Information Sciences 527 (2020) 548–559

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Cloud-assisted privacy-conscious large-scale Markowitz portfolio

Yushu Zhang [a,b], Jin Jiang [a], Yong Xiang [b,*], Ye Zhu [b], Liangtian Wan [c], Xiyuan Xie [d]

[a] College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, China
[b] School of Information Technology, Deakin University, Victoria 3125, Australia
[c] School of Software, Dalian University of Technology, Dalian 116620, China
[d] School of Electronics and Information Engineering, Southwest University, Chongqing 400715, China

ARTICLE INFO

Article history:
Received 13 March 2018
Revised 29 October 2018
Accepted 24 December 2018
Available online 27 December 2018

Keywords:
Privacy computing
Cloud computing
Computation outsourcing
Markowitz mean-variance model
Portfolio

ABSTRACT

The theory of Markowitz portfolio has had enormous value and extensive applications in finance since it came into being. With the advent of the Big-Data era and the increasingly complicated financial market, the resource consumption of computing portfolio investments is significantly increasing. Cloud computing offers a good platform to efficiently compute large-scale portfolio investments, in particular, for resource-limited investors. In this paper, a Markowitz model (MM) is taken into consideration for outsourcing to a public cloud in a privacy-conscious way. As in general computation outsourcing, outsourcing MM inevitably faces four issues, namely, input/output privacy, correctness, verification, and substantial computation gain for investors; it has consistent complexity with the original methods when the cloud solves the encrypted version. However, the proposed cloud-assisted privacy-conscious MM employs location-scrambling and value-alteration encryption operations, which can protect the MM's input/output privacy well. Moreover, the correctness of solving MM over an encrypted domain in the cloud side can be demonstrated and the results returned by the cloud can be verified. Furthermore, both theoretical and experimental analyses validate that the investor can obtain a huge amount of computational gain, and the cloud complexity consistent with that of the original case when solving the encrypted version.

1. Introduction

When portfolio theory was proposed by Markowitz in the early 1950s [26], finance entered into the stage of quantificational analysis, which can be deemed as the beginning of financial mathematics. In that work, the Markowitz mean-variance model (MM) was put forward for the risk portfolio investment. Its core idea is to minimize the risk of portfolio investment by diversifying the investment to offset the risk of the portfolio without dropping the expected return. The MM is very valuable and widely used in the finance field. The MM has some defects, like the hypothesis that the investor's decision is merely based on the risks and benefits of assets, and subsequent works [15,31,50] were presented to fix the defects. In spite of this, the basic framework of MM remains unchanged. To date, MM has been a classical and popular model to measure the risk and benefit of a portfolio and balance them for asset allocation.

* Corresponding author.
E-mail address: yxiang@deakin.edu.au (Y. Xiang).

https://doi.org/10.1016/j.ins.2018.12.055
0020-0255/

With the advent of the Big-Data era [3,39] and the increasing number of investors pouring into the market, investors are always seeking a strategy to efficiently control the risk of asset investment. MM can be an alternative strategy for these investors. However, the expansion and spread of big data increases the scale of MM and the number of portfolios. In addition, the computational task of MM itself is especially complex. Resource constraints are a major problem for investors. Fortunately, the rapid development of cloud computing presents a good alternative for investors, as the cloud computing has vast computation resources, huge storage capacity, high cost efficiency, and convenient service mode. The resource-constrained investors can break the limit of the local computing resources by outsourcing their MMs to commercial public clouds.

However, enjoying these enormous benefits is not an easy task. The investor loses command of the model processing after outsourcing the model to a public cloud, and thus, handing the MM over to a public cloud also faces four inevitable issues, as in the case of general computation outsourcing [4,6,12,36]. The first issue is the input/output privacy. On the one hand, MM is not expected to be directly handled by the cloud, as the sensitive information of anticipated return rates will be exposed to the cloud. As a result, an encryption operation must be added by the investor for input privacy prior to outsourcing. On the other hand, after the cloud fulfills the computation of the encrypted MM, the results reflect the portfolio information; thus, the output privacy must be ensured. The second issue is correctness. Differing from the computation of the original MM, the computation of the encrypted MM should guarantee that the computation results can be correctly decrypted by the investor. Otherwise, the outsourcing would be meaningless. The third issue is verification. The cloud is a third-party platform in which the internal staff may bear malicious intent for the computation or arbitrarily fabricate a result for resource savings. In addition, the returned result may suffer from some occasional bugs, hardware faults, or even external disturbances. Thus, verification is essential. The last issue is the complexity. Apparently, the local computational complexity in the investor's hand should be greatly decreased so that it is much smaller than the complexity of the original MM, which is the initial intent of outsourcing. Currently, the computational complexity between the original MM and the encrypted version can remain consistent.

