Hindawi
Mathematical Problems in Engineering
Volume 2019, Article ID 6410626, 7 pages
https://doi.org/10.1155/2019/6410626

Research Article

Outsourcing Computing of Large Matrix Jordan Decomposition

Hongfeng Wu and Jingjing Yan

College of Science, North China University of Technology, Beijing 100144, China

Correspondence should be addressed to Hongfeng Wu; whfmath@gmail.com

Received 28 March 2019; Revised 29 June 2019; Accepted 30 July 2019; Published 19 August 2019

Academic Editor: Elio Masciari

The Jordan decomposition of matrix is a typical scientific and engineering computational task, but such computation involves enormous computing resources for large matrices, which is burdensome for the resource-limited clients. Cloud computing enables computational resource-limited clients to economically outsource such problems to the cloud server. However, outsourcing Jordan decomposition of large-scale matrix to the cloud brings great security concerns and challenges since the matrices usually contain sensitive information. In this paper, we present a secure, verifiable, efficient, and privacy preserving algorithm for outsourcing Jordan decomposition of large-scale matrix. Security analysis shows that our algorithm is practically secure. Efficient verification algorithm is used to verify the results returned from the cloud.

1. Introduction

With the rapid development of information technology in recent years, cloud outsourcing computing has attracted more and more attention from the network industry [1, 2]. Compared with the traditional computing mode, cloud computing greatly saves computing time and cost for users. Therefore, cloud computing is favored by many users.

However, while outsourcing cloud computing brings many benefits to cloud users, there are also many security threats [3]. Since users usually have no way to know whether the cloud is honest or not, how to ensure the security of users' privacy information in outsourcing computing has become the primary issue. When the user outsources the computing task to the cloud for computing, the privacy information contained in the original computing task may be known by the cloud, so the security of the user's privacy information cannot be guaranteed. Therefore, when users outsource computing tasks to the cloud, they need to encrypt the privacy information contained in the original computing tasks so as to ensure the security of users' privacy data. How does this protocol ensure that the results returned by the cloud to users are correct? For example, in order to reduce the cost and gain more benefits, the cloud returns an incorrect result to the user. At this time, in order to ensure the user's own interests, the user needs to verify the result.

On the basis of solving the above two security threats, we should ensure the effectiveness of outsourcing computing. It takes much more time for users to calculate by themselves than by outsourcing, so outsourcing computing is chosen to improve the calculation efficiency.

In recent years, many scholars have proposed outsourcing computing protocols for different aspects of matrix computing. In 1998, Atallah et al. [4] proposed an outsourcing computing protocol for large-scale matrix multiplication, but did not realize the verifiability of the results. In 2008, Benjamin and Atallah [5] used homomorphism encryption technology and realized the verifiability of the outsourced computing protocol results, but the complexity of the encryption process was high. In 2009, Gentry [6] proposed a solution to large-scale linear equations, but its operating cost was too high. In 2017, Yang and Wu [7] also proposed a large-scale matrix multiplication. However, based on the multiplication problem of nonsquare matrix, the idea of classification discussion was adopted, which increased the complexity of outsourcing calculation. In the same year, Zhou and Li [8] proposed an outsourcing computing protocol for large-scale matrix decomposition, including eigenvalue decomposition of matrix, SVD decomposition, and so on, but there was no outsourcing computing protocol involving matrix Jordan decomposition. Jordan decomposition has been widely used

in matrix equation theory, ordinary differential equation, modern cybernetics, and other aspects [9]. At the same time, based on the large-scale matrix, Jordan decomposition requires a lot of material and financial resources.

In this paper, we propose an outsourcing computing protocol based on large-scale matrix Jordan decomposition, which realizes the verifiability and efficiency of the results on the basis of ensuring the security.

This paper is mainly divided into the following parts. In Section 2, we introduce the relevant background knowledge. In Section 3, we introduce the system model. In Section 4, we design an outsourcing computing protocol based on Jordan decomposition of large-scale matrix. In Section 5, we analyze the outsourcing computing protocol proposed in this paper. Finally, the paper is summarized.

