)**.** *The algorithm LHCA is COA and KPA security.*

Proof. For FHCA, we can prove it is COA and KPA security. Server has three ciphertexts c_1, c_2 and c_3 or $\widetilde{c}_3$. $c_2 - c_3 = [\lfloor q/t \rfloor [M_2]_t + e_2 + hs_2]_q - [\lfloor q/t \rfloor [M_1]_t + e_2 + hs_2]_q = [\lfloor q/t \rfloor [M_2 - M_1]_t]_q$. But, according to the $\alpha - BDD$ problem of [16], server can not obtain M_1 and M_2. Similarly, server can not obtain M_1 and M_2 from $\widetilde{c}_3 - c_1$. $c_1 - c_3 = [\lfloor q/t \rfloor [M_1]_t + e_1 + hs_1]_q - [\lfloor q/t \rfloor [M_1]_t + e_2 + hs_2]_q = e_1 - e_2 + h(s_1 - s_2)$. According to the RLWE assumption, server can not obtain s_1 and s_2 or e_1 and e_2.

7.6 Efficiency and Security Analysis of SIFT Algorithm in the Encrypted Domain

In this section, we provide an analysis for efficiency and security of SIFT algorithm in the encrypted domain.

7.6.1 Extremum Detection In Scale Space

About extremum detection in the scale space, server compares each sampling point with its adjacent points. According to LHCA, server can independently perform comparison computation on encrypted data. Such result can be obtained by the following computing formula: $\widetilde{E}(D(x,y,\sigma)) = \sum_{i,j} \widetilde{E}(I(x-i,y-j))\widetilde{E}(G(x,y,\sigma))$, $\widetilde{E}(D(x,y+1,\sigma)) = \sum_{i,j} \widetilde{E}(I(x-i,y+1-j))\widetilde{E}(G(x,y,\sigma))$. Server compares $\widetilde{E}(D(x,y+1,\sigma))$ and $\widetilde{E}(D(x,y,\sigma))$ by LHCA. Finally, server gets $D(x,y+1,\sigma) < D(x,y,\sigma)$ or $D(x,y+1,\sigma) > D(x,y,\sigma)$.

7.6.2 Locating Key Point

Above obtained extreme points are extreme points in discrete space. Usually, server can obtain extremum in the continuous space by interpolation with our LHDA and LHD.

Server can take advantage of LHDA to obtain the offset of extreme point by $\widetilde{E}(\hat{X}) = \widetilde{E}(-\frac{\partial^2 D^{-1}}{\partial X^2})\widetilde{E}(\frac{\partial D}{\partial X})$. The final location of extreme point is $\widetilde{E}(X + \hat{X})$.

Based on LHDA, server can eliminate the points at the edge of image. From $\widetilde{E}(\frac{\partial^2}{\partial x^2}D(x,y,\sigma))$ and $\widetilde{E}(\frac{\partial^2}{\partial x \partial y}D(x,y,\sigma))$, server can compute: $\widetilde{E}(Tr(H)) = \widetilde{E}(D_{xx}) + \widetilde{E}(D_{yy}) = \alpha + \beta$, $\widetilde{E}(Det(H)) = \widetilde{E}(D_{xx})\widetilde{E}(D_{yy}) - \widetilde{E}((D^2_{yy})) = \alpha \cdot \beta$, where $\widetilde{E}(Tr(H))$ represents trace of matrix H, $\widetilde{E}(Det(H))$ represents determinant of H. Let $\gamma = \frac{\alpha}{\beta}$ represents ratio of the largest eigenvalue to the smallest eigenvalue, server has $\frac{\widetilde{E}(Tr(H)^2)}{\widetilde{E}(Det(H))} = \frac{(\alpha+\beta)^2}{\alpha \cdot \beta} = \frac{(\gamma+1)^2}{\gamma}$. Then, making use of LHCA, server can get $\frac{Tr(H)^2}{Det(H)} < \gamma_1$, where γ_1 is the pre-set threshold.

7.6.3 Determining Feature Point Direction

Server can compute gradient argument and gradient direction by LHD and LHCA. 360° is divided into 36 parts, $\wp_i$ is the corresponding angle. The ciphertext of $\wp_i$, which is threshold for angle, can be pre-computed. Server compares $\widetilde{E}(tan(\wp_i))$ with $\frac{\widetilde{E}(Diff_y)}{\widetilde{E}(Diff_x)}$ by LHD and LHCA. Therefore, gradient direction can be confirmed.

Gradient argument can be computed by $\widetilde{E}(\sharp(x,y)) = \|\widetilde{E}(Diff_x)\| + \|\widetilde{E}(Diff_y)\|$. Server can obtain histogram by $\Sigma_{0^\circ \sim 10^\circ}\widetilde{E}(\sharp(x,y))\widetilde{E}(\omega_{x,y})$, $\Sigma_{10^\circ \sim 20^\circ}\widetilde{E}(\sharp(x,y))\widetilde{E}(\omega_{x,y})$, $\cdots$, and $\Sigma_{350^\circ \sim 360^\circ}\widetilde{E}(\sharp(x,y))\widetilde{E}(\omega_{x,y})$. Finally, server obtains the main direction of feature point.