This study focuses on the outsourcing of MM, which can successfully address the above-mentioned four issues simultaneously. An MM computation problem is partitioned into two subproblems: a covariance matrix (CM) calculation problem and a quadratic programming (QP) problem, which are outsourced successively to a public cloud. This is because the CM contained in MM is calculated via the sample return-rate matrix in the investments, but this calculation has very high computational complexity and consumes more resources if performed by the investor locally. In order to protect the input privacy, the sample return matrix and the QP are encrypted by some operations, including location scrambling and value alteration, before being sent to the cloud. Meanwhile, after fulfilling the calculation of CM and the solving of QP, the cloud has no knowledge of the output privacy without these operations. Although the cloud exploits the calculation of CM and the solving of QP over the encrypted domain, the correctness can still be ensured. Moreover, the proposed outsourcing can effectively verify the results if the cloud has abnormal behaviors. Additionally, the experimental analysis demonstrated that the investor is able to save large amounts of computing resources by outsourcing the MM task to the public cloud, whereas the proposed encryption scheme presents no substantial overhead in the cloud.

The main contributions of this study are summarized as follows.

- We develop a cloud-assisted privacy-conscious MM for the first time, which simultaneously addresses the four issues of input/output privacy, correctness, verification, and efficiency.
- The original MM problem is divided into two subproblems, namely, a CM calculation problem and a QP solving problem, which are separately and efficiently outsourced to the public cloud.
- The proposed encryption design not only protects the input/output privacy, but also ensures the correctness of the calculation of CM and the solving of QP over the encrypted domain.
- The proposed outsourcing helps the investor economically produce large-scale Markowitz portfolios because of the substantial local computation savings without introducing extra overhead in the cloud.

The rest of this paper proceeds as follows. The next section provides an overview of the related works. Section 3 is about the problem formulation, consisting of the background of the Markowitz mean-variance model, system-and-threat model, and design goals. Cloud-assisted privacy-conscious MM is discussed in Section 4, in which the overall design framework comprising seven algorithms is presented and the details of each algorithm are separately elaborated. Some theoretical analyses, including the correctness of CM decryption, the correctness of portfolio proportion decryption, the privacy analysis, and the computation complexity analysis, are shown in Section 5, followed by the experimental analysis in Section 6. Finally, the last section concludes this study.

2. Related works

The present work is related to privacy-preserving computation outsourcing aimed at the indicators of a matrix, such as inversion, the determinant, matrix multiplication, and matrix decomposition. Lei et al. applied random permutation matrices and diagonal matrices, which separately work for location confusion and value diffusion, into different cases of matrix indicators, such as large-matrix inversion [13], large-matrix determinant [12], large-matrix rank decomposition [14], and large-matrix multiplication [11]. Large-scale matrix eigen-decomposition, singular-value decomposition [49], and QR factorization [23] are securely outsourced to a public cloud with privacy-preserving transformations. Luo et al. also suggested two

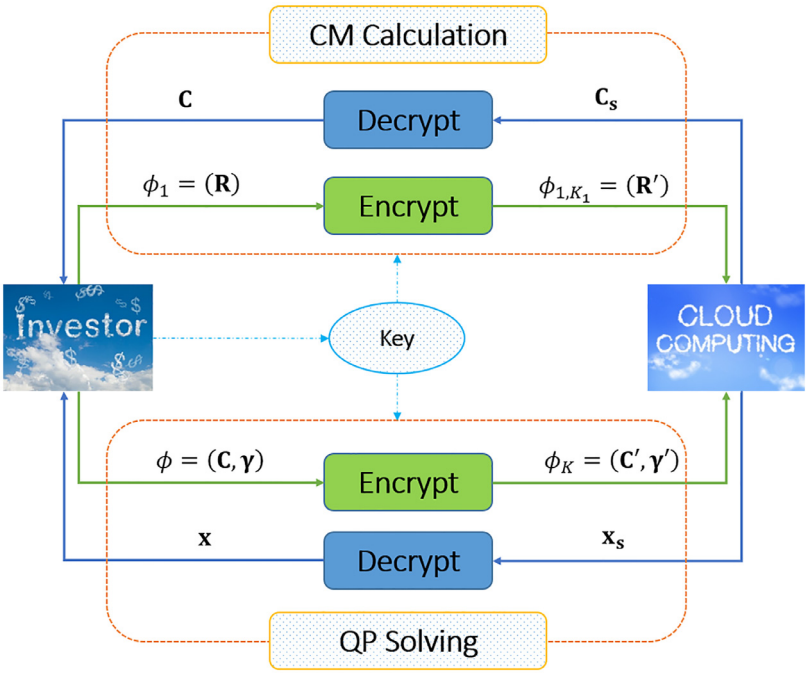

Fig. 1. The system model of MM outsourcing.

