Journal of Information Hiding and Multimedia Signal Processing ©2016 ISSN 2073-4212
Ubiquitous International Volume **7**, Number **2**, March **2016**

The Encryption Scheme for DICOM Based on the Homomorphic Higher Degree Residue System in Cloud Storage

Yibing Wang

Center of Computer Teaching
Anhui University
No.111 Jiulong Road, Hefei, Anhui, 236061, China
wyb@ahu.edu.cn

Received August, 2015; revised October, 2015

Abstract. *With the rapid development of the cloud technology, more and more users outsource their complex data management to the cloud server. When managing data in a more economical and more flexible way, personal privacy and data security are always worried about. Aiming at the security and privacy issues of DICOM, a novel encryption algorithm called Homomorphic Higher Degree Residue Numbers Encryption, is presented in this paper with which users can directly operate ciphertext at the circumstance of no plaintext and it effectively protects users' privacy. Further, it gives a model of protecting strategy for DICOM in cloud computing and may store data in a distributed way in every cloud server with the ciphertext form. The results of the simulation experiment indicate this algorithm has good safety and executive efficiency.*
Keywords: Homomorphic higher degree residue, DICOM medical image, Multiple ciphertext shares, Edge detection.

1. **Introduction.** The technology of the cloud computing, a revolution of information technological industrial field, has been an important direction of the coming development of information technological industrial field [1]. Cloud computing integrates computing resource on internet into very-large-scale resource pool [2], and offer to users in all kinds of forms. The cloud storage technology, a concept extended and progressed from the cloud computing, is a system for gathering various of different types of storage device on internet with utility software through the function of cluster application, grid technology, or distributed file system, etc., making them cooperatively work and together completing the external functions of data storage and business access.

In cloud storage, users keep data in the cloud server, which results in completely losing the ability to control their own data, so that they are afraid of the problems of security storage and privacy leaking about data, thus, they require cloud service providers to provide an effective security guarantee, in order that they can trust the data security and integrity in the new surrounding. Compared to the traditional storage forms, users can not fully trust the cloud storage, consequently, there are some difficulties for spreading and popularizing the cloud storage service. So, data security of cloud storage has become one of main factors to hinder developing and popularizing the cloud computing.

DICOM (Digital Imaging and Communications in Medicine) is a standard, formulated by American College of Radiology and National Electrical Manufacturers Association, and compared to other image formats, besides image information of patients, it includes

many data related to patients' fundamental state, such as name, age, medical record and so on, those of which may assist doctors to diagnose the illness better. Following the popularization of DICOM standard and the development of cloud computing, medical images are easy to be assaulted by hackers or modified by malware, because of involving to the privacy of patients and the responsibility of hospital, it is fairly vital how to effectively encrypt the medical image in cloud storage. Datum shows that America on account of data leaking causes economical damage in medical care alone high to 70 billions every year. [3] Till 2017,it is estimated that the expense of health care alone in America will reach 54 billions one year in cloud computing.

The security of medical image files can not be ignored, however, few research about encryption is done. Based on the clinical experiences, Patel *et al.*[4] discussed the challenges for DICOM Image Management and the offered cloud service model for Medical Imaging applications on cloud computing systems, and analyzed the key problems in storage of DICOM in detail. Andrs *et al.*[5] put forward a new strategy, which involved processes of extracting, anonymizing, and serializing matadata comprised in DICOM medical images into RDF/XML. Finally, these processes allowed for semantically enriching and sharing the metadata of DICOM medical files through the Linked Health Data cloud. As to the problems of medical information sharing, Liu *et al.*[6] proposed and constructed the middleware technology based on DICOM, which can successfully realize the remote storage and communication of medical images. While, all these methods mentioned above no more than dealt with the images without protecting the users' privacy. Based on the design principle of AES, Xiang *et al.*[7] proposed an improved algorithm to encrypt the DICOM images, which could effectively maintain the compatibility of DICOM file format, but this method come across the disadvantage of too large dimension of the ciphertext. The traditional encryption schemes consider plaintext as binary data, and take no account of the nature of medical image, which results in the algorithm not much adapting to DICOM image. Two aspects embody as following: (1) The data of images have the characteristic of multidimensional distribution, which makes traditional way of block encryption may reveal geometric distribution information of original image content, (2) Traditional encryption destroyed the format of the image data, further to cause the decoder to work abnormally.

