Secure Image Denoising over Two Clouds

Xianjun Hu, Weiming Zhang(✉), Honggang Hu, and Nenghai Yu

Key Laboratory of Electromagnetic Space Information, CAS,
University of Science and Technology of China, Hefei, China
hxj2012@mail.ustc.edu.cn, {zhangwm,hghu2005,ynh}@ustc.edu.cn

Abstract. Multimedia processing with cloud is prevalent now, which the cloud server can provide abundant resources to processing various multimedia processing tasks. However, some privacy issues must be considered in cloud computing. For a secret image, the image content should be kept secret while conducting the multimedia processing in the cloud. Multimedia processing in the encrypted domain is essential to protect the privacy in cloud computing. Hu et al. proposed a novel framework to perform complex image processing algorithms in encrypted images with two cryptosystems: additive homomorphic encryption and privacy preserving transform. The additive homomorphic cryptosystem used in their scheme causes huge ciphertext expansion and greatly increases the cloud's computation. In this paper, we modified their framework to a two-cloud scheme, and also implemented the random nonlocal means denoising algorithm. The complexity analysis and simulation results demonstrate that our new scheme is more efficient than Hu's under the same denoising performance.

Keywords: Secure image denoising · Image sharing
Random nonlocal means · Double-cipher

1 Introduction

Multimedia processing in the cloud has been widely used in recent years, such as photo-editing app Prisma[1], and video and photo editing app Artisto[2]. The cloud servers can offer high computation and large storage resources; client can outsource local large data and complex computing tasks to the cloud servers to save the local resource. However, cloud server is a third party, and it may not be trusted. The outsourced sensitive multimedia content may be leaked, which will lead to security and privacy issues. For outsourced storage, the simplest way to overcome these issues is to use traditional symmetric cryptography, such as

This work was supported in part by the Natural Science Foundation of China under Grant U1636201, 61572452, 61522210, and 61632013, and the Fundamental Research Funds for the Central Universities in China (WK2101020005).

[1] http://prisma-ai.com/.

[2] https://artisto.my.com/.

Y. Zhao et al. (Eds.): ICIG 2017, Part III, LNCS 10668, pp. 471–482, 2017.
https://doi.org/10.1007/978-3-319-71598-8_42

3DES or AES, to encrypt the outsourced sensitive multimedia content. While for outsourced multimedia processing, secure multimedia processing is still a huge challenging problem.

Signal processing in the encrypted domain is desired in cloud computing [2]. Modern cryptography provides some vital encryption schemes, such as homomorphic encryption [10,11,17,18,20,29,30,32], secret sharing [1,5,19,34], and secure multiparty computation [3,4,16,21,22,37], to handle multimedia processing in the encrypted domain.

The concept of homomorphic encryption is first proposed by Rivest et al. [32] as privacy homomorphism. Since then, nearly 30 years, only partial homomorphism has been achieved, such as Elgamal cryptosystem [18] can perform multiplicative homomorphism, and Paillier cryptosystem [30] can perform additive homomorphism. A breakthrough of fully homomorphic encryption was achieved by Gentry in 2009 [20]. After that, full homomorphic encryption is constantly being improved [10,11,17,29]. Even though for practical application, homomorphic encryption is inefficient, signal processing in the encrypted domain based on homomorphic encryption is still a hot research direction. Encrypted domain discrete cosine transform and discrete Fourier transform based on Paillier cryptosystem were implemented by Bianchi et al. [6,8]. And then encrypted domain discrete wavelet transform and Walsh-Hadamard transform based on Paillier cryptosystem were implemented by Zheng et al. [38–40]. A privacy-preserving face recognition system based on fully homomorphic cryptosystem was presented in [36], and meanwhile, fully homomorphic encryption was applied to genetic data testing [15].