7.6.4 Feature Point Description

Server has obtained location, scale and direction of feature point in the encrypted domain. Descriptor of feature point is a set of vector. It not only includes feature points' information, but also contains pixels' information around feature points. These pixels also contributes to feature point.

In order to guarantee rotation invariance of feature vector, server rotates the coordinate axis (main direction of feature point) in the neighborhood of feature point. That is, rotate coordinate axis into main direction of feature point. After rotation, new coordinate of pixels in the neighbourhood can be computed in the following method. According to $\widetilde{E}(tan\theta) = \frac{\widetilde{E}(L(x,y+1)) - \widetilde{E}(L(x,y-1))}{\widetilde{E}(L(x+1,y)) - \widetilde{E}(L(x-1,y))}$, server can get $\widetilde{E}(sec^2\theta) = \widetilde{E}(1+tan^2\theta)$ and $\widetilde{E}(csc^2\theta) = \widetilde{E}(1+cot^2\theta)$. That is, server can get $\widetilde{E}(cos\theta)$ and $\widetilde{E}(sin\theta)$. Hence, server get the new rotated coordinates of pixels within neighborhood by

$$\begin{pmatrix} \widetilde{E}(x') \\ \widetilde{E}(y') \end{pmatrix} = \begin{pmatrix} \widetilde{E}(cos\theta) & \widetilde{E}(-sin\theta) \\ \widetilde{E}(sin\theta) & \widetilde{E}(cos\theta) \end{pmatrix} \begin{pmatrix} \widetilde{E}(x) \\ \widetilde{E}(y) \end{pmatrix}.$$

By LHD, server obtains rotated encrypted image.

After performing rotation on the location and image gradient direction in the neighborhood of feature point, server takes a $3\sigma 4 \times 3\sigma 4$ image region in the rotated image. Such image region is equally divided into 4×4 sub-regions. After division, each interval of sub-regions has 3σ pixels. Then, gradient direction histogram can be computed in each sub-region. But, different from above computing on feature point gradient direction, gradient histogram of each sub-region is divided into eight directions in the range of $0^\circ \sim 360^\circ$. Each range is 45°. The cumulative value

of each gradient direction can be computed as follows: $\Sigma_{0^\circ\sim45^\circ}\widetilde{E}(\sharp(x,y))\widetilde{E}(\omega_{x,y})$, $\Sigma_{45^\circ\sim90^\circ}\widetilde{E}(\sharp(x,y))\widetilde{E}(\omega_{x,y})$, $\cdots$, and $\Sigma_{270^\circ\sim360^\circ}\widetilde{E}(\sharp(x,y))\widetilde{E}(\omega_{x,y})$. Therefore, server gets gradient intensity information of eight direction for each seed. There are 4×4 sub-regions. Server obtains 128 dimensional feature vector in the encrypted domain.

7.6.5 Security Analysis of SIFT Algorithm in the Encrypted Domain

We mainly consider KPA and COA attacks. Server S is honest-but-curious. For our new secure SIFT algorithm in the encrypted domain, server can not get encryption parameters e, s and image content (including pixel value, key point location and extracted feature).

Theorem 3 (Security of privacy-preserving SIFT Algorithm)**.** *Security of our new privacy-preserving SIFT algorithm is COA and KPA security.*

Proof. We can prove our new scheme of SIFT in the encrypted domain is COA and KPA security. $\widetilde{E}$ is based on $RLWE_{d,q,\chi_{error}}$ assumption, which is hard problem. $\widetilde{E}$ is IND-CPA security. During the process of extremum detection in the scale space, server only knows the relationship between $D(x, y+1, \sigma)$ and $D(x, y, \sigma)$ by LHCA. Based on theorem [2], such comparison algorithm in the encrypted domain is KPA and COA security. For key point location, server makes use of LHD and LHDA, which are IND-CPA security. Because IND-CPA security has stronger security than COA and KPA security, server can not get encryption parameters e, s and $(x, y, I_{original}(x, y))$. During the processing of eliminating points at the edge of image and determining feature point direction and feature point description, server can not obtain encryption parameters e, s and $(x, y, I_{original}(x, y))$. Therefore, server can not get encryption parameters e, s and $(x, y, I_{original}(x, y))$.

For the security of content on image, we show that S can not get the content on encrypted image. Our SIFT algorithm in encrypted domain may reveal the local characteristics of blocks (sub-images) in the encrypted domain. For example, there is an image of size $\aleph \times \aleph$ with each pixel of 8-bit length. According to encryption algorithm, image I is split $\emptyset \times \emptyset$ blocks and scrambled, S do not know the exact location of blocks. Namely, for $\emptyset \times \emptyset$ blocks, which are scrambled, there are $(\emptyset^2)!$ different possibilities to recover the original blocks (sub-images). When $\emptyset = 10$, server S recovers the the original blocks with probability $P = \frac{1}{(10^2)!} = \frac{1}{9.332621544394415\times10^{157}}$. Therefore, this is an exponential computational complexity problem for recovering the original image I. Such problem is impractical for S to perform an exhaustive search. S is unable to get the content of original image I by what it has obtained.