outsourcing algorithms for large-scale QR and LU factorizations [24]. In addition, Domingo–Ferrer systematically investigated how to exploit public clouds to calculate scalar products and matrix products on privacy-protected data [7]. The MM is not involved in these existing works.

Our work has a connection with secure outsourcing related to equations such as linear equation, linear programming, quadratic programming, convex optimization and linear regression. With the help of the iterative method that can be easily implemented, Wang et al. realized a secure outsourcing for large-scale systems of linear equations to a public cloud [34,35]. Likewise, they put forward the secure outsourcing of large-scale linear programming by employing a set of secret matrices and vectors to hide the sensitive information [32,33]. These two kinds of outsourcing are further improved in [5], in which some special linear transformations are designed for linear equation solving without homomorphic encryptions and interactions with the cloud, and a new standard natural form is proposed for linear programming solutions. Chen et al. used the sparse matrix to outsource large-scale systems of linear equations for the first time, and this algorithm only requires one-round communication, which avoids frequent interactions with the cloud [6]. Salinas et al. also exploited the sparsity for linear equations with a lower computational complexity and a smaller number of input/output operations [28]. A novel distributed outsourcing version in ad-hoc clouds was proposed in [30], in which a robust consensus-based algorithm was designed for distributively outsourcing linear algebraic equations into each cloud agent with guaranteed privacy. The case of efficiently outsourcing large-scale linear regression was investigated in [4], in which two protocols with different advantages are constructed. Zhou and Li harnessed the public cloud to build a secure, verifiable, and efficient outsourcing of large-scale quadratic programming [48]. Xu et al. suggested the case of convex optimization with an efficient integrity verification based on the inherent structure of the optimization problems [40]. In addition, Liao et al. aimed at large-scale convex separable programming and developed an efficient outsourcing framework based only on arithmetic operations [16]. These works did not consider the privacy-preserving outsourcing of MM.

The proposed scheme is similar to some privacy-preserving applications. In the image field, compressive sensing can offer high-efficiency large-scale image data sampling, but the reconstruction complexity is extremely large. Thus, securely outsourcing the image-reconstruction service to the public cloud was studied in [36]. Analogously, Hu et al. established an outsourcing framework for image-reconstruction and identity-authentication services by integrating compressive sensing and optimization outsourcing [10]. For more efficient reconstruction, the image-sparse reconstruction service was outsourced in parallel multi-clouds [47], which effectively protects the support set based on the parallel compressive sensing theories [8,41,45]. Later, this sparse reconstruction service was extended to the sparse robustness decoding service when two-dimensional signals are transmitted over a lossy channel with packet loss [46]. In a smart grid field, some grid data must be stored and managed by the cloud computing resources, and these data are often highly confidential; thus, Sarker et al. developed an economic dispatch linear programming with applications in grid data management [29]. In the numerical computation field, Liu et al. presented some privacy-preserving computation schemes in terms of floating-point numbers [20], public data[21], and rational numbers [17]. Furthermore, they extended these numerical computations to some

application scenarios, such as a calculation toolkit [19] and support vector machine for drug discovery [18] in the same privacy-preserving manner. In the scholarly big-data field, deep learning has widespread applications, such as [27,37,38,42], and Zhang et al. designed privacy-preserving deep computation models in the cloud for big-data feature learning [43,44]. Our scheme focuses on the privacy-conscious MM in the portfolio field for the first time.

3. Problem formulation

3.1. Background of Markowitz mean-variance model

The purpose of MM is to maximize the anticipated return by assembling a portfolio of assets when the level of risk that behaves as a variance is given, which is aimed at a global consideration, i.e., a portfolio's risk and return, rather than an asset's. This theory is based on the following assumptions [26]:

- An investor can reconsider the investment portfolio according to the probability distribution of the asset return during the holding period of the trade.
- An investor estimates the risk of the portfolio based on the expected return rate of assets.
- An investor's decision is merely based on the risks and benefits of assets.
- Investors expect the maximum benefit at a certain level of risk. In contrast, investors expect the minimum risk at a certain level of income.