Wang *et al.*[8] put forward an efficient linear homomorphic encryption scheme based on R-LWE (ring-learning with errors). In this scheme, the data was encrypted first and then stored in cloud with a distributed manner, which could effectively meet the needs of users' privacy protection. Due to the linear homomorphic scheme, in which the data can only dealt via linear operation in cloud but with great restriction of its applications. Fully homomorphic encryption [9], beginning with the conception of homomorphic encryption referred by Rivest, etc in 1978, is an encryption algorithm permitting directly making random algorithm for ciphertext in the case of no key. This special property makes fully homomorphic encryption a broad application prospect, for example the cloud computing security, ciphertext retrieval, secure multi-party algorithm, anonymous voting and so on. Before 2009, many kinds of homomorphic encryptions only supported homomorphism of addition or multiplication to realize simply and easy statistic analysis, all of which had no fully homomorphic, until 2009, Gentry constructed the first real fully homomorphic encryption scheme, which simultaneously held homomorphism of addition and multiplication, which can ensure that the operation is taking place after encrypting sensitive data encrypted and won't leak data information [10]. From 2009 to now, a lot of fully homomorphic encryption scheme appeared in succession, realized and optimized. [11-16] Such as, Ren *et al.*[11] introduced the mechanism of fully homomorphic encryption, and gave a data security framework of cloud computing, which made certain of data in cloud

computing to transfer securely from users to cloud server or from cloud server back to users and the storage security in cloud server, and was convenient to users' retrieving, but existed the problem of an over-large amount of algorithm. Brakerski *et al.*[12] put forward the mechanism of fully homomorphic encryption based on learning with errors (LWE), and made use of the "heavy linearization technology" to realize the fully homomorphic algorithm ability in the field of ciphertext. However, owing to depend on a set of extra operation of evaluation key in heavy linearization process and the direct proportion of size and the number of ciphertext multiplication, the size of evaluation key became one serious defect for this algorithm while calculating the complicated ciphertext calculation. In order to settle the problem of over-size of evaluation key, Guang *et al.*[13] put forward one fully homomorphic encryption mechanism based on identity, which can effectively overcome the influence of the over-large size of evaluation key for fully homomorphic encryption application efficiency, as well as saving the expenditure of identifying and managemen, but this way had some safety potential issues, namely existing the risk of key leaking when facing attacks. The above fully homomorphic encryption algorithms had their own advantages, but all had no breakout in structure way, which related to the radical reason of low efficiency, and still have a distance away from actual application. Gomathisankaran *et al.*[14] put forward a novel encryption system, HORNS, that exploits the inherent parallelism present in the cloud and uses residue number system to create multiple ciphertext shares.

Based on the above analysis, motivated by the work in [14], the Homomorphic Higher Degree Residue Numbers encryption is proposed in this paper, and applied to DICOM medical image in cloud storage, whose effectiveness and superiority have been proved by the experiments.

The latter part of this paper demonstrates as following: the second part will describe the related knowledge about homomorphic encryption in the first place, next to introduce the Homomorphic Higher Degree Residue Numbers encryption, the third part will describe the simulation experiments and enumerates the results, as well as to analyze the safety and efficiency of this proposed algorithm; the fourth part will make a conclusion.

2. Homomorphic Higher Degree Residue Numbers encryption system.

The homomorphic higher degree residue numbers encryption we proposed is a random encryption method. Compared to other algorithms, it can improve the performance with the following advantages. (1) Due to homomorphism, users can make random algorithm directly for ciphertext without key via this homomorphism algorithm. (2) Encryption speed grows exponentially with very high efficiency as for the homomorphic higher degree residue numbers encryption is a modular exponentiation computing with fixed base number. (3) The method can save bandwidth by small expansion ratio, while some other encryption algorithms always encrypt with large expansion rate, maybe more than 1 KB enciphered data generated in case of some bits of less information encrypted.