Because LHCA in the whole privacy-preserving SIFT algorithm is COA and KPA security, our new privacy-preserving SIFT algorithm is COA and KPA security.

8 Experiment Evaluation

In this section, we perform various experiments to evaluate feature detection, edge effect elimination and image matching of our secure SIFT algorithm in the encrypted domain. We perform experiments on a desktop computer with an Intel core(TM)i7-4710MQ running at 2.9 GHZ and 8G of memory.

8.1 Experiment Evaluation of Feature Detection and Image Matching

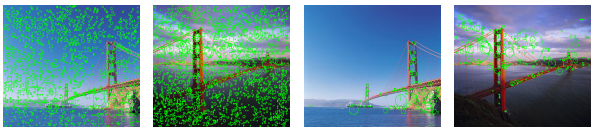
Our experiments evaluate and compare feature detection, edge effect elimination and image matching on original SIFT algorithm [12], the schemes of Hsu et al. [14], Hu et al. [25] and our new scheme. We conduct experiments on real image dataset: Caltech-256 Object Category [49], which contains a challenging set of 256 object categories with a total of 30607 images. The original images are Fig. 3(a)(b), which are 376×300 resolution. Fig. 3(c)(d) are the grey images of original images.

8.1.1 Interpolation of Unstable Key Point and Eliminating Edge Effect of SIFT Algorithm

Fig. 4(a)(b), Fig. 5(a)(b) and Fig. 6(a)(b) are the results of key point detection on the schemes of Hsu et al. [14], Hu et al. [25] and our scheme, where operations on interpolation of unstable key points and eliminating edge effect are not performed.

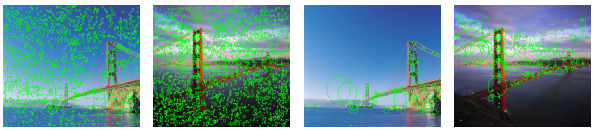
(a) (b) (c) (d)

Fig. 3. (a) and (b) Two original images of 376 X 300 resolution, (c) and (d) Grey images of two original images.



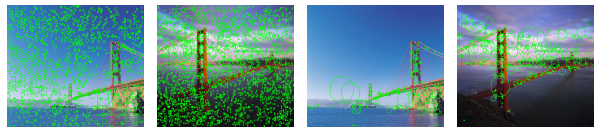
(a) (b) (c) (d)

Fig. 4. (a) and (b) Results of the scheme of Hsu et al. [14] after recovery in plaintext domain. (c) and (d) Results of original SIFT Algorithm [12].



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

Fig. 5. Results of the scheme of Hu et al. [25] after recovery in plaintext domain.



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

(e) (f)

Fig. 6. Results of our new secure SIFT algorithm after recovery in plaintext domain.

For the scheme of Hsu et al. [14], it can not perform operations on interpolation of unstable key points and eliminating edge effect in the encrypted domain. Fig. 4(a)(b) are

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TDSC.2017.2751476, IEEE Transactions on Dependable and Secure Computing

its results of feature point detection. The scheme of Hsu et al. [14] has the largest number of feature points.

For the scheme of Hu et al. [25], it can not perform division and derivative computing in the encrypted domain. Fig. 5(c)(d) show the result of eliminating unstable key points based on their key points localization protocol. The result of feature point detection is very close to the original SIFT algorithm [12].

Because our new scheme supports division and derivative operation in the encrypted domain, hence, we can get Fig. 6(c)(d) for the interpolation of unstable key points and Fig. 6(e)(f) for eliminating edge effect.

From these results, we can conclude that the result of our new scheme is the closest to Fig. 4(c)(d) of original SIFT algorithm [12] in the plaintext domain. The scheme of Hu et al. [25] is closer to original SIFT algorithm [12] in the plaintext domain.

Fig. 7. (a) Result of image matching on the scheme of Hsu et al. [14], (b) Result of image matching on the scheme of Hu et al. [25], (c) Result of image matching based on our scheme, and (d) Result of image matching based on original SIFT algorithm [12].

8.1.2 Image Matching Based on SIFT Algorithm in the Encrypted Domain

Fig. 7(a) shows that the scheme of Hsu et al. [14] has seven matching feature points. Fig. 7(b) shows that the scheme of Hu et al. [25] has nine matching feature points. Such result is closer to the original SIFT algorithm [12] in the plaintext domain. Fig. 7(c) shows that our new scheme has fifteen matching feature points. Such result is the closest to matching result of the original SIFT algorithm [12], which has sixteen matching feature points in Fig. 7(d).

Our new scheme and the scheme of Hu et al. [25] can improve efficiency of image matching. But, our new scheme can eliminate unstable key points and edge effect. Therefore, our new scheme is more efficient than the scheme of Hu et al. [25]. The scheme of Hsu et al. [14] only performs key points extraction, but can not perform elimination of unstable key points and edge effect. Therefore, the scheme of Hsu et al. [14] has lowest efficiency of image matching.

8.2 Effectiveness Evaluation of Our Scheme

In this section, we will evaluate effectiveness of feature point descriptor by metrics. Namely, we choose feature point matching experiment to evaluate validity and robustness of feature points detection.