Assume that there are n risky assets and let $\boldsymbol{\gamma}$ be an $n \times 1$ vector of anticipated return rate, where γ_i is the anticipated return rate in asset i; $\mathbf{x}$ be an $n \times 1$ vector of the decision variable, where x_i is the proportion of wealth invested in asset i; $\mathbf{C}$ be the $n \times n$ CM of sample return rate in the investments; and ε be the expected portfolio return of the portfolio. It can be represented by the following form:

$$\begin{cases} \min \sum\limits_{i,j=1}^{n} C_{ij} x_i x_j \\ \max\ \varepsilon = \sum\limits_{i=1}^{n} \gamma_i x_i \\ \text{s.t.} \sum\limits_{i=1}^{n} x_i = 1 \end{cases}, \tag{1}$$

or

$$\begin{cases} \min\ \mathbf{x}^{\mathrm{T}} \mathbf{C} \mathbf{x} \\ \max\ \varepsilon = \mathbf{x}^{\mathrm{T}} \gamma \\ \text{s.t.} \sum\limits_{i=1}^{n} x_i = 1 \end{cases}, \tag{2}$$

which, from a mathematical perspective, can be regarded as a quadratic programming problem that determines the values for a series of variables to minimize an objective function, subject to some constraints. In general, the CM is generated by the sample return rate data, because it is difficult to obtain it directly. Let $\mathbf{R}$ be the sample data of n assets in a time horizon, which generates the covariance matrix $\mathbf{C}$, where C_{ij} is the covariance of the ith column and jth column.

3.2. System and threat model

An investor with limited computation resources intends to outsource the MM to a cloud service center in which there is a robust computation capacity and abundant computation software. The computation outsourcing of MM in (2) can be simply marked as $\phi = (\mathbf{C}, \gamma)$. Prior to solving ϕ, a subproblem, the computation of CM $\mathbf{C}$, must first be outsourced to the cloud, which returns the result to the investor. This is because the complexity of computing $\mathbf{C}$ is very high, and it is not cost-effective to compute it locally. This subproblem is marked as $\phi_1 = (\mathbf{R})$, as $\mathbf{C}$ is generated by $\mathbf{R}$. Thus, the proposed MM outsourcing is decomposed into two outsourcing tasks: the CM computation $\phi_1 = (\mathbf{R})$ and the QP solving $\phi = (\mathbf{C}, \gamma)$. The system model is illustrated in Fig. 1.

The first task is to outsource the CM. The investor encrypts $\phi_1 = (\mathbf{R})$ with a secret key K_1 to generate an encrypted $\mathbf{R}'$, which is then delivered to the cloud. Upon receiving the encrypted $\phi_{1,K_1} = (\mathbf{R}')$, the cloud carries out the CM computation and returns the computation result $\mathbf{C_s}$ to the investor, who verifies the returned result and decrypts it to acquire the real covariance matrix $\mathbf{C}$.

The second task is to outsource the QP. The investor generates a secret key K different from K_1 to encrypt $\phi = (\mathbf{C}, \gamma)$ into $\phi_K = (\mathbf{C}', \gamma')$. Later, the encrypted version ϕ_K is sent to the cloud, which solves it using the relevant software. The solution $\mathbf{x_s}$ is sent back to the, who checks the result and decrypts it to obtain the real solution.

In general, the investor cannot fully trust a cloud that is curious about the computation content and attempts to learn some sensitive information from it, even though it is able to faithfully perform its computation duty. Thus, to tackle this kind of semi-trusted cloud, input/output privacy designs are essential. In addition to the cloud's curiosity, malicious behaviors should be considered. Compared with the former, the latter has a greater threat to outsourcing, as the result may be