2.1. pre-knowledge.

If the plaintext is m , cryptographic operation is E, then the result will be e after being encrypted.

$$e = E(m), m = E'(e) \tag{1}$$

If the plaintext had operation f, then the F can be consumed by E, which results in:

$$F(e) = E(f(m)) \tag{2}$$

The E is a homomorphic encryption for f. And it is called as fully homomorphic encryption, if the corresponding F can be produced for any complex plaintext operation f.

Fully homomorphic encryption ensured that the data processor can only process data without knowing of data context. This technology can be widely applied in current field of Cloud Computing. There are lots of institute who are insufficient in data processing need to entrust a data processing third party such as hospital who is obliged to protect the patients' privacy. They can get a favorable treatment via entrusting the data to third party (i.e. The Cloud Computing Center). But because it is immoral and illegal to give the data to third party directly. While the fully homomorphic encryption realized the target that medical institutions can give the data to third party after been encrypted which will be handed back after been processed, in which the data is completely transparent for third party.

2.2. **Homomorphic Higher Degree Residue Numbers encryption.** To convenient stating, several conceptions are introduced:

1. Higher Degree Residues.

Higher Degree Residues refers to that assuming n is a RSA[8] modulus, and p and $\varphi(n)$ are integers which are prime numbers each other, the c is the higher degree residue for n and p from formula $c = x^p \ mod \ n$.

2. Expansion Ratio.

Expansion Ratio is the ratio between ciphertext length and plaintext length.

3. The Chinese Remainder Theorem.

Generally speaking, if the prime number n can be divided into $n = p_1 \times p_2 \times \cdots \times p_i$, among which some prime numbers could be appeared for several times or equal to others, the equation group $(x \ mod \ p_i) = a_i, i = 1, 2, \cdots, t$ have an only value and x is less than n.

The homomorphic higher degree residue numbers encryption in this paper can not only meet the safety requirement of semantics with a comparatively small expansion ratio so as to save bandwidth but also it is homomorphic. The algorithm is as follows:

(1)Assuming RSA modulu is $n = pq$ and $\varphi(n)$ is Euler function for n, i.e. the number of positive integers which is less than n and they are prime numbers to each other, $\varphi(n) = (p-1) \times (q-1)$.

(2)Assuming σ is square-root-free B-smooth odd integer, where B is small integer and $\varphi(n)$ can be divided by σ , in which σ and $\varphi(n)/\sigma$ is prime number to each other. In general, it adopts $|B| = 10$, and $\sigma > 2^{160}$ is suggested.

(3)Assuming the following conditions are met: $g^k = 1 \ mod \ n$, where $k = l \times \sigma$, among which l is an integer.

(4)Let n, g are public keys and p, q, σ are private keys, among which the σ is optional item.

(5)When encrypting information, m is divided into m_i which is less than σ. For each group, the c_i ($c_i = x^\sigma g^{m_i} \mod n$) shall be computed, among which x is a random number smaller than n. Some pre-processing technique can accelerate encryption during the encryption. There are no requirements for that p_i is a prime number. σ should not be known. In fact, it would be fine that the p_i is prime number to each other. For example, p_i can be a square to one prime number.

(6)The original text is figured out via the Chinese Remainder Theorem.

2.3. **The Key Generation.** The usual method to generate key is that select even number (noted as k) of different small odd prime numbers and compute $u = \prod_{i=1}^{k/2} p_i$, $v = \prod_{i=\frac{k}{2}+1}^{k} p_i$ and $\sigma = uv = \prod_{i=1}^{k} p_i$, and select two big prime numbers a and b, let $p = 2au + 1$ and $q = 2bv + 1$ are prime numbers to each other and figure out the modulus $n = p \times q$.