2. Background Knowledge

2.1. Jordan Decomposition. Let A be a real symmetric $n \times n$ matrix, λ a real number, and μ a nonzero vector such that

$$A\mu = \lambda\mu, \tag{1}$$

then λ is the eigenvalue of matrix A and μ is the eigenvector corresponding to the eigenvalue λ.

Given a general $n \times n$ matrix A, Jordan decomposition can be used to decompose it as follows:

$$A = (T_1, T_2, \ldots, T_k)\begin{pmatrix} J_{n_1}(\lambda_1) & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & J_{n_k}(\lambda_k) \end{pmatrix}$$
$$(T_1, T_2, \ldots, T_k)^{-1} = TJT^{-1}, \quad n_1 + n_2 + \ldots + n_k = n, \tag{2}$$

where T_i is an invertible $n \times n_i$ matrix and $J_{n_i}(\lambda_i)$ is called the Jordan block, $i = 1, 2, \cdots, k$. We use T to denote the matrix $(T_1, T_2, \cdots, T_k)$ and use J to denote the matrix $\begin{pmatrix} J_{n_1}(\lambda_1) & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & J_{n_k}(\lambda_k) \end{pmatrix}$. The Jordan matrix J is called the Jordan canonical form of matrix A, and its diagonal elements $\lambda_1, \lambda_2, \cdots, \lambda_k$ are the eigenvalues of the matrix A.

If the cloud does not know the eigenvalues of the matrix, then the cloud will not know the Jordan canonical form of the matrix. T is called the transformation matrix.

Assuming that λ is the eigenvalue of matrix A and μ is the eigenvector corresponding to λ, then

$$A\mu = \lambda\mu. \tag{3}$$

For any $n \times n$ matrix A, carry out the following transformation:

$$\begin{pmatrix} A \\ E \end{pmatrix} \xrightarrow{\text{Row and column reciprocal elementary transformation}} \begin{pmatrix} J \\ T \end{pmatrix}, \tag{4}$$

where E is the identity matrix, from which we can get the Jordan canonical form J and the transformation matrix J of matrix A [10–13], so if the cloud does not know A and J, it will not get T.

2.2. Permutations and δ Function. Using Cauchy representation, we map the set S to itself and express the permutation as follows:

$$\begin{pmatrix} s_1 & \cdots & s_n \\ s_1' & \cdots & s_n' \end{pmatrix}. \tag{5}$$

In this representation, the first row is the preimage of the mapping, and the second row is the image under the mapping. In equation (5), we can represent permutation as a bijective function $\psi(s_i)$ whose range and domain are set as set S.

The function δ_{xy} is defined as follows:

$$\delta_{xy} = \begin{cases} 1, & x = y, \\ 0, & x \neq y. \end{cases} \tag{6}$$

We randomly pick a bunch of numbers $\eta_1, \ldots, \eta_n$, and each of these numbers is randomly picked from space $\Omega = \{-1, 1\}$. Then, the encrypted matrix P can be defined as

$$P(i, j) = \eta_i \delta_{\psi(i), j}, \quad 1 \leq i, j \leq n, \tag{7}$$

where both the range and the definition domain of bijective function $\psi(i)$ are $S = \{1, 2, \ldots, n\}$. It can be seen from the definition that in the encrypted matrix P, each row and each column have only one nonzero element, and the encrypted matrix P is an orthonormal matrix.

3. System Model

In outsourcing computing, security computing models can be divided into two categories according to the differences between adversaries: semihonest model and malicious model [14]. In the semihonest model, although the cloud complies with the protocol, it will passively attempt to obtain the privacy information in the user's original computing task [14]. In the malicious model, the cloud will not only proactively obtain privacy information from the client but also violate the protocol and arbitrarily return an error result to the client, unwilling to be detected by the client. Therefore, in the malicious cloud model, users must be able to verify the results returned by the cloud to resist cloud spoofing. In this article, we consider the malicious model.