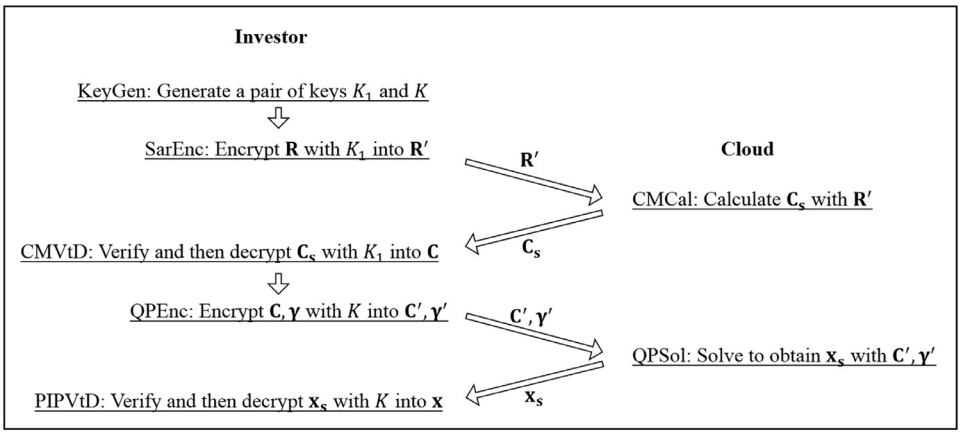

Fig. 2. The flow of the design framework.

altered or an arbitrary result could be returned to the investor. Therefore, the returned result can be reliably verified. Considering the portfolio closely linked to the investor's benefit, we select the malicious type of cloud as the threat model. In addition, the cloud is considered to launch a ciphertext-only attack that only leverages the intercepted ciphertext to reveal the confidentiality.

3.3. Design goals

The proposed outsourcing should satisfy the following four goals.

- Correctness. A correct answer $\mathbf{x}$ of the MM can be finally obtained, provided that both the investor and the cloud behave faithfully in accordance with the outsourcing protocol.
- Input/output privacy. The cloud cannot learn meaningful knowledge of the input $\phi_{1,K_1} = \left(\mathbf{R}'\right)$ and $\phi_K = \left(\mathbf{C}', \gamma'\right)$ or the output $\mathbf{C_s}$ and $\mathbf{x_s}$.
- Verification. The results $\mathbf{C_s}$ and $\mathbf{x_s}$ returned by the cloud can be verified by the investor and are exactly what the investor wants to know.
- Efficiency. The local computation workload for the investor can be greatly decreased through outsourcing CM and QP to the cloud. Meanwhile, the cloud's computation workload can remain consistent with the original system.

4. Cloud-assisted privacy-conscious MM

In this section, we first discuss the overall design framework of cloud-assisted privacy-conscious MM, which is composed of seven algorithms (KeyGen, SarEnc, CMCal, CMVtD, QPEnc, QPSol, and PIPVtD), and then elaborate upon the details of the individual algorithms.

4.1. Design framework

In our design framework, the idea of random transformation is to hide the information of data outsourced to the cloud; thus, some random matrices controlled by the security parameter are utilized to secretly transform ϕ_1 and ϕ. The flow of the design framework is visually described in Fig. 2, in which there are seven algorithms in total, including five algorithms (KeyGen, SarEnc, CMVtD, QPEnc, PIPVtD) handled by the investor and two (CMCal, QPSol) on the cloud side. They are summarized below and instantiated later.

- KeyGen$(1^\lambda) \rightarrow \{K_1, K\}$. It is a key-generation algorithm, operated by the investor, that inputs a security parameter λ and outputs a pair of keys $K_1 = \{\mathbf{U}, \mathbf{Q_1}, \mathbf{Q_2}\}$ and $K = \{\mathbf{Q_3}\}$.
- SarEnc$(\phi_1, K_1) \rightarrow \{\phi_{1,K}\}$. It is a sample return matrix encryption algorithm, operated by the investor, that inputs $\phi_1 = (\mathbf{R})$ and K_1, and outputs $\phi_{1,K_1} = \left(\mathbf{R}'\right)$, where $\mathbf{R}'$ represents the encrypted version of the sample return rate matrix $\mathbf{R}$.
- CMCal$(\phi_{1,K_1}) \rightarrow \{\mathbf{C_s}\}$. It is a covariance matrix calculation algorithm, operated by the cloud, that inputs $\phi_{1,K_1} = \left(\mathbf{R}'\right)$ and outputs the covariance matrix $\mathbf{C_s}$.
- CMVtD$(\mathbf{C_s}, K_1) \rightarrow \{\mathbf{C}\}$. It is a covariance matrix verification-then-decryption algorithm, operated by the investor, that inputs $\mathbf{C_s}$ and K_1 and outputs $\mathbf{C}$ if $\mathbf{C_s}$ is validated.
- QPEnc$(\phi, K) \rightarrow \{\phi_K\}$. It is a quadratic programming encryption algorithm that inputs the tuple $\phi = (\mathbf{C}, \gamma)$ and K, and outputs the encrypted tuple $\phi_K = \left(\mathbf{C}', \gamma'\right)$, where $\mathbf{C}'$ and γ' are the encrypted versions of the covariance matrix $\mathbf{C}$ and the anticipated return rate vector γ', respectively.