Secret sharing scheme was independently proposed by Blakley [9] and Shamir [34]. The Shamir's secret sharing scheme is the most frequently used, which supports additive homomorphism [5]. Some secure signal processing schemes based on secret sharing were proposed. A privacy protect wavelet denoising with secret sharing was presented in [33]. However, after every multiplication operation, each party needs to communicate with each other to renormalizing the threshold. In [27], Lathey et al. proposed to perform image enhancement in the encrypted domain with multiple independent cloud servers, and the novelty of their work is that it can deal with arithmetic division operation for nonterminating quotients. In [28], secure cloud-based rendering framework based on multiple cloud centers was presented, and to overcome the computation of real number operation in the encrypted domain, secret sharing scheme without modulus was adopted.

Secure multiparty computation was proposed by Yao [37], which can be used as a general method to perform encrypted domain computation [4,21]. The BGW protocol is a good example [4]. General multiparty computation based on linear secret sharing scheme was proposed [16]. In [31], a scheme for wavelet denoising was proposed, which is based on Lattice cryptography. However, maybe it is not efficient to deal with nonlocal means image denoising algorithm. In [41], Zheng et al. proposed to perform privacy-preserving image denoising using external cloud databases, and their scheme is based on two cloud servers, which one is the image database for storage encrypted image patches, and the other cloud server is to generate the garbled circuits and send them to image cloud database to perform comparison operations. For a large image, the communication load between these two cloud servers is considerably huge.

Image denoising in the encrypted domain is a concrete research in secure multimedia processing. In [23,24], Hu et al. proposed a double-cipher scheme to perform nonlocal image denoising. Two encryption schemes, partial homomorphic encryption and privacy-preserving transform were adopted in their scheme. The bottleneck in their scheme is the efficiency of partial homomorphic encryption, which causes cipher expansion and the cloud server performing large computation. In this paper, we presented a new scheme with two non-colluding servers, and the new scheme is more concise and efficient. It can achieve the same denoised performance, while the communication load between cloud servers and client, and the computation complexity in cloud servers side and client side are better than Hu's scheme.

The rest of this paper is organized as follows. In Sect. 2, we introduce Hu's double-cipher scheme in detail. A comprehensive introduction of our new scheme will be given in Sect. 3. We analyze the computation complexity and communication load about our scheme in Sect. 4. In Sect. 5, we give some discussion about our proposed scheme. Finally, Sect. 6 concludes this paper.

2 Double-Cipher Image Denoising

In this section, we describe the details of double-cipher scheme. Hu et al. proposed the double-cipher scheme in [23,24]. Monte Carlo nonlocal means image denoising algorithm [14] was adopted as an example to perform nonlinear operation in the encrypted domain. In their framework, the cloud server will get two different cipher images encrypted by two different encryption schemes: Paillier encryption [30] and privacy-preserving Johnson-Lindenstrauss (JL) transform [26] from the same image. The cloud server performed mean filter on the cipher image encrypted by Paillier encryption, while performed nonlocal search on the other cipher image generated by privacy-preserving transform. Here, we firstly present a full description of Hu's double-cipher scheme, and more details can be read in [24].

We can summarize the double-cipher scheme as three main algorithms: image encryption in the client side, secure image denoising in the cloud, and image decryption in the client side.

2.1 Image Encryption

Binarization attack presented in [24] shows that the cloud server can recover the cipher image through the strong correlation between adjacent image pixels, because spatial close image pixels tend to have similar or even identical pixel value. Therefore, to enhance the security, image scrambling was used to perform decorrelation before image encryption. Because of two encryption schemes, an n-pixel image I was performed two different image scrambling, block image scramble and pixel image scramble, with the same pseudorandom permutation sequence, respectively.

For block image scramble, the image I was first split with each pixel as the center in overlapping n image blocks with size $l \times l$. Then each block was made into a vector as an $n \times l^2$ matrix α. Here with the pseudorandom permutation sequence, matrix α was performed row scrambling to output a block scrambled image $\bar{I}$. While for pixel image scramble, the image I was scrambling by the same pseudorandom permutation sequence to output a pixel scrambled image $\tilde{I}$. The indices of rows in $\bar{I}$ corresponds to the indices of pixels in $\tilde{I}$, and this makes sure the encrypted image can be denoised.