There are lots of defects to generate keys via such approach such as inefficiency and time-consuming, such defect is particularly obvious for a lager modulus n. The cause of this result lies in cases such as it is required to select a in an appropriate value range as well as judge $p = 2au + 1$ is a prime number or not. And when the two conditions are met, the selection is succeed. But this process is extremely inefficient. So does for b selection. This paper perfected the above approach to generate a key and the processes are as follows:

STEP 1. Firstly, a, b, u and v are generated. This step is unrelated with the judgment of prime number for p, q.

STEP 2. Select a couple of prime numbers $p^{'}, q^{'}$ with 24bit, among which they will not be applied in encryption and ensure the results of $p = 2aup^{'} + 1$ and $q = 2bvq^{'} + 1$ are prime numbers. So it is suggested that when selecting $p^{'}$ and $q^{'}$, the $p^{'}q^{'}$ and σ are prime numbers each other and $p^{'} \neq q^{'}$.

In terms of selecting a base number g, we can randomly select an integer g and check it whether its degree is $\varphi(n)/4$ or not, i.e. to judge whether the $g^{\varphi(n)/4} = 1 \ mod \ n$ make sense. And ensure that the g have p_i square root or not. That is to say each $g^{\varphi(n)/p_i} \neq 1 \ mod \ n$ should be ensured to be an equation for $i \leq k$.

The success probability is

$$\pi = \prod_{i=1}^{k}(1 - \frac{1}{p_i}) \tag{3}$$

Get the logarithm from two sides of Eq. (3), which can be expressed as

$$ln(\pi) \approx -\sum_{i=1}^{k}\frac{1}{p_i} \tag{4}$$

In case $p_i(i \leq k)$ are the former k prime numbers, then it can be approximately reckoned as $-lnlnk$ and the total approximate probability is $\pi \approx 1/lnk$, which is absolutely acceptable.

2.4. **Decryption Process.** According to the Chinese remainder theorem, assume $\sigma = p_1 \times p_2 \times \cdots \times p_i \times \cdots \times p_k$, where p_i is prime divisor of σ. It can be known that there is an integer m to let the following equation group set.

$$m \ mod \ p_i = m_i, i = 1, 2, \cdots, k \tag{5}$$

That is to say if the m_i is available separately the m can be confirmed.

Given the ciphertext c and assuming x^{σ} is fixed (noted it as 1). $c = g^m \ mod \ n$. In order to confirm the value of m_i, two processes shall be operated.

STEP 1. Compute: $c_i = c^{\frac{\varphi(n)}{p_i}} \ mod \ n$:

Assuming $y_i = \frac{m - m_i}{p_i} \ mod \ n$, and

$$c_i = c^{\frac{\varphi(n)}{p_i}} = g^{\frac{m\varphi(n)}{p_i}} = g^{\frac{(m_i + yp_i)\varphi(n)}{p_i}} = g^{\frac{m_i\varphi(n)}{p_i}} g^{y_i\varphi(n)} = g^{\frac{m_i\varphi(n)}{p_i}} \ mod \ n \tag{6}$$

STEP 2. The value of m_i can be found via comparison between the results and all possible power operation results of $g^{\frac{j\varphi(n)}{p_i}}$.

From the above, the computing method of plaintext m can be shown in by the following form:

Procedure HRDecry()

{

for $i = 1$ to k
{
$c_i = g^{\frac{\varphi(n)}{p_i}} \ mod \ n$
for $j = 0$ to $p_i - 1$
{
if $(c_i == g^{\frac{j\varphi(n)}{p_i}} \ mod \ n)$ then
$m_i = j$
}
}
ChineseRemainder($\{m_i\}, \{p_i\}$)
}

In the proposed algorithm in this paper, x^σ is a random number, the ciphertext will be confirmed by $c_i = x^\sigma g^{m_i} \ mod \ n$, but the decryption method is in accordance with the decryption process. Because in spite of the ciphertext multiplied x^σ, but in the process of decryption, the ciphertext is eliminated because of the exponentiation arithmatic to $\frac{\varphi(n)}{p_i}$, that is :

$$c_i = c^{\frac{\varphi(n)}{p_i}} = x^{\frac{\sigma\varphi(n)}{p_i}} g^{\frac{m\varphi(n)}{p_i}} = x^{\varphi(n)} \prod_{j \neq i} p_i g^{\frac{m\varphi(n)}{p_i}} = g^{\frac{m_i\varphi(n)}{p_i}} \ mod \ n \tag{7}$$