Based on encrypted image matching with SIFT algorithm, if quotient is less than some proportional threshold ($Ratio$), pair of matching points is accepted. We perform

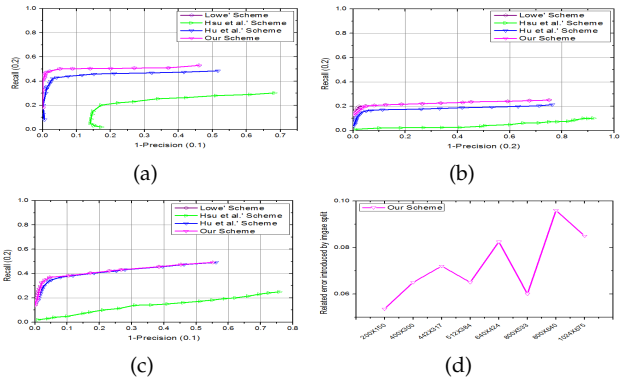


Fig. 8. *Recall* versus *1-precision*. (a) Result of match with image rotated $30°$ and scaled by 10%. (b) Result of match with image blurred. (c) Result of match with image on illumination. (d) The average relative error introduced by image split.

comparative analysis of secure SIFT algorithm with the schemes of Hsu et al. [14], Hu et al. [25] and our new scheme using image dataset [50]. This image dataset is provided by Visual Geometry Group of University of Oxford. Images of dataset [50] evaluates six kinds of image transformations. We make use of general evaluation metrics *recall* and 1-*precision* defined in [25] to perform image matching evaluation: $recall = \frac{number\ of\ correct-matches}{total\ number\ of\ positives}$ and 1-$precision = \frac{number\ of\ false-matches}{total\ number\ of\ matches}$. A *correct-match* is matching, where the two feature points correspond to the same physical location. A *false-match* is matching, where the two feature points come from different physical locations. *The total number of positives* is known as *a priori* for the given dataset. In order to show image split with very low impact on feature point detection, we define the average relative error as:

$$AErr_{v'} = \frac{\widetilde{v} - v'}{\widetilde{v}},$$

where $\widetilde{v}$ is number of feature point detected by original SIFT algorithm [12], v' is number of feature point detected by our new scheme.

In Fig. 8, we show *Recall* VS. 1-*precision* for image matching with different *Ratio*. Fig. 8(a) shows the result of image rotated $30°$ and scaled by 10%. When *Ratio* is very small, our new scheme and Lowe's scheme [12] has a higher *Recall* and lower 1-*precision* than the schemes of Hsu et al. [14] and Hu et al. [25]. *Recall* of our new scheme and Lowe's scheme [12] are all very close to 0.6, and 1-*precision* of our new scheme and Lowe's scheme [12] are all smaller than 0.1. When the value of *Ratio* is close to 1, 1-*precision* value of our new scheme and Lowe's scheme [12] is smaller than 0.5, but that of Hu et al. [25] is larger than 0.5, and that of Hsu et al. [14] is very close to 0.7. Such result indicates that the schemes of Hsu et al. [14] and Hu et al. [25] induce more false matching feature points than our new scheme and Lowe's scheme [12].

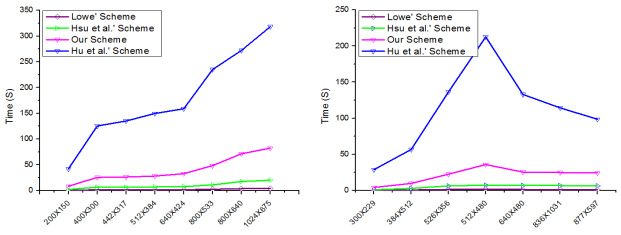
Fig.8 (b) presents result of feature points matching with image blurred. Namely, the same scene of image has different camera focus. Our new scheme and Lowe's scheme [12] have better matching result than the schemes of Hsu et al. [14] and Hu et al. [25], because our new scheme and Lowe's scheme [12] can effectively eliminate unstable key points.

Fig. 8(c) shows result of feature points matching with image on illumination. Our new scheme and Lowe's scheme [12] have better matching result than the schemes of Hsu et al. [14] and Hu et al. [25], because our new scheme and Lowe's scheme [12] can normalize the generated feature vectors to find some very prominent feature points. These feature points are not affected by illumination.

Fig. 8(d) presents the result of average relative error introduced by image split. Operation of image split has very low impact on image matching. Such effect can be ignored.

8.3 Efficiency Evaluation of Secure SIFT Algorithm

In this section, we compare efficiency of feature point detection and image matching on original SIFT algorithm [12] with the schemes of Hsu et al. [14], Hu et al. [25] and our new scheme.



(a) (b)

Fig. 9. (a) Time of feature points detection. (b) Time of image match.

Fig. 9(a) presents time comparison of feature point detection. Our new scheme is much more efficient than scheme of Hu et al. [25]. The reason of such result is that our encryption scheme based our new encoding schemes is much more efficient than Fully (somewhat) homomorphic encryption from RLWE, and that our new LHCA algorithm is non-interactive. Fig. 9(b) shows time comparison of image matching. Our new scheme is much more efficient than scheme of Hu et al. [25].