A privacy-preserving Johnson-Lindenstrauss (JL) transform was proposed by Kenthapadi et al. [26] based on Johnson-Lindenstrauss theorem [25], which can preserve Euclidean distance, and Hu et al. used this privacy preserving JL transform on image encryption, which was performed in Algorithm 1. After Algorithm 1 performed, an $n \times k$ matrix E^{JL} can be generated as ciphertext, where $k < l^2$. Here we should mention that the size of E^{JL} is about k times larger than that of the original image I. The block size l was chosen as 5, and the projected dimension k was $9 \sim 18$ in [24].

For the second cipher image E^{Pail}, the client encrypted the pixel scrambled image $\tilde{I}$ pixel by pixel with Paillier encryption.

After encryption, the client uploaded the two cipher images to the cloud server.

Algorithm 1. JL Transform-based Private Projection

```
Input: $n \times l^2$ matrix $\bar{I}$; projected dimension $k$; Noise parameter $\zeta$.
Output: The projected $n \times k$ matrix $E^{JL}$.

1. Generate a $l^2 \times k$ $N(0, 1/k)$ Gaussian distribution matrix $P$;
2. Generate an $n \times k$ $N(0, \zeta^2)$ Gaussian distribution noise matrix $\Delta$;
3. $E^{JL} = \bar{I}P + \Delta$.
```

2.2 Secure Image Denoising

Image denoising can be described in a matrix-vector form as:

$$\mathbf{y} = \mathbf{w}\boldsymbol{I} \tag{1}$$

where $\mathbf{y}$, $\boldsymbol{I}$, and $\mathbf{w}$ are the matrix-vector form of noisy image, original image, and the weight of the filter, respectively.

The filter matrix $\mathbf{w}$ is computed from a nonlocal means kernel function K_{ij} [12, 13], representing the similarity between i-th and j-th image block:

$$K_{ij} = e^{\frac{-||\mathbf{y}(N_i) - \mathbf{y}(N_j)||^2}{h^2}}, \tag{2}$$

where N_i is an image block centered at i, and h denotes the smoothing factor.

In the encrypted domain, the kernel function K_{ij} can be calculated by JL transformed data matrix E^{JL}, so K_{ij} can be replaced as:

$$\tilde{K}_{ij} = e^{\frac{-||E^{JL}(i)-E^{JL}(j)||^2-2k\zeta}{h^2}}, \tag{3}$$

where $E^{JL}(i)$ denotes the i-th row of matrix E^{JL}.

Therefore, the estimated image $\tilde{\mathbf{y}}$ can be described as follows:

$$\tilde{\mathbf{y}} = \mathbf{D}^{-1}\tilde{\mathbf{K}}\mathbf{z} = \tilde{\mathbf{w}}\boldsymbol{I}, \tag{4}$$

where $\mathbf{D}$ is a diagonal matrix denoting a normalization factor.

The cloud server can perform encrypted the image denoising algorithm with the weight matrix $\tilde{\mathbf{w}}$ on the cipher image $E^{Pail}[\boldsymbol{I}]$. The denoised encrypted image is presented as follows:

$$E^{Pail}[\boldsymbol{I}'] = (E^{Pail}[\boldsymbol{I}])^{\tilde{\mathbf{w}}}. \tag{5}$$

Calculating the weight matrix $\mathbf{w}$ by the classic nonlocal means algorithm [12] is extraordinary time-consuming, because the computation complexity is about $O(n^2)$, and n is the number of image pixel. Monte Carlo Non-Local Means (MCNLM) [14] is a random sampling algorithm, and for each image pixel, it only selects a small number of image blocks to calculate the weight matrix, which was implemented in the encrypted domain to speed up the classic nonlocal means denoising algorithm in [24].

2.3 Image Decryption

After image denoising in the cloud server, the cloud server sent back the encrypted denoised image, and the client decrypted the cipher image $E^{Pail}[\mathbf{I}']$ pixel by pixel with Paillier decryption. At last, pixel inverse scramble was performed, and the client got the denoised image $\mathbf{I}'$.