After the plaintext being processed by homomorphism higher degree residue numbers encryption system, multiple ciphertext shares can be produced. Because of the characteristics of super-large scale computing power of the cloud, it can concurrent processing all the generated multiple ciphertext shares. The security of the encryption algorithm mentioned above is not only based on the refractory hypothesis of factorization problem of large number, but also the refractory hypothesis of higher degree residue numbers. Due to the homogeneity and randomness of this algorithm, it can conduct operations of arithmetic of addsubtractmultiply directly, and it can meet the requirements of semantic security, compared with other algorithms, its expansion rate $|n|/|\sigma|$ is relatively small.

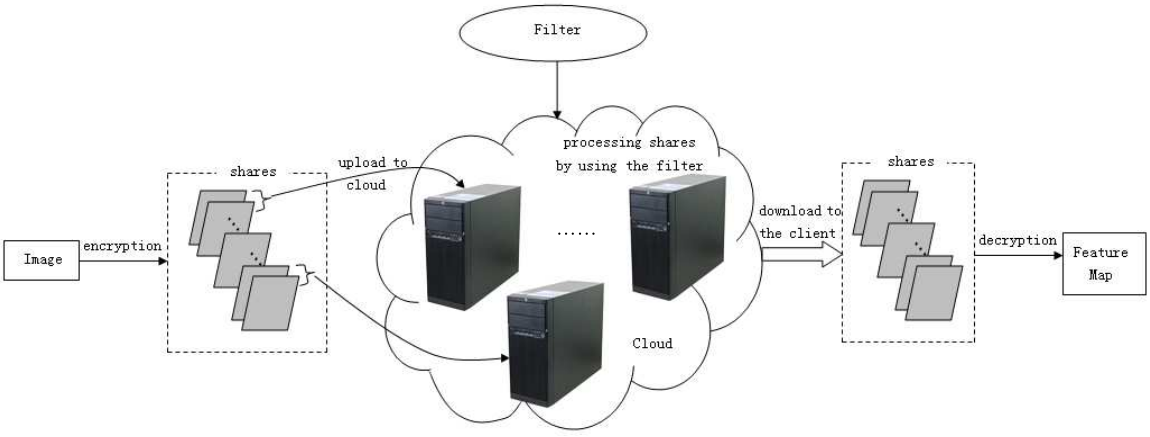


FIGURE 1. The system diagram

Users' data can't be accessed by users locally in cloud storage, and are stored in uncontrollable cloud pace in a distributed way. Each data has borderless characteristics, so users should be assured: their data can be accessed only to themselves, even the cloud servise providers don't have access permission of users' data. For this reason, this paper proposes homomorphic higher degree residue numbers encryption scheme to configure the

protection strategy model of DICOM of the cloud storage, as shown in figure 1. In this system, data is always presented in the cloud in the form of ciphertext, and is storaged in each servers of the cloud in a distributed way, that is the cloud service provider can operate the ciphertext data according to the users need, but they can't understand the plaintext information. Using the homomorphic characteristic of homomorphic residue numbers encryption, cloud services can conduct aggregate process to ciphertext data so that could achieve the effect of compression. In addition, the expansion rate of the ciphertext is low, that is through the process of encryption of the plaintext, the corresponding ciphertext size won't be too big, so it can save the users' storage costs.

3. **Simulation Experiment.** We conduct data simulation experiment through Python platform and adopt DCMTK to realize the reading and writing of DICOM. In experiments, machine configuration is as follows: Intel Core 3.10GHZ, 8GB memory,64-bit Linux operation system, plaintext image is DICOM of the standard 440×440 pixel. Experiments are set up in a community service oriented cloud computing environment which is based on Web, with KVM and Hadoop HDFS as the underlying support technology, and deployed on the integration which is integrated by 40 sets servises. We arrange two experiments, the first set of experiments is to verify sensitivity about proposed algorithm towards the data; the second set is to verify the validity and superiority of the designed systerm in this paper.