The matrix Jordan decomposition outsourcing model designed by us is shown in Figure 1. Firstly, the client generates the key K, uses K to encrypt the original matrix A, protects the privacy information contained in the original matrix A, and obtains the encrypted matrix B. Secondly, the client sends the encrypted matrix B to the cloud and uses the cloud's powerful computing power and huge storage capacity to decompose the encrypted matrix B with Jordan decomposition. Then, the cloud returns the Jordan decomposition result of the encrypted matrix B to the client. After receiving the result returned from the cloud, the client verifies the result. If the result is correct, the client decrypts it to get the Jordan decomposition result of the original matrix A. Otherwise, the client will deny the result returned from the cloud and let the cloud calculate again.

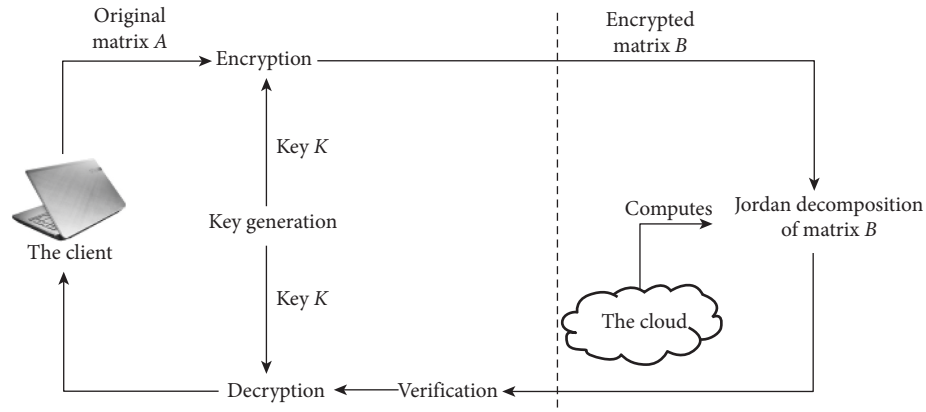

Figure 1: An outsourcing system model for Jordan decomposition of a large matrix.

Our large-scale matrix Jordan decomposition outsourcing protocol needs to achieve the following objectives [15]:

(1) Security: no user's private information can be obtained by the cloud.

(2) Verifiability: the client can verify the correctness of the result with a high probability and can also find the wrong result from the cloud with a nonnegligible probability.

(3) Validity: compared with Jordan decomposition of the matrix calculated by the client itself, the client can greatly reduce the local computation through outsourcing.

Our system model is divided into five parts, as shown in Figure 1:

(1) Key generation: the client randomly generates a key K to encrypt the original matrix A.

(2) Encryption: the client uses the key K to encrypt the original matrix A. In the process of encrypting the original matrix A into the encrypted matrix B, we use matrix multiplication and linear mapping.

(3) Cloud computing: the cloud decomposes the encrypted matrix B with Jordan decomposition and returns the results to the client.

(4) Verification: the client verifies the results returned by the cloud. If the result is correct, the client accepts the result. Otherwise, the client will reject the returned result and ask the cloud to compute again.

(5) Decryption: if the cloud returns the correct result, the client decrypts the Jordan canonical form, the transformation matrix, and the inverse of the transformation matrix of matrix B into the Jordan standard form, the transformation matrix, and the inverse of the transformation matrix of matrix A. Then, the Jordan decomposition result of the original matrix A is obtained.

4. Protocol Design

4.1. Key Generation. In this paper, we use the encrypted matrix to generate the key, and the process of generating the encrypted matrix is as follows:

$$Q(i, j) = q_i \delta_{\psi(i), j}, \quad 1 \le i, j \le n, \tag{8}$$

where q_i is the random number generated by space Ω.The definition of space Ω we have described in the background knowledge, $\psi(i)$, is a bijective function.

4.2. Encryption. In the Jordan decomposition outsourcing of matrix, the user's purpose is to get the Jordan standard form, transformation matrix, and the inverse of transformation matrix of matrix A. At the same time, the user wants matrix A, matrix A's Jordan standard form, and transformation matrix not to be exposed in the cloud. In this paper, we will encrypt the user's privacy information, that is, we will encrypt the original matrix A and the Jordan standard form and transformation matrix of the original matrix A. As can be seen from Section 4.1, we only need to encrypt the original matrix A and its eigenvalues and eigenvectors.