3 Secure Image Denoising over Two Clouds

Paillier encryption is an additive homomorphic encryption, which brings large ciphertext expansion and causes heavy communication load between the cloud server and the client, and also the calculation of the modular multiplication and modular exponentiation in the cloud server is remarkably time-consuming. Therefore, to reduce this ciphertext expansion and avoid the modular operations in the encrypted domain, we modified their scheme to a new one with two cloud servers. In our new scheme, the cloud servers only need to perform normal addition and multiplication in the cipher images, and the computation complexity is much lower than previous one.

In this section, we present the details of our proposed scheme. In our scheme, we need two cloud servers to perform MCNLM, and the framework of our scheme is presented in Fig. 1. From this framework, we can see that the client also uses two different encryption scheme to encrypt the image. The client uses JL transform to get the cipher image E^{JL}, and uses the other encryption scheme (This encryption scheme will be described later.) to divide the image into two shares E^{S_1}, E^{S_2}. Then the client uploads E^{JL}, E^{S_1} to cloud server 1 (CS 1), and uploads E^{JL}, E^{S_2} to cloud server 2 (CS 2) as step 1 showed in the Fig. 1. As described above, MCNLM is a randomized algorithm, for solving the synchronization problem, CS 1 computes the sample indices, and sent the indices to CS 2 as step 2 showed. With the same sample indices, the two cloud servers can calculate the weight matrix with E^{JL}, and perform the linear denoising on E^{S_1} and E^{S_2}, respectively. After each cloud server completes the denoising algorithm, they sends back their denoised image shares $E^{S_1'}, E^{S_2'}$ to the client as step 3 showed. The client will get two denoised image shares, and the denoised image will be reconstructed.

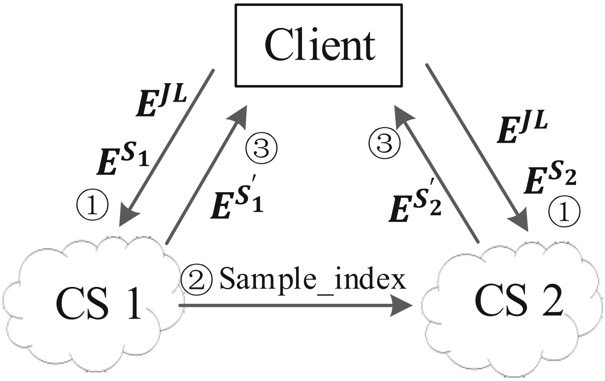

Fig. 1. Framework of two-cloud based secure image denoising.

Our new scheme is based on Hu's double-cipher scheme, and some procedures are the same, in order to simplify the description of our new scheme, we omit the same part and focus on the different part.

3.1 Image Sharing

For an n-pixel image I, and each pixel value is 8-bit, to encrypt this image, the client first generates a matrix E^{S_1} with n elements, and each element is randomly chosen from a uniform distribution. Then the client encrypts the image I as: $E^{S_2} = I + E^{S_1}$. The cipher image shares E^{S_2}, E^{S_1} are additive homomorphism, which can be used to replace the cipher image generated by Paillier encryption.

3.2 Image Sampling

MCNLM is a randomized algorithm, and the weight of each image pixel is computed from a subset of the image, if the two cloud servers independently compute the weight matrix, it will cause the two weight matrices different, and the following denoising fails. In order to solve the synchronization problem, we let one of the cloud servers perform random sampling, and sends the sampling indices to the other cloud server. Two sampling patterns were described in [14], which is uniform sampling and optimal sampling. Each pixel block is sampling based on a fixed probability in the uniform sampling pattern, while an optimization problem need to be solved in the optimal sampling pattern.

4 Complexity Analysis

In this section, the complexity of our proposed scheme will be analyzed. The complexity of the scheme includes communication complexity and computation complexity. We also compare our scheme with Hu's scheme.