3.1. **Sensitivity Analysis. 3.1.1 Sensitivity Analysis For Plaintext**

In order to test the sensitivity about proposed algorithm in this paper towards the plaintext, we use two nearly identical plaintext images (chest.dcm from DICOM, among which only the pixel value in position (0,0) is different), encrypt to them, we get the difference images of corresponding ciphertext after the encryption, as shown in figure 2, compare their pixel value, in figure 2, white pixels represent the same part and black pixels represent different parts. We find that, the encrypted images are almost completely different, known by calculation, only when the two images pixel value in position (0,0) is not the same, the 99.73% of pixels of the corresponding ciphertext are different.

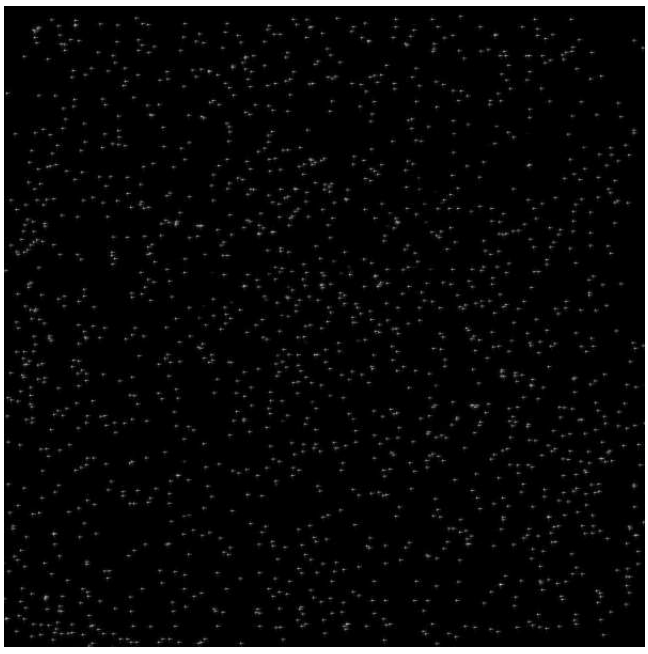
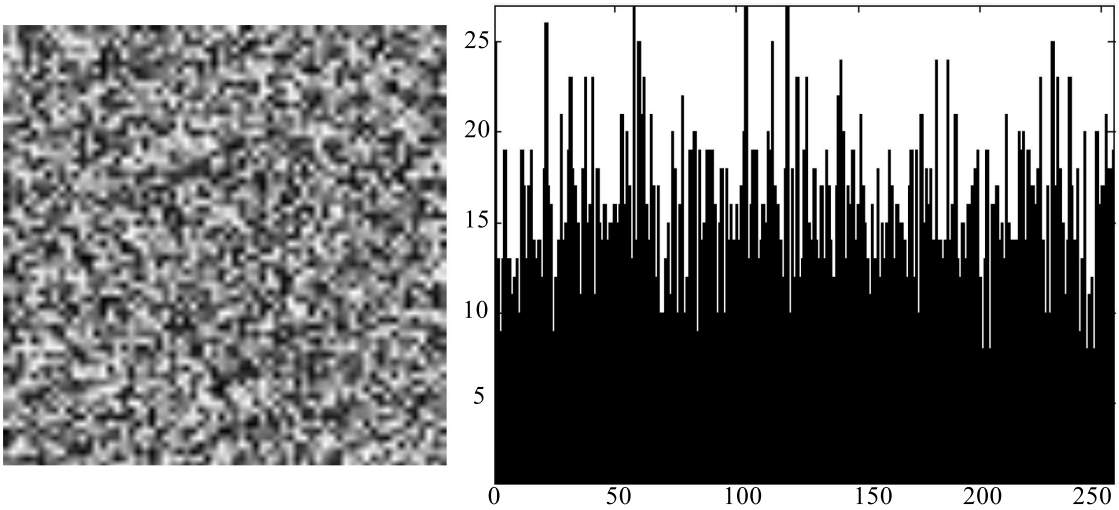