(i) In the matrix Jordan decomposition outsourcing process, if the user directly sends original matrix A to the cloud, the privacy information in matrix A may be exposed to the cloud. In order not to disclose the privacy information of A, the client first selects α randomly from the real number set R and then encrypts the original matrix A as follows:

$$A' = \alpha A. \tag{9}$$

According to the above encryption method, we can know that the eigenvectors of matrix A' and matrix A are the same, and the relationship between the

eigenvalue λ' of A' and the eigenvalue λ of A is as follows:

$$\lambda' = \alpha\lambda. \tag{10}$$

The above formula is proved as follows: suppose λ is an eigenvalue of matrix A and x is the eigenvector corresponding to the eigenvalue λ, then

$$A'x = \alpha Ax = \alpha\lambda x = \lambda' x. \tag{11}$$

End of proof.

Since the cloud does not know the exact value of α, if the client sends A' to the cloud, then the privacy information contained in matrix A itself and its eigenvalues are protected. However, matrix A and matrix A' have the same eigenvector. In other words, the privacy information contained in the eigenvector of matrix A is not protected. So, we are also going to encrypt the eigenvectors of A.

(ii) Now, we encrypt the eigenvectors of matrix A. First, let us say

$$x = Q^T y \ (Q \text{ is an orthonormal matrix}). \tag{12}$$

Then, equation (11) can be expressed as

$$A'Q^T y = \lambda' Q^T y. \tag{13}$$

Multiply both sides of equation (13) by matrix Q:

$$QA'Q^T y = \lambda' QQ^T y = \lambda' y. \tag{14}$$

Therefore, we can write equation (13) as follows:

$$By = \lambda' y, \tag{15}$$

where $B = QA'Q^T$. At this point, the original matrix A is transformed into the encrypted matrix B through encryption, and their relation can be expressed as $B = QA'Q^T = Q\alpha AQ^T = \alpha QAQ^T$. According to equation (12), the encrypted matrix B hides the privacy information contained in the eigenvector x of matrix A.

4.3. Cloud Computing. Through the above encryption, we have completed the protection of the user's privacy information. At this point, we send the encrypted matrix B to the cloud, which decomposes it with Jordan decomposition and returns the decomposed result to the client. Then, the client gets the Jordan standard form, the transformation matrix, and the inverse of the transformation matrix of the encrypted matrix B, which are

The Jordan standard form J' of matrix B is $J' = \begin{pmatrix} J'_{n_1}(\lambda_1) & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & J'_{n_k}(\lambda_k) \end{pmatrix}$, the transformation matrix T' of matrix B is $T' = (T'_1, T'_2, \ldots, T'_k)$, and the inverse of the transformation matrix T'^{-1} of matrix B is $T'^{-1} = (T'_1, T'_2, \ldots, T'_k)^{-1}$, where $J'_{n_i}(\lambda_i)$, $i = 1, 2, \ldots, k$ is a Jordan block, J' is a Jordan matrix, and the diagonal elements of J' are the eigenvalues of matrix B; T'_i, $i = 1, 2, \ldots, k$ is an $n \times n_i$ matrix, and T' is an invertible transformation matrix.

In the encryption process, we use the encrypted matrix Q to ensure that as long as the cloud returns an orthonormal eigenvector, the client will also get an orthonormal eigenvector after decryption. The proof is as follows:

Suppose y_1 and y_2 are the orthonormal eigenvectors of matrix B returned from the cloud, and the client decrypts them, and the obtained two eigenvectors satisfy $x_1 = Q^T y_1$ and $x_2 = Q^T y_2$, then we can get $x_1^T x_1 = y_1^T QQ^T y_1 = y_1^T y_1 = 1$ and $x_2^T x_2 = y_2^T QQ^T y_2 = y_2^T y_2 = 1$, and because of $x_1^T x_2 = y_1^T QQ^T y_2 = y_1^T y_2 = 0$, x_1 and x_2 are also orthonormal.