9 Conclusion

The privacy-preserving feature extraction on secure SIFT algorithm can obtain feature descriptor on encrypted image without exposing privacy of image. Homomorphic encryption provides possibility of privacy-preserving feature extraction in the encrypted domain. But, previous schemes had not provide a perfect efficient implementation on secure privacy-preserving SIFT algorithm. For improving efficiency of secure privacy-preserving SIFT algorithm in the encrypted domain, we present a new secure privacy-preserving SIFT algorithm based on our new encoding methods, LHCA, LHD and LHDA. According to our scheme, we can get the scheme of privacy-preserving feature extraction on secure SIFT algorithm with smaller storage and higher efficiency. As the original SIFT algorithm, robust feature point detection, accurate feature point description and matching can be obtained by $\widetilde{E}$, LHCA, LHD and LHDA.

Acknowledgment

The work described in this paper was supported by the National Natural Science Foundation of China (Grant No. 61370203), Lu'an Commission Directed City-Level Key Research (Grant No. 2010LWA004) and Lu'an Commission Directed City-Level Research (Grant No. 2012LW022).

References

[1] R. L. Rivest, L. Adleman, and M. L. Dertouzos, On data banks and privacy homomorphisms, Foundations of Secure Computation, 1978, pp. 169-179.
[2] C. Gentry, Computing arbitrary functions of encrypted data, Communications of the Acm, 53(3), 2010, pp. 97-105.
[3] C. Gentry, Fully homomorphic encryption using ideal lattices, Proc Stoc, 2009(4), 2011, pp. 169-178.
[4] Song, Dawn Xiaodong, D. Wagner, and A. Perrig, "Practical Techniques for Searches on Encrypted Data," IEEE Symposium on Security and Privacy, 2000, pp. 44-55.
[5] Swaminathan, Ashwin, et al., "Confidentiality-preserving rank-ordered search," ACM Workshop on Storage Security and Survivability, Storagess 2007, Alexandria, Va, Usa, October, 2007, pp. 7-12.
[6] Zhang, Wei, et al., "Privacy Preserving Ranked Multi-Keyword Search for Multiple Data Owners in Cloud Computing," IEEE Transactions on Computers, 65(5), 2016, pp. 1566-1577.
[7] J. Shashank, et al., "Private Content Based Image Retrieval," IEEE Conference on Computer Vision and Pattern Recognition, 2008, pp. 1-8.
[8] Lu, Wenjun, "Enabling search over encrypted multimedia databases," Proceedings of SPIE - The International Society for Optical Engineering, 7254, 2009, pp. 725418-725418-11.
[9] Wang, Qian, et al., CloudBI: Practical Privacy-Preserving Outsourcing of Biometric Identification in the Cloud, Computer Security C ESORICS 2015, Springer International Publishing, 2015.
[10] Zhang, Lan, et al., "PIC: Enable Large-Scale Privacy Preserving Content-Based Image Search on Cloud," 2015 44-th International Conference on Parallel Processing (ICPP) IEEE Computer Society, 2015, pp. 949-958.
[11] Ren, Kui, C. Wang, and Q. Wang, "Security Challenges for the Public Cloud," IEEE Internet Computing, 16(1), 2012, pp. 69-73.
[12] Lowe, David G, "Distinctive Image Features from Scale-Invariant Keypoints," International Journal of Computer Vision, 60(2), 2004, pp. 91-110.
[13] Lu, Chun, Shien, "Homomorphic encryption-based secure SIFT for privacy-preserving feature extraction," Proceedings of SPIE - The International Society for Optical Engineering, 7880(2), 2011, pp. 788005-788005-17.
[14] Hsu, C. Y., C. S. Lu, and S. C. Pei, "Image feature extraction in encrypted domain with privacy-preserving SIFT," IEEE Transactions on Image Processing A Publication of the IEEE Signal Processing Society, 21(11), 2012, pp. 4593-4607.
[15] Amir, S, M. Solomon, and Z. Amit, "Notes on non-interactive secure comparison in "image feature extraction in the encrypted domain with privacy-preserving SIFT"," 2014, pp. 135-140.
[16] Linzhi, Jiang, et al. "Statistical learning based fully homomorphic encryption on encrypted data." Soft Computing, 2016, pp. 1-11.
[17] Lu, Wenjun, et al., "Secure image retrieval through feature protection," IEEE International Conference on Acoustics, 2009, pp. 1533-1536.
[18] Naehrig, Michael, K. Lauter, and V. Vaikuntanathan, "Can homomorphic encryption be practical?," ACM Cloud Computing Security Workshop, Ccsw 2011, Chicago, Il, Usa, October, 2011, pp. 113-124.
[19] Qin, Zhan, et al., "Towards Efficient Privacy-preserving Image Feature Extraction in Cloud Computing," the ACM International Conference, 2014, pp. 497-506.
[20] Wang, Shumiao, et al., Secure and Private Outsourcing of Shape-Based Feature Extraction, Information and Communications Security, Springer International Publishing, 2013, pp. 90-99.
[21] Huang, Yan, et al., "Faster secure two-party computation using garbled circuits," Usenix Conference on Security, 2011, pp. 35-35.
[22] Boldyreva, Alexandra, et al., "Order-Preserving Symmetric Encryption," Advances in Cryptology - EUROCRYPT 2009, International Conference on the Theory and Applications of Cryptographic Techniques, Cologne, Germany, April, 26-30, 2009, Proceedings, 2009, pp. 224-241.