4.1 Communication Complexity

First, we analyze the communication complexity of our proposed scheme. The cipher image E^{JL} should be uploaded to each cloud server, while the cipher image shares E^{S_1}, E^{S_2} should be upload to CS 1 and CS 2, respectively. And also the denoised encrypted image shares $E^{S'_1}$ and $E^{S'_2}$ should be sent back to the client. Therefore, for an n-pixel 8-bit image, the projected dimension k of JL transform is chosen as $9 \sim 18$, and there are two independent cloud servers. Thus the upload communication data is slightly more than $2 \times n \times (k+1)$ bytes, and the download communication data is slightly more than $2 \times n$ bytes. While in Hu's scheme for 1024-bit encryption key, the upload communication data is about $n \times (k + 256)$ bytes, and the download communication data is about $8n$ bytes by using ciphertext compression [7]. For $k = 12$ in our scheme, that is one-tenth the upload communication data of Hu's scheme, and a quarter the download communication data of Hu's scheme. The communication data between cloud servers and the client is significantly decreased. In our new scheme, CS 1 should send the sampling indices to CS 2, for sampling ratio is ρ, this communication data is $\rho n \log(n)$ bits, while in Hu's scheme, this is not required. For the sampling ratio is very small, most of the sampling indices are 0, while the sampling ratio is very big, most of the sampling indices are 1. The sampling indices can be compressed effectively. In Hu's scheme, the sampling ratio set to 0.01 is enough. We list the communication complexity in Table 1.

Table 1. Communication Complexity

	Hu's scheme	Our scheme
Upload	$n \times (k + 256)$	$2n \times (k + 1)$
Download	$8n$	$2n$
Cloud-to-cloud	None	$\rho n \log(n)/8$

4.2 Computation Complexity

The computation complexity in our scheme includes the client side and the cloud side. In the client side, client needs to perform image scramble, JL Transform, image sharing, and image reconstruction. Image sharing in our scheme is very concise, which can be efficiently computed. While in Hu's scheme, the client side needs to perform Paillier encryption, which is more complicated than image sharing. In the cloud side, the cloud server should the perform modular operations in Hu's scheme, while in our scheme, each cloud server only needs to perform the normal operations as in the plain image. Image decryption in Hu's scheme is also complicated operation.

On the client side, the difference between our scheme and Hu's scheme is image sharing and Paillier encryption, therefore, we only compare these two parts in our simulation. A simulation was given on an Intel i5 CPU at 2.5 GHz computer running Ubuntu 32-bit v13.04. Time cost of different parts for a 256 × 256 image is listed in Table 2, and we simulated Hu's double-cipher scheme by their fast algorithm implementation. We can see that our scheme in the client side is much faster than Hu's scheme. The Paillier encryption is more complicated than image sharing, which brings more calculation and time-consuming. So, our new scheme is more practical.

Table 2. Time cost in client side

	Paillier Encryption	Paillier Decryption
Hu's scheme	1.0	4.1
	Image sharing	Image reconstruct
Our scheme	0.1	0.1

On the cloud server side, in Hu's double-cipher scheme, the complicated modular multiplication and modular exponentiation need to be performed, while in our new scheme, the cloud servers only need to perform the normal addition and multiplication as in the plain image. The computation time of our proposed scheme approximately equals the plain MCNLM algorithm on the cloud server side.

5 Discussion

In this section, we give some discussion about our proposed scheme.

Security. In our new scheme, we adopted a very concise image encryption, image sharing, to replace Paillier additive homomorphic encryption. So in our scheme, we assume that the two cloud servers are non-colluding, and they are honest-but-curious. If we consider a malicious model, we need more complicated secure multiparty computation protocol, and this also can be implemented in our framework, which will increase much communication traffic and computation complexity for obtaining higher security.

In our scheme, CS 1 received a random matrix generated by the client, and this matrix is independent of the input image itself. Therefore, CS 1 can get nothing about input image from its image sharing. CS 2 gets an image matrix hiding by adding CS 1's random matrix. This image splitting method guarantees the security of the image content against cloud servers. The random matrix will be changed every time in the client side to encrypt the image.

Some optimizations. In our scheme, one cloud server needs to perform the image sampling and the other server waits for the sampling indices. A optimal scheme can be given in Fig. 2. The two cloud servers each select half of the image to perform image sampling and denoising, After completing its own denoising, the two cloud servers send their respective indices to the other party, and it can reduce the waiting time.