4.4. Verification. For the results returned to the client from the cloud, it is necessary for the client to verify the results. To ensure that the results T', J', and T'^{-1} are correct, the user first checks whether J' is a Jordan matrix. If J' is not a Jordan matrix, the user thinks the result is incorrect. If J' is a Jordan matrix, then the user verifies whether the following formula is correct:

$$B = (T'_1, T'_2, \ldots, T'_k)\begin{pmatrix} J'_{n_1}(\lambda_1) & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & J'_{n_k}(\lambda_k) \end{pmatrix} (T'_1, T'_2, \ldots, T'_k)^{-1} = T'J'T'^{-1}, \quad n_1 + n_2 + \ldots + n_k = n. \tag{16}$$

This is equivalent to verifying whether the following formula is true:

$$J'T'^{-1} = T'^{-1}B. \tag{17}$$

The verification method is as follows:

(1) Choose a random $n \times 1$ vector u; each element in u is selected randomly from the set $\{0, 1\}$.

(2) The user computes $v_1 = J'(T'^{-1}u)$ and $v_2 = T'^{-1}Bu$.

(3) Repeat steps 1 and 2 for a total of l rounds of tests. In the l rounds of tests, if $v_1 = v_2$ is constant, the client will accept the result. Otherwise, the client judges that the result is wrong and does not accept the result.

4.5. Decryption. If the client verifies that the result returned by the cloud is correct, then the client will decrypt the result and convert it into the Jordan standard form, the transformation matrix, and the inverse of the transformation matrix of matrix A. The decryption process is as follows:

$$\begin{aligned} T_i &= Q^T T_i', \\ J_{n_i}(\lambda_i) &= \frac{J'_{n_i}(\lambda_i)}{\alpha}, \\ T_i^{-1} &= \left(Q^T T_i'\right)^{-1}, \\ i &= 1, 2, \ldots, k. \end{aligned} \tag{18}$$

Through equation (18), the Jordan standard form J, the transformation matrix T, and the inverse of the transformation matrix T^{-1} of the original matrix A can be obtained as

$$\begin{aligned} T &= Q^T T', \\ J &= \frac{J'}{\alpha}, \\ T^{-1} &= \left(Q^T T'\right)^{-1}. \end{aligned} \tag{19}$$

Then, we can get

$$\begin{aligned} A &= TJT^{-1} = Q^T T' \frac{J'}{\alpha} \left(Q^T T'\right)^{-1}, \\ \Rightarrow A &= \left(Q^T T_1', \ldots, Q^T T_k'\right) \begin{pmatrix} \frac{J'_{n_1}(\lambda_1)}{\alpha} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \frac{J'_{n_k}(\lambda_k)}{\alpha} \end{pmatrix} \\ &\quad \cdot \left(Q^T T_1', \ldots, Q^T T_k'\right)^{-1}. \end{aligned} \tag{20}$$

Then, the client gets the Jordan decomposition of the original matrix A.

5. Protocol Analysis

5.1. Security Analysis. The original matrix A is encrypted as a matrix $B = \alpha Q A Q^T$, where Q is an encrypted matrix and α is a random number generated by the set of real numbers R. Firstly, the original matrix A is encrypted as $A' = \alpha A$. Assuming that the cloud knows matrix A', if the cloud wants to use violent means to obtain matrix A through A', it also needs to guess $|R|^2$ times, where $|R|$ represents the number of elements in the real number set R. Then, the encrypted matrix Q is used to encrypt matrix A'; that is, $B = QA'Q^T$. This encryption causes the elements in matrix A' to be rearranged in the following form:

$$B = QA'Q^T = \begin{pmatrix} \frac{q_1}{q_1} A'_{\pi(1),\pi(1)} & \cdots & \frac{q_1}{q_n} A'_{\pi(1),\pi(n)} \\ \vdots & \ddots & \vdots \\ \frac{q_i}{q_1} A'_{\pi(1),\pi(1)} & \cdots & \frac{q_i}{q_n} A'_{\pi(i),\pi(n)} \\ \vdots & \ddots & \vdots \\ \frac{q_n}{q_1} A'_{\pi(1),\pi(1)} & \cdots & \frac{q_n}{q_n} A'_{\pi(n),\pi(n)} \end{pmatrix}. \tag{21}$$

According to equation (21), the elements of A' are rearranged by the function π and narrowed by $q_i\,(1 \le i \le n)$, where π has $n!$ possibilities and q_i has two possibilities. At this time, even if the cloud wants to get matrix A' from matrix B, its probability is only $1/(2^n \cdot n!)$, so when n is large, its probability is negligible.