[23] Wang, Qian, et al., "Catch me in the dark: Effective privacy-preserving outsourcing of feature extractions over image data," IEEE INFOCOM 2016 - IEEE Conference on Computer Communications, 2016.
[24] Wang, Qian, et al., "SecHOG: Privacy-Preserving Outsourcing Computation of Histogram of Oriented Gradients in the Cloud," The ACM 2016.
[25] Hu, Shengshan, et al., "Securing SIFT: Privacy-preserving Outsourcing Computation of Feature Extractions Over Encrypted Image Data," IEEE Transactions on Image Processing, 2016, pp. 1-1.
[26] Brakerski, Zvika, and V. Vaikuntanathan, "Fully Homomorphic Encryption from Ring-LWE and Security for Key Dependent Messages," Advances in Cryptology - CRYPTO 2011 - Cryptology Conference, Santa Barbara, Ca, Usa, August, 14-18, 2011, Proceedings, 2011, pp. 505-524.
[27] Ximeng, Liu et al. "Efficient and Privacy-Preserving Outsourced Calculation of Rational Numbers." IEEE Transactions on Dependable and Secure Computing (2016):1-1.
[28] Y. Elias, K. E. Lauter, and E. Ozman, et al., Provably weak instances of Ring-LWE, Advances in Cryptology-CRYPTO 2015, Springer Berlin Heidelberg, 2015, pp. 63-92.
[29] Hoffstein, Jeffrey, J. Pipher, and J. H. Silverman, "NTRU: A ring-based public key cryptosystem," International Symposium on Algorithmic Number Theory Springer-Verlag, 1998, pp. 267-288.
[30] Costache, Ana, and N. P. Smart, Which Ring Based Somewhat Homomorphic Encryption Scheme is Best?, Topics in Cryptology - CT-RSA 2016, Springer International Publishing, 2016.
[31] Lpezalt, Adriana, E. Tromer, and V. Vaikuntanathan, "On-the-fly multiparty computation on the cloud via multikey fully homomorphic encryption," Proceedings of the Annual Acm Symposium on Theory of Computing, 2012, pp. 1219-1234.
[32] Rohloff, Kurt, and D. B. Cousins, A Scalable Implementation of Fully Homomorphic Encryption Built on NTRU, Financial Cryptography and Data Security, Springer Berlin Heidelberg, 2014, pp. 221-234.
[33] Bos, W. Joppe, et al., Improved Security for a Ring-Based Fully Homomorphic Encryption Scheme, Cryptography and Coding, Springer Berlin Heidelberg, 2013, pp. 45-64.
[34] Stehlé, Damien, and R. Steinfeld, "Making NTRU as Secure as Worst-Case Problems over Ideal Lattices," Advances in Cryptology - EUROCRYPT 2011 - International Conference on the Theory and Applications of Cryptographic Techniques, Tallinn, Estonia, May, 15-19, 2011, Proceedings, 2011, pp. 27-47.
[35] Clear, Michael, and C. Mcgoldrick, Multi-identity and Multi-key Leveled FHE from Learning with Errors, Advances in Cryptology - CRYPTO 2015, 2015, pp. 630-656.
[36] Albrecht, Martin, B. Shi, and L. Ducas, A Subfield Lattice Attack on Overstretched NTRU Assumptions, Advances in Cryptology-CRYPTO 2016, 2016.
[37] Lepoint, Tancrède, and M. Naehrig, A Comparison of the Homomorphic Encryption Schemes FV, and YASHE, Progress in Cryptology-AFRICACRYPT 2014, Springer International Publishing, 2014, pp. 318-335.
[38] Junfeng Fan and Frederik Vercauteren, Somewhat practical fully homomorphic encryption, IACR Cryptology ePrint Archive, 2012.
[39] David G. Lowe, "Object recognition from local scale-invariant features," International Conference on Computer Vision, Corfu, Greece, September, 1999, pp. 1150-1157.
[40] David G. Lowe, "Local feature view clustering for 3D object recognition," IEEE Conference on Computer Vision and Pattern Recognition, Kauai, Hawaii, December, 2001, pp. 682-688.
[41] K. Mikolajczyk and C. Schmid, A performance evaluation of local descriptors, IEEE Trans. Pattern Anal. Mach. Intell., vol. 27, no. 10, pp. 1615-1630, Oct. 2005.
[42] P. Paillier, Public-key cryptosystems based on composite degree residuosity classes, Advances in Cryptology-Eurocrypt, 547(1), 1999, pp. 223-238.
[43] V. Lyubashevsky, C. Peikert, and O. Regev, On Ideal Lattices and Learning with Errors over Rings, Lecture Notes in Computer Science, 60(6) 2013, pp. 1-23.
[44] J. Alperin-Sheriff, C. Peikert, Faster Bootstrapping with Polynomial Error, Advances in Cryptology-CRYPTO 2014, Springer Berlin Heidelberg, 2014, pp. 297-314.
[45] V. Lyubashevsky, C. Peikert, and O. Regev, On ideal lattices and learning with errors over rings, Advances in Cryptology - EUROCRYPT 2010, LNCS, 6110, Springer, Heidelberg, 2010, pp. 1-23.
[46] Z. Brakerski, C. Gentry, and V. Vaikuntanathan, (Leveled) Fully Homomorphic Encryption without Bootstrapping, Acm Transactions on Computation Theory, 18, 2011, pp. 169-178.
[47] N. P. Smart, and F. Vercauteren, "Fully homomorphic encryption with relatively small key and ciphertext sizes," International Conference on Practice and Theory in Public Key Cryptography Springer-Verlag, 2010, pp. 420-443.
[48] N. P. Smart, and F. Vercauteren, "Fully homomorphic SIMD operations," Designs, Codes and Cryptography, 2011(1), 2014, pp. 57-81.
[49] Caltech-256 Object Category Dataset, in 2007. [Online]. Available: http://www.vision.caltech.edu/Image-Datasets/Caltech256/.
[50] Affine Covariant Regions Datasets, accessed on Sep.2004. [Online]. Available: http://www.robots.ox.ac.uk/ vgg/research/affine/.
[51] Yue Wu, Yicong Zhou, Joseph P. Noonan, Karen Panetta, and Sos Agaian. Image encryption using the sudoku matrix. Proceedings of SPIE - The International Society for Optical Engineering, 2010, 7708(1), pp.247-247.