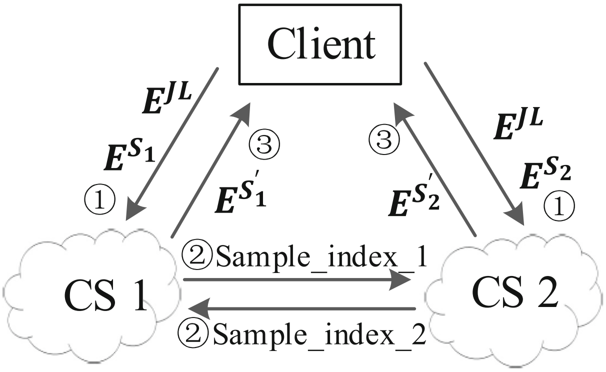

Fig. 2. An improvement of two-cloud based our secure image denoising

Our proposed scheme is based on two cloud servers, and we can also change our scheme to a multi-cloud framework based on secret sharing to resist colluding of cloud servers as showed in Fig. 3. Then the communication load between cloud servers and the client, cloud server to cloud server will increase with the number of the cloud servers. If we consider about other deterministic image denoising algorithm [35] in our framework, then the communication load between cloud servers can be omitted, and it will be more efficient.

As showed in Figs. 1 and 2 and complexity analysis in Sect. 4, our framework abandons the extraordinary complicated Paillier cryptosystem, the communication load between cloud servers and the client, and the computation cost in the cloud server are significantly decreased. Our new scheme can achieve the same image denoising performance as Hu's scheme.

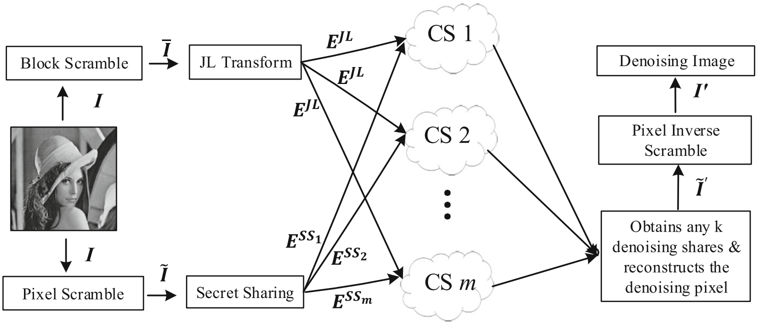

Fig. 3. A variant of our secure image denoising

6 Conclusion and Future Work

In this paper, we modified Hu's double-cipher scheme into a two cloud servers scheme, and gave some optimizations. In our scheme, the cloud servers can perform encrypted image denoising as same as in the plain image, and our proposed scheme almost does not increase the amount of calculation for each cloud server. The main drawback of our proposed scheme is probably that we should rent two non-colluding cloud serves, and the client should communicate with each cloud server. But we reduced the cipher expansion effectively, and the total communication load is still lower than Hu's scheme. The client side's computational complexity is significant reduction. The cloud servers don't need to perform complex modular operations in the encryption domain.

Efficient implementation of the multimedia nonlinear operation in the encrypted domain sill remains as a difficult problem. Working on more image processing algorithms in the encrypted domain are our future research direction.

References

1. Two verifiable multi secret sharing schemes based on nonhomogeneous linear recursion and LFSR public-key cryptosystem. Inf. Sci. **294**, 31–40 (2015). Innovative Applications of Artificial Neural Networks in Engineering
2. Aguilar-Melchor, C., Fau, S., Fontaine, C., Gogniat, G., Sirdey, R.: Recent advances in homomorphic encryption: a possible future for signal processing in the encrypted domain. IEEE Signal Process. Mag. **30**(2), 108–117 (2013)
3. Barak, B., Sahai, A.: How to play almost any mental game over the net - concurrent composition via super-polynomial simulation. In: 46th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2005), pp. 543–552, October 2005
4. Ben-Or, M., Goldwasser, S., Wigderson, A.: Completeness theorems for non-cryptographic fault-tolerant distributed computation. In: Proceedings of the Twentieth Annual ACM Symposium on Theory of Computing, STOC 1988, pp. 1–10 (1988)