- QPSol(ϕ_K) → {$\mathbf{x_s}$}. It is a quadratic programming solving algorithm, operated by the cloud, that inputs $\phi_K = \left(\mathbf{C}', \gamma'\right)$ and outputs $\mathbf{x_s}$, which is the encrypted version of the real portfolio proportion vector $\mathbf{x}$.
- PIPVtD($\mathbf{x_s}$)→ {$\mathbf{x}$}. It is a portfolio proportion verification-then-decryption algorithm, operated by the investor, that inputs $\mathbf{x_s}$ and K, and outputs $\mathbf{x}$ if $\mathbf{x_s}$ is validated.

4.2. Algorithm details

In the following, we elaborate upon the details of each algorithm listed above.

4.2.1. Key-generation algorithm

A scrambling matrix is a matrix in which there is only one "1" in each row and each column, and all the remaining elements are "0"s [25]. It has extensive applications in engineering and some special properties. For example, its inverse matrix is its transpose. In the proposed key-generation algorithm, three scrambling matrices $\mathbf{Q}_\epsilon$, $1 \le \epsilon \le 3$, are defined as

$$Q_\epsilon(i, j) = \delta_{\sigma_\epsilon(i), j}, 1 \le i, j \le n, \tag{3}$$

where $\sigma(\cdot)$ is a one-to-one function whose domain and range are the same set $\{1, 2, \cdots, n\}$, and $\delta_{i,j}$ is the Kronecker delta function satisfying

$$\delta_{i,j} = \begin{cases} 0, i \neq j \\ 1, i = j \end{cases} \tag{4}$$

Furthermore, the investor needs to generate an $n \times n$ diagonal matrix $\mathbf{U}$ in which the principle diagonal elements are n random positive numbers u_i, $i = 1, 2, \cdots n$. The value range of u_i depends on that of $\mathbf{C}$, i.e., maintaining a consistent range of $\mathbf{C}$.

4.2.2. Sample return matrix encryption algorithm

Before outsourcing the sample data $\mathbf{R}$ to the cloud, the investor is able to protect the privacy of $\mathbf{R}$ through the following transformation using the key K_1,

$$\mathbf{R}' = \mathbf{Q}_1\mathbf{R}\mathbf{P}, \tag{5}$$

where

$$\mathbf{P} = \mathbf{Q}_2^{-1}\mathbf{U}. \tag{6}$$

$\mathbf{Q}_1$ and $\mathbf{P}$ have full rank; thus, $\mathbf{Q}_1^{\mathrm{T}}$ and $\mathbf{P^T}$ must be full-rank. Although we limit the size of the sample matrix $\mathbf{R}$ to $n \times n$ for convenience, it is enough to generate the valid CM when n is not too small.

4.2.3. Covariance matrix calculation algorithm

After receiving $\mathbf{R}'$, the cloud calculates the CM of $\mathbf{R}'$ based on the cloud computing resources and the relevant software, and the calculation result is marked as $\mathbf{C_s}$. In addition, the cloud may attempt to know K_1 to reveal the input privacy of $\mathbf{R}'$ and the output privacy of $\mathbf{C_s}$, because decrypting $\mathbf{R}'$ into $\mathbf{R}$ and $\mathbf{C_s}$ into $\mathbf{C}$ requires knowledge of $\mathbf{Q}_1^{\mathrm{T}}$ and $\mathbf{P^T}$. Without knowing $\mathbf{Q}_1^{\mathrm{T}}$ and $\mathbf{P^T}$, it is impossible for the cloud to determine the exact elements of $\mathbf{R}$ and $\mathbf{C}$.

4.2.4. Covariance matrix verification-then-decryption algorithm

When receiving $\mathbf{C_s}$, the investor needs to verify whether it is feasible before decryption. The method of random sampling can be used in our verification, which has been used in some computation outsourcing [11–13]. The investor randomly selects a few vectors in $\mathbf{R}$ and calculates the corresponding covariance. According to these calculations, the investor determines the covariance values of the same positions in $\mathbf{C}$ and checks whether they are all correct. Repeating this process multiple times, at least twenty times in general [2], ensures that $\mathbf{C}$ is a correct answer if it passes all the verifications. Otherwise, the investor rejects it. The decryption method is

$$\mathbf{C} = \mathbf{Q}_2^{\mathrm{T}}\mathbf{U}^{-1}\mathbf{C_s}\mathbf{U}^{-1}\mathbf{Q}_2, \tag{7}$$

which is demonstrated in the next section.