After the user sends the encrypted matrix B to the cloud, the cloud decomposes matrix B with Jordan decomposition, so the cloud knows the eigenvalue λ' and the eigenvector y of matrix B. The following analysis shows how the protocol protects the eigenvalue λ and eigenvector x of matrix A.

The relation between the eigenvalue of matrix A and matrix B is equation (10). If the cloud does not know the random real number α, it will not use λ' to get λ.

The relation between the eigenvector of matrix A and matrix B is equation (12). Obviously, the encrypted matrix Q protects the eigenvector x well. According to the previous discussion, the probability of the cloud to get matrix Q is $1/(2^n \cdot n!)$, so when n is large, the cloud has no way to decrypt x from the eigenvector y.

Based on the above analysis, we believe that users' privacy information is well protected.

5.2. Result Verifiability Analysis. In this section, we demonstrate that our verification algorithm is highly resistant to cloud spoofing. The certification process is as follows.

First, we prove that any correct results returned from cloud computing can be successfully verified. If the cloud returns a correct result, then $T'J'T'^{-1}$ is equal to matrix B, so in each round of check, no matter what the value of u is, there is $v_1 = v_2$; that is, any correct result can be verified.

Second, we prove that the probability of any incorrect results returned from the cloud being verified is negligible. If the returned matrix J' is not a Jordan matrix, the client directly assumes that the result is wrong and therefore cannot receive the wrong result. When J' is a Jordan matrix, we show that our protocol has a high probability of resisting false results. The error analysis of our verification algorithm shows that the probability of error result passing verification is less than $1/2^l$ [16]; that is, the probability of error result

passing verification decreases exponentially with the increase of verification times (l represents the verification times). Therefore, if the appropriate l value is selected, any error results returned from the cloud are unlikely to pass verification.

5.3. Effectiveness Analysis. Compared with directly computing the matrix Jordan decomposition, users can reduce the computational burden by outsourcing calculation. Our system model requires the client to perform four algorithms: key generation, encryption, verification, and decryption. The computational complexity of the four algorithms is analyzed in the following sections to verify the effectiveness of the outsourcing protocol.

5.3.1. Key Generation. Generating the encrypted matrix Q is all the tasks of the algorithm, and its algorithm complexity is $O(n)$.

5.3.2. Encryption Algorithm. We use encryption algorithm to encrypt the original matrix A into matrix B, and $B = QA'Q^T = Q\alpha AQ^T = \alpha QAQ^T$. In this encryption process, the multiplication of the encrypted matrix Q and the original matrix A is the most time-consuming operation. According to equation (21), its algorithm complexity is $O(n^2)$.

5.3.3. Verification Algorithm. In each round of random detection, the client has to calculate $J'(T'^{-1}u)$ and $T'^{-1}Bu$, and matrix and vector multiplication is the most complex calculation operation, and its algorithm complexity is $O(n^2)$. Because the verification algorithm requires l rounds of random detection, the overall computational complexity of the verification algorithm is $O(\ln^2)$. Compared with the large matrix A, l is far less than n, that is, $l \ll n$, so the computational complexity of the verification algorithm is $O(n^2)$.

5.3.4. Decryption Algorithm. If the results T', J', and T'^{-1} returned by the cloud are correct, the client decrypts them and gets T, J, and T^{-1} of the original matrix A. The decryption algorithm is as follows:

$$J = \frac{J'}{\alpha},$$
$$T = Q^T T', \tag{22}$$
$$T^{-1} = \left(Q^T T'\right)^{-1}.$$

The most time-consuming operation is to calculate $Q^T T'$, whose computational complexity is $O(n^2)$.