5. Benaloh, J.C.: Secret sharing homomorphisms: keeping shares of a secret secret (Extended abstract). In: Odlyzko, A.M. (ed.) CRYPTO 1986. LNCS, vol. 263, pp. 251–260. Springer, Heidelberg (1987). https://doi.org/10.1007/3-540-47721-7_19
6. Bianchi, T., Piva, A., Barni, M.: On the implementation of the discrete fourier transform in the encrypted domain. IEEE Trans. Inf. Forensics Secur. **4**(1), 86–97 (2009)
7. Bianchi, T., Piva, A., Barni, M.: Composite signal representation for fast and storage-efficient processing of encrypted signals. IEEE Trans. Inf. Forensics Secur. **5**(1), 180–187 (2010)
8. Bianchi, T., Piva, A., Barni, M.: Encrypted domain DCT based on homomorphic cryptosystems. EURASIP J. Inf. Secur. **2009**(1), 716357 (2009)
9. Blakley, G.R.: Safeguarding cryptographic keys. In: Proceedings of the National Computer Conference 1979, vol. 48, pp. 313–317 (1979)
10. Brakerski, Z., Vaikuntanathan, V.: Efficient fully homomorphic encryption from (standard) LWE. In: 2011 IEEE 52nd Annual Symposium on Foundations of Computer Science, pp. 97–106, October 2011
11. Brakerski, Z.: Fully homomorphic encryption without modulus switching from classical GapSVP. In: Safavi-Naini, R., Canetti, R. (eds.) CRYPTO 2012. LNCS, vol. 7417, pp. 868–886. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-32009-5_50
12. Buades, A., Coll, B., Morel, J.M.: A review of image denoising algorithms, with a new one. Multiscale Model. Simul. **4**(2), 490–530 (2005)
13. Buades, A., Coll, B., Morel, J.M.: A non-local algorithm for image denoising. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005. CVPR 2005, vol. 2, pp. 60–65. IEEE (2005)
14. Chan, S.H., Zickler, T., Lu, Y.M.: Monte carlo non-local means: random sampling for large-scale image filtering. IEEE Trans. Image Process. **23**(8), 3711–3725 (2014)
15. Check, H.E.: Cloud cover protects gene data. Nature **519**(7544), 400 (2015)
16. Cramer, R., Damgård, I., Maurer, U.: General secure multi-party computation from any linear secret-sharing scheme. In: Preneel, B. (ed.) EUROCRYPT 2000. LNCS, vol. 1807, pp. 316–334. Springer, Heidelberg (2000). https://doi.org/10.1007/3-540-45539-6_22
17. van Dijk, M., Gentry, C., Halevi, S., Vaikuntanathan, V.: Fully homomorphic encryption over the integers. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 24–43. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-13190-5_2
18. Elgamal, T.: A public key cryptosystem and a signature scheme based on discrete logarithms. IEEE Trans. Inf. Theory **31**(4), 469–472 (1985)
19. Feldman, P.: A practical scheme for non-interactive verifiable secret sharing. In: 28th Annual Symposium on Foundations of Computer Science (SFCS 1987), pp. 427–438, October 1987
20. Gentry, C.: A fully homomorphic encryption scheme. Ph.D. thesis, Stanford University (2009)
21. Goldreich, O., Micali, S., Wigderson, A.: How to play any mental game. In: Proceedings of the Nineteenth Annual ACM Symposium on Theory of Computing, STOC 1987, pp. 218–229 (1987)
22. Hirt, M., Nielsen, J.B.: Robust multiparty computation with linear communication complexity. In: Dwork, C. (ed.) CRYPTO 2006. LNCS, vol. 4117, pp. 463–482. Springer, Heidelberg (2006). https://doi.org/10.1007/11818175_28