4.2.5. Quadratic programming encryption algorithm

The QP problem in (2) should be encrypted before being sent to the cloud; therefore, the key K is adopted to hide the sensitive information of $\mathbf{C}$ and $\boldsymbol{\gamma}$ as follows:

$$\begin{cases} \mathbf{C}' = \mathbf{Q}_3^{\mathrm{T}}\mathbf{C}\mathbf{Q}_3, \\ \gamma' = \mathbf{Q}_3^{\mathrm{T}}\gamma. \end{cases} \tag{8}$$

The QP problem will be changed into the encrypted form:

$$\begin{cases} \min\ \mathbf{x}^{\mathrm{T}}\mathbf{C}'\mathbf{x}, \\ \max\ \varepsilon = \mathbf{x}^{\mathrm{T}}\gamma', \\ s.t.\ \sum_{i=1}^{n} x_i = 1. \end{cases} \tag{9}$$

4.2.6. Quadratic programming solving algorithm

The MM is a QP problem under the constrained condition, which has a unique solution for the given expected revenue with the minimum variance. The cloud can solve (9) using its computing resources and relevant software. For example, the Frontcon function in MATLAB can be exploited to solve this problem. Because $\mathbf{Q}_3$ has full rank, $\mathbf{Q}_3^{\mathrm{T}}$ must be full-rank. Without the knowledge of $\mathbf{Q}_3$, it is impossible for the cloud to reveal the exact elements of $\mathbf{C}$, $\boldsymbol{\gamma}$, and $\mathbf{x}$.

4.2.7. Portfolio proportion verification-then-decryption algorithm

For the verification of the answer, two methods can be provided for the investor. One method is to only utilize $\sum_{i=1}^{n} x_i = 1$, and the other method is to utilize the KKT condition, in which the optimal solution can make the partial derivative of the Lagrange function equal to zero [1]. The first method is a weak verification because the cloud still fabricates the answer to enable $\sum_{i=1}^{n} x_i = 1$. The second is accurate and can distinguish the answer with a probability of 100%. However, the complexity of the second method is much higher than that of the first one. Which one is employed depends on the threat level of the cloud. The decryption method is

$$\mathbf{x} = \mathbf{Q}_3\mathbf{x}_{\mathbf{S}}, \tag{10}$$

which is demonstrated in the next section.

5. Theoretical discussions

5.1. The correctness of covariance matrix decryption

Theorem 1. *The covariance matrix can be correctly decrypted.*

Proof. In order to easily understand, we describe $\mathbf{R}$ and $\mathbf{C}$ in the intuitive forms:

$$\mathbf{R} = \begin{bmatrix} R_{11} & R_{12} & \cdots & R_{1n} \\ R_{21} & R_{22} & \cdots & R_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ R_{n1} & R_{n2} & \cdots & R_{nn} \end{bmatrix}. \tag{11}$$

and

$$\mathbf{C} = \begin{bmatrix} C_{11} & C_{12} & \cdots & C_{1i} & \cdots & C_{1j} & \cdots & C_{1n} \\ C_{21} & C_{22} & \cdots & C_{2i} & \cdots & C_{2j} & \cdots & C_{2n} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ C_{i1} & C_{i2} & \cdots & C_{ii} & \cdots & C_{ij} & \cdots & C_{in} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ C_{j1} & C_{j2} & \cdots & C_{ji} & \cdots & C_{jj} & \cdots & C_{jn} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ C_{n1} & C_{n2} & \cdots & C_{ni} & \cdots & C_{nj} & \cdots & C_{nn} \end{bmatrix}. \tag{12}$$

In essence, calculating $\mathbf{C}$ with $\mathbf{R}$ takes the following equation:

$$C_{ij} = \frac{1}{n-1}\sum_{l=1}^{n}\left(R_{li} - \overline{R_i}\right)\left(R_{lj} - \overline{R_j}\right), \tag{13}$$

where $\overline{R_i}$ is the mean of the ith column $\mathbf{R_i}$.