To sum up, the computational complexity of all the calculations that need to be performed by the client is $O(n^2)$. In contrast, the computational complexity of directly computing the matrix Jordan decomposition is $O(n^3)$. If n is big enough, there will be a big difference between $O(n^2)$ and $O(n^3)$. Therefore, users can save a lot of computing time and cost by outsourcing computing, thus achieving the efficiency.

6. Conclusion

In this paper, we design a secure, verifiable, efficient large-scale matrix Jordan decomposition outsourcing protocol. We use efficient encryption technology to ensure the security of users' privacy information. At the same time, we design an efficient verification algorithm to ensure that the client can effectively verify whether the cloud returns the correct results. Matrix Jordan decomposition has a wide range of applications in the field of science, which we will further study.

Data Availability

The data of this article are computed by the algebra method, which are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The first author was supported by the Yuyou Team Support Program of Northern University of Technology (107051360019XN137/007).

References

[1] W. Liu, "Research on the characteristics and key application fields of cloud computing," *Intelligence*, vol. 30, 2014.

[2] C. Wang, K. Ren, and J. Wang, "Secure and practical outsourcing of linear programming in cloud computing," in *Proceedings of the 2011 Proceedings IEEE INFOCOM*, pp. 820–828, Shanghai, China, April 2011.

[3] D.-G. Feng, M. Zhang, Y. Zhang, and Z. Xu, "Study on cloud computing security," *Journal of Software*, vol. 22, no. 1, pp. 71–83, 2011.

[4] M. J. Atallah, K. N. Pantazopoulos, J. R. Rice, and E. H. Spafford, "Secure outsourcing of scientific computations," *Advances in Computers*, vol. 54, pp. 216–272, 2002.

[5] D. Benjamin and M. J. Atallah, "Private and cheating-free outsourcing of algebraic computation," in *Proceedings of the Sixth Annual Conference on Privacy, Security and Trust*, pp. 240–245, IEEE, Fredericton, Canada, October 2008.

[6] C. Gentry, "Fully homomorphic encryption using ideal lattices," in *Proceedings of the 41st Annual ACM Symposium on Symposium on Theory of Computing—STOC '09*, vol. 9, no. 4, pp. 169–178, Bethesda, MA, USA, May 2009.

[7] B. Yang and D. Wu, "The efficient and verifiable secure outsourcing calculation of matrix product," *Journal of Cryptography*, vol. 4, no. 4, pp. 322–332, 2017.

[8] L. Zhou and C. Li, "Outsourcing eigen-decomposition and singular value decomposition of large matrix to a public cloud," *IEEE Access*, vol. 4, pp. 869–879, 2017.

[9] Y. Zhao, "Discussion on the role of matrix Jordan canonical form in matrix analysis," *Journal of Western Yunnan Normal University of Science and Technology*, vol. 1, pp. 119–124, 2015.
[10] L. Chen, R. Hou, and L. Wang, "Jordan canonical forms of matrices over quaternion field," *Applied Mathematics and Mechanics*, vol. 17, no. 6, pp. 559–568, 1996.
[11] L. Huang, "Jordan canonical form of a matrix over the quaternion field," *Northeastern Mathematical Journal*, vol. 10, no. 1, pp. 18–24, 1994.
[12] J. Tongsong and C. Li, "Generalized diagonalization of matrices over quaternion field," *Applied Mathematics and Mechanics*, vol. 20, no. 11, pp. 1297–1304, 1999.
[13] T. Jiang and W. Zhuang, "A simple proof of Jordan canonical form for matrices over the quaternion field, (Chinese)," *Mathematica Applicata*, vol. 14, pp. 204–207, 2001.
[14] S. Guo, *Research on Outsourcing Computing in Cloud Computing*, University of the Chinese Academy of Sciences, Beijing, China, 2013.
[15] S. Duncan, "Verifiable computation of security outsourcing matrix and its application," *Chinese Science: Information Science*, vol. 43, no. 7, pp. 842–852, 2013.
[16] R. Motwani and P. Raghavan, *Randomized Algorithms*, Cambridge University Press, Cambridge, UK, 1995.