FIGURE 2. difference-value image of the ciphertext

3.1.2 Sensitivity Analysis For The Key

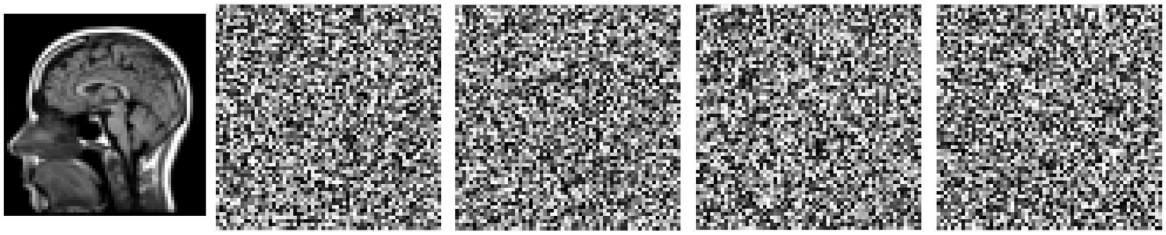
In this experiment, we change the original key slightly, modify k_2 to 9 from 7, the other key stays the same, the corresponding decrypted image is shown in figure 3. It is observed that, decrypted image is in the random distribution state, its histogram is uniformed. That is to say, even though there is tiny difference between encryption key and decryption, the system also can decrypt correctly. It is showed that algorithm in this paper can resist various attacks based on key sensitivity. Through the above analysis, it can be found, the proposed algorithm has favorable diffusing effect and meet the safety requirements.



(a) Decrypted image (b) Histogram of decrypted image

FIGURE 3. The results of sensitivity analysis for the key

3.2. **Analysis Of Effectiveness And Efficiency.** In these experiments, we encrypt the input image into multiple shares by the proposed algorithm. Then the shares are sent to the simulated cloud system, which returns to the client the processed shares that are to be aggregated for decryption. In order to facilitate the analysis, we cut the 440×440 standard DICOM image to 80×80 in advance. Generally speaking, for any image in our algorithm shares are generated. So the system will generate 6399 ciphertext shares, each share is itself a 80×80 matrix. We selected four copies from 6399 shares, as shown in figure 4. From figure 4, we can find that the content of each share shows no correlation to the original image. It demonstrates that our method hides the original image successfully.



(a) Original image (b) Share 1 (c) Share 2(d) Share 3(e) Share 4

FIGURE 4. Result images

Table I reports the time used to process the images with our method. Due to generate the multiple ciphertext shares, the time used for processing images is significantly greater. However, in the cloud, the computation time can be reduced, as shown in the fourth row of Table I.

TABLE 1. Time used for processing the images (in 10^{-1} seconds)

Image size	16×16	80×80
Without encryption	1.69	2.57
With encryption	127.03	5069.49
With encryption and cloud	113.17	4679.31

4. **Conclusion.** The related new type of password technology becoming the research hotspot in the current field of cryptography because of the rapid development of cloud computing, the fully homomorphic encryption is able to realize various operations of the ciphertext, so it has important application value in the cloud computing environment. According to the demand of the privacy of image data, aiming at the characteristics of DICOM, this paper presents protection strategy model of DICOM medical image, data can be distributed stored on the cloud servers in the form of ciphertext. In the view of security issues of image, this paper presents homomorphic higher degree residue numbers encryption, but in the situation of not knowing plaintext, direct manipulation on ciphertext can improve storage efficiency of data greatly, moreover, the algorithm under the premise of meeting the safety requirement, inflation is reduced greatly. At last, we conduct simulation experiment to the proposed algorithm, it turns out that, algorithm in this article not only has favorable security and execution efficiency, but also maintain the compatibility of DICOM file format.

Acknowledgment. This work was supported by the National Science Foundation of China under Contracts 51477001 and 61402003. The authors would like to thank the Associate Editor and the Reviewers for their valuable comments and suggestions.