Linzhi Jiang received the BS degree in mathematics education from the Shihezi University and the MS degree in applied mathematics from Guilin University of Electronic Technology, both in China. He is now the Ph.D. candidate from University of Electronic Science and Technology of China, and a Lecturer of West AnHui University, China. His research interests include applied cryptography, network security, and cloud computing security. Email: linzjiang@hotmail.com.

Chunxiang Xu received her B.Sc., M.Sc. and Ph.D. degrees at Xidian University, in 1985, 1988 and 2004 respectively, China. She is presently engaged in information security, cloud computing security and cryptography as a professor at University of Electronic Science and Technology of China.

Xiaofang Wang is now the Ph.D. candidate in School of Telecommunications Engineering Xidian University. Her research interests include Feedback shift registers, cryptographic Boolean functions and sequence design.

Bo Luo received the BE degree from the University of Sciences and Technology of China in 2001, the MPhil degree from the Chinese University of Hong Kong in 2003, and the PhD degree from The Pennsylvania State University in 2008. He is currently an assistant professor with Electrical Engineering and Computer Science Department at the University of Kansas. He is interested in information retrieval, information security, and privacy. He is a member of the IEEE Computer Society.

Huaqun Wang received the BS degree in mathematics education from the Shandong Normal University and the MS degree in applied mathematics from the East China Normal University, both in China, in 1997 and 2000, respectively. He received the Ph.D. degree in information security from Nanjing University of Posts and Telecommunications in 2006. He is currently a professor of Nanjing University of Posts and Telecommunications, China. His research interests include applied cryptography, network security, and cloud computing security.

APPENDIX A
MAIN OPERATIONS OF SIFT ALGORITHM

SIFT algorithm contains following aspects:

1. Gauss Scale Space Generation. We define scale space of an image as a function $L(x, y, \sigma)$. The convolution of a variable-scale Gaussian $(G(x, y, \sigma))$ with an input image $(I(x, y))$ produces scale space of an image. Namely, $L(x, y, \sigma) = G(x, y, \sigma) \otimes I(x, y)$, where symbol $\otimes$ is the convolution operation. Two-dimensional Gaussian kernel function is $G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2}$. SIFT algorithm recommends that DoG can be achieved by substraction of images on two adjacent Gaussian scale space. Then, the response value image $(D(x, y, \sigma))$ of DoG is used to detect the feature points. Namely, $D(x, y, \sigma) = (G(x, y, k\sigma) - (G(x, y, \sigma)) \otimes I(x, y) = L(x, y, k\sigma) - L(x, y, \sigma)$, where k is a constant multiple on adjacent scale space.

2. Key Point Detection, Location and Optimization. Extremum detection is performed in DoG space. Such detection makes use of previous point as center and $3pixel * 3pixel * 3pixel$ as a neighborhood to determine whether current point is the local maximum or the local minimum. The above extreme point detection is in the discrete space, therefore, extreme point is not true extreme point. Usually, we make use of interpolation to get close to true extreme point. For a two-dimensional function, Taylor expansion is $f(x,y)(\approx f(0,0) + (\frac{\partial f}{\partial x}x + \frac{\partial f}{\partial y}y) + \frac{1}{2}(\frac{\partial^2 f}{\partial x \partial x}x^2 + \frac{\partial^2 f}{\partial x \partial y}xy + \frac{\partial^2 f}{\partial y \partial y}y^2))$. After locating key points, we will eliminate some of unstable key points. This kind of eliminated points have a very low response to DoG. DoG has a strong response to the edge of image, therefore, points at the edge of image are unstable feature points.