23. Hu, X., Zhang, W., Hu, H., Yu, N.: Non-local denoising in encrypted images. In: Hsu, R.C.-H., Wang, S. (eds.) IOV 2014. LNCS, vol. 8662, pp. 386–395. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-11167-4_38
24. Hu, X., Zhang, W., Li, K., Hu, H., Yu, N.: Secure nonlocal denoising in outsourced images. ACM Trans. Multimedia Comput. Commun. Appl. **12**(3), 40:1–40:23 (2016)
25. Johnson, W.B., Lindenstrauss, J.: Extensions of lipschitz mappings into a hilbert space. Contemp. Math. **26**(189–206), 1 (1984)
26. Kenthapadi, K., Korolova, A., Mironov, I., Mishra, N.: Privacy via the johnson-lindenstrauss transform. arXiv preprint arXiv:1204.2606 (2012)
27. Lathey, A., Atrey, P.K.: Image enhancement in encrypted domain over cloud. ACM Trans. Multimedia Comput. Commun. Appl. **11**(3), 38:1–38:24 (2015)
28. Mohanty, M., Atrey, P., Ooi, W.T.: Secure cloud-based medical data visualization. In: Proceedings of the 20th ACM International Conference on Multimedia, MM 2012, pp. 1105–1108 (2012)
29. Nuida, K., Kurosawa, K.: (Batch) Fully homomorphic encryption over integers for non-binary message spaces. In: Oswald, E., Fischlin, M. (eds.) EUROCRYPT 2015. LNCS, vol. 9056, pp. 537–555. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-46800-5_21
30. Paillier, P.: Public-key cryptosystems based on composite degree residuosity classes. In: Stern, J. (ed.) EUROCRYPT 1999. LNCS, vol. 1592, pp. 223–238. Springer, Heidelberg (1999). https://doi.org/10.1007/3-540-48910-X_16
31. Pedrouzo-Ulloa, A., Troncoso-Pastoriza, J.R., Prez-Gonzlez, F.: Image denoising in the encrypted domain. In: 2016 IEEE International Workshop on Information Forensics and Security (WIFS), pp. 1–6, December 2016
32. Rivest, R.L., Adleman, L., Dertouzos, M.L.: On data banks and privacy homomorphisms. Found. Secur. Comput. **4**(11), 169–180 (1978)
33. SaghaianNejadEsfahani, S.M., Luo, Y., c. S. Cheung, S.: Privacy protected image denoising with secret shares. In: 2012 19th IEEE International Conference on Image Processing, pp. 253–256, September 2012
34. Shamir, A.: How to share a secret. Commun. ACM **22**(11), 612–613 (1979)
35. Talebi, H., Milanfar, P.: Global image denoising. IEEE Trans. Image Process. **23**(2), 755–768 (2014)
36. Troncoso-Pastoriza, J.R., Prez-Gonzlez, F.: Fully homomorphic faces. In: 2012 19th IEEE International Conference on Image Processing, pp. 2657–2660, September 2012
37. Yao, A.C.: Protocols for secure computations. In: 23rd Annual Symposium on Foundations of Computer Science (SFCS 1982), pp. 160–164, November 1982
38. Zheng, P., Huang, J.: Discrete wavelet transform and data expansion reduction in homomorphic encrypted domain. IEEE Trans. Image Process. **22**(6), 2455–2468 (2013)
39. Zheng, P., Huang, J.: Implementation of the discrete wavelet transform and multiresolution analysis in the encrypted domain. In: Proceedings of the 19th ACM International Conference on Multimedia, MM 2011, pp. 413–422 (2011)
40. Zheng, P., Huang, J.: Walsh-hadamard transform in the homomorphic encrypted domain and its application in image watermarking. In: Kirchner, M., Ghosal, D. (eds.) IH 2012. LNCS, vol. 7692, pp. 240–254. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-36373-3_16
41. Zheng, Y., Cui, H., Wang, C., Zhou, J.: Privacy-preserving image denoising from external cloud databases. IEEE Trans. Inf. Forensics Secur. **12**(6), 1285–1298 (2017)