REFERENCES

[1] Jena, K. Rabindra, P. K. Mahanti, Computing in the cloud: Concept and trends. *International Review on Computers and Software*. vol.6, no.1, pp.1–10, 2011.

[2] T. Wang, J. P. Yu, Y. J. Yang, Linear Homomorphic Encryption Scheme for Privacy Protection of Cloud storage. *Signal Processing*. vol.29, no.11, pp.1463–1469, 2013.

[3] G. Mahadevan, X. H. Yuan, P. Kamongi, Ensure Privacy and Security in the Process of Medical Image Analysis. *2013 IEEE International Conference on Granular Computing (GRC)*, pp.120–125, 2013.

[4] Patel, G. J. Hai, DICOM Medical Image Management the challenges and solutions: Cloud as a Service (CaaS), *2012 3rd International Conference on Computing, Communication and Networking Technologies*, *ICCCNT*, July 26, 2012 - July 28, 2012

[5] T. Andrs, L. C. Alexandra, S. Vctor *et al.*,RDF-ization of DICOM medical images towards linked health data cloud, *6th Latin American Congress on Biomedical Engineering*, vol.49, pp. 757–760, 2014.

[6] L. J. Liu, Q. S. Huang, CloudDICOM: A large-scale online storage and sharing system for DICOM images, Advanced Materials Research, vol.756–759, pp. 2037–2041, 2013.

[7] X. Tao, C. Y. Yu, et al. Research on the Security of DICOM Medical Images Based on Improved AES Encryption Algorithm, *Acta Electronica Sinica*, vol.40, no.2, pp.406–411, 2012.

[8] T. Wang, J. P. Yu, Y. J. Yang, Linear Homomorphic Encryption Scheme for Privacy Protection of Cloud storage, *Signal Processing*, vol.29, no.11, pp.1463–1469, 2013.

[9] R. Rivest, A. Shamir, and L. Adleman, A methos for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, vol.21, no.2, pp.120–126, 1978.
[10] V. Vaikuntanathan, Computing blindfolded: new developments in fully homomorphic encryption. Proc of the 52nd Annual Symposium on Foundations of Computer Science. *Washington DC: IEEE Computer Society*. pp. 5–16, 2011.
[11] F. L. Ren, Z. X. Zhu, A cloud computing security solution based on fully homomorphic encryption. *Journal of Xi'an University of Post and Telecom.* vol.18, no.3, pp.92–95, 2013.
[12] Z. Brakerski, V. Vaikuntanathan, Efficient fully homomorphic encryption from (standard) LWE[A]. *Proceeding of IEEE 52nd Annual Symposium on Foundations of Computer Science(FOCS2011)*. Palm Springs, CA, USA. pp.97–1068, 2011.
[13] Y. Guang, J, L. Fei, Identity-based fully homomorphic encryption from learning with error problem. *Journal on Communications*. vol.35, no.2, pp.111–117, 2014.
[14] G. Mahadevan, N. Kamesh, T. Akhilesh, HORNS: A semi-perfectly secret homomorphic encryption system. 3rd International Conference on Computing Communication and Networking Technologies. India, 2012.
[15] C. Gentry, A. Sahai, B. Waters, Homomorphic encryption from learning with errors: conceptually-simpler, asymptotically-faster, attribute-based. *Proc of the 33rd Annual Cryptology Conference*. Berlin: Springer. pp.75–92, 2013.
[16] Z. BRAKERSKI, Fully homomorphic encryption without modulus switching from classical GapSVP. *Proc of the 32nd Gryptology Conference*. Berlin: Springer. pp.868–886, 2012.
[17] R. Bendlin, I. Damgard, C. Orlandi, S. Zakarias, Semi-homomorphic Encryption and Multiparty Computation. Kenneth G. Paterson. *Proceedings of 30th Int. Conf. on the Theory and Applications of Cryptographic Techniques (EUROCRYPT)*. Berlin: Springer. pp.169–188, 2011.