3. Determining Feature Point Direction. Determination of feature point direction includes computing on neighbourhood gradient direction and amplitude, histogram of gradient direction, and determining direction of feature point. Argument and amplitude can be computed in the following formula: $\sharp(x, y) = \sqrt{(Diff_x)^2 + (Diff_y)^2}$, $\theta(x) = arctan(\frac{Diff_y}{Diff_x})$, where $L(x + 1, y) - L(x - 1, y) = Diff_x$ and $L(x, y + 1) - L(x, y - 1) = Diff_y$. After computing gradient for Gaussian image on neighborhood of feature points, histogram is used to compute gradient direction and amplitude of pixels in neighbourhood. With histogram of gradient direction, the maximum value of histogram is found. Such direction is the direction of feature point.

4. Descriptor. Producing descriptor includes determining sub-sampling area of descriptor, descriptor generation and feature vector normalization. Descriptor of SIFT algorithm is a representation of Gaussian image gradient in the vicinity neighbourhood of feature point. It is usually expressed as a vector. To ensure rotation invariance of feature vector, it is necessary to rotate $x-$axis of the original image to same direction as the main direction. The rotation formula is as follows:

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} cos\theta & -sin\theta \\ sin\theta & cos\theta \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}.$$

After rotating direction of gradient on neighborhood image near feature point, the fixed size area is taken from rotated image, and divided equally into some sub-regions. Each sub-region contains a fixed number pixels. In each sub-region, histogram of gradient direction on eight directions is computed. The cumulative value of each gradient direction is plotted to form a seed point. In this case, gradient histogram of each sub-region is divided into 8 directions in the range of $0^\circ \sim 360^\circ$. Each seed point has 8 intensity information of gradient for each direction. Because there are 4×4 sub-regions, a total of $4 \times 4 \times 8 (= 128)$ data are obtained. Finally, these data form a 128 dimensional feature vector. Final step is to normalize feature vectors. In order to eliminate influence of illumination change, we need to normalize generated feature vectors.

APPENDIX B
LEVEL HOMOMORPHIC ENCRYPTION BASED ON NTRU

Let R denote a polynomial ring. Let d be a positive integer, and $\Phi_d(Y)$ $(\in Z[Y])$ be the cyclotomic polynomial. We define $R = Z[Y]/(\Phi_d(Y))$, which is the set of polynomials with integer coefficients of degree less than $n = \varphi(d)$, where φ is Euler's totient function. For $\forall a \in R$ (a polynomial), $a = \sum_{i=0}^{i-1} a_i Y^i$, $a_i \in Z$. Maximum norm of a is defined as $\|a\|_\infty = max_i\{|a_i|\}$. Let δ be the maximal norm expansion. $\delta = sup\{\|g \cdot h\|_\infty / (\|g\|_\infty \|h\|_\infty) : g, h \in R\}$. Let $d = 2^k$, $n = 2^k - 1$ and $\chi = D_{Z^n,\sigma}$. χ denotes a probability distribution on R. $D_{Z^n,\sigma}$ denotes the discrete Gaussian distribution. Let $R_q = R/_q R$ denote ciphertext space and $R_t = R/_t R$ $(1 < t < q)$ denote the message space. Module q is an integer. If there exists a polynomial $f^{-1} \in R$ such that $ff^{-1} = \widetilde{f}$, where $\widetilde{f} = \sum_{i=0}^{i-1} b_i Y^i$ with $b_0 = 1 \bmod q$ and $b_j = 0 \bmod q$ for $j \neq 0$, polynomial $f \in R$ is invertible modulo q.

The Scheme of LHEBN:

LHEBN-param-Gen(λ): Given the security parameter (λ). Output $(\chi_{key}, \chi_{error}, q, t, d, \varpi)$. $\varpi (> 1)$ is an arbitrary integer.

LHEBN-Key-Gen$(q, t, d, \chi_{key}, \chi_{error}), \varpi)$: Taking s and e from distribution $\chi_{error}^{\ell^3_{\varpi,q}}$. Compute $evk = \gamma = [f^{-1} P_{\varpi,q}(D_{\varpi,q}(f) \otimes D_{\varpi,q}(f)) + e + hs]_q \in R^{\ell^3_{\varpi,q}}$. Output $(pk, sk, evk) = (h, f, \gamma)$.

LHEBN-Enc(h, M): Taking s and e from distribution χ_{error}. Output ciphertext $c = [\lfloor q/t \rfloor [M]_t + e + hs]_q \in R$.

LHEBN-Dec(f, c): Given $sk = f$, decrypt a ciphertext c to obtain plaintext $M = [\lfloor \frac{q}{t}[fc]_q \rfloor]_t$.

LHEBN-KeySwitch$(\widetilde{c}_{LHEBN_{Mult}}, evk)$: Compute $[< D_{\varpi,q}(\widetilde{c}_{LHEBN_{Mult}}), evk >]_q$.

FHEBN-Add(c_1, c_2): $c_{LHEBN-Add} = [c_1 + c_2]_q$.

FHEBN-Mult(c_1, c_2): Compute

$$\widetilde{c}_{LHEBN-Mult} = [\lfloor \frac{t}{q} P_{\omega,q}(c_1) \otimes P_{\omega,q}(c_2) \rfloor]_q,$$

$$FHEBN_{KeySwitch}(\widetilde{c}_{LHEBN_{Mult}}, evk) = c_{LHEBN-Mult}.$$