We consider the effect of $\mathbf{Q}_1$, $\mathbf{Q}_2$, and $\mathbf{U}$ because of $\mathbf{R}' = \mathbf{Q}_1\mathbf{R}\mathbf{Q}_2^{-1}\mathbf{U}$. First of all, the function of $\mathbf{Q}_1$ randomly rearranges the rows of $\mathbf{R}$, which means that it rearranges the terms in (13) and the result of $\sum_{l=1}^{n}\left(R_{li} - \overline{R_i}\right)\left(R_{lj} - \overline{R_j}\right)$ is unchanged. Thus, $\mathbf{Q}_1$ cannot alter $\mathbf{C}$. □

Moreover, regarding $\mathbf{Q}_2$, it randomly rearranges the columns of $\mathbf{R}$. Let $\mathbf{R}^{(1)} = \mathbf{R}\mathbf{Q}_2^{-1}$, and the corresponding CM be $\mathbf{C}^{(1)}$. We provide an example to understand this transform process. If we only exchange the ith and jth columns in $\mathbf{R}$, then the covariance values related to the ith column and the jth column $\mathbf{C}^{(1)}$ will be changed, which can be expressed as

$$C_{mi}^{(1)} = C_{mj} = \frac{1}{n-1}\sum_{l=1}^{n}\left(R_{lm} - \overline{R_m}\right)\left(R_{lj} - \overline{R_j}\right), \tag{14}$$

$$C_{im}^{(1)} = C_{jm} = \frac{1}{n-1}\sum_{l=1}^{n}\left(R_{lj} - \overline{R_j}\right)\left(R_{lm} - \overline{R_m}\right), \tag{15}$$

$$C_{mj}^{(1)} = C_{mi} = \frac{1}{n-1}\sum_{l=1}^{n}\left(R_{lm} - \overline{R_m}\right)\left(R_{li} - \overline{R_i}\right), \tag{16}$$

$$C_{jm}^{(1)} = C_{im} = \frac{1}{n-1}\sum_{l=1}^{n}\left(R_{li} - \overline{R_i}\right)\left(R_{lm} - \overline{R_m}\right). \tag{17}$$

From the above equations, it can be inferred that for each C_{ij} in $\mathbf{C}$, there is a unique $C_{i'j'}^{(1)}$ in $\mathbf{C^{(1)}}$ that is mapped to it. It can be understood that each C_{ij} is moved to some $C_{i'j'}^{(1)}$ from $\mathbf{C}$ to $\mathbf{C^{(1)}}$ in both row and column directions. Specifically, it can be represented by

$$\mathbf{Q}_2\mathbf{C}\mathbf{Q}_2^{\mathrm{T}} = \mathbf{C}^{(1)}. \tag{18}$$

Finally, we investigate $\mathbf{U}$. Let $\mathbf{R}^{(2)} = \mathbf{RU}$, and the corresponding CM is $\mathbf{C}^{(2)}$, in which each element can be computed as

$$C_{ij}^{(2)} = \frac{u_i u_j}{n-1}\sum_{l=1}^{n}\left(R_{li} - \overline{R_i}\right)\left(R_{lj} - \overline{R_j}\right). \tag{19}$$

Further, we have

$$\mathbf{C}^{(2)} = \begin{bmatrix} u_1{}^2 & u_1u_2C_{12} & \cdots & u_1u_nC_{1n} \\ u_2u_1C_{21} & u_2{}^2C_{22} & \cdots & u_2u_nC_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ u_nu_1C_{n1} & u_nu_2C_{n2} & \cdots & u_n{}^2C_{nn} \end{bmatrix}. \tag{20}$$

As a consequence,

$$\mathbf{C}^{(2)} = \mathbf{UCU}. \tag{21}$$

Based on the above analyses, one can finally determine the relationship between $\mathbf{C}$ and $\mathbf{C_s}$, as follows

$$\mathbf{C} = \mathbf{Q}_2^{\mathrm{T}}\mathbf{U}^{-1}\mathbf{C_s}\mathbf{U}^{-1}\mathbf{Q}_2, \tag{22}$$

which implies that the proposed encryption method yields the correct decryption result.

5.2. Correctness of portfolio proportion decryption

Theorem 2. *The portfolio proportion can be correctly decrypted.*

Proof. Let $\mathbf{x_s}$ be the solution of the encrypted QP problem in (9), and associating (8), one obtains

$$\mathbf{x_s}^{\mathrm{T}}\mathbf{C}'\mathbf{x_s} = \mathbf{x_s}^{\mathrm{T}}\mathbf{Q_3}^{\mathrm{T}}\mathbf{C}\mathbf{Q_3}\mathbf{x_s} = (\mathbf{Q_3}\mathbf{x_s})^{\mathrm{T}}\mathbf{C}(\mathbf{Q_3}\mathbf{x_s}) \tag{23}$$

and

$$\mathbf{x_s}^{\mathrm{T}}\gamma' = \mathbf{x_s}^{\mathrm{T}}\mathbf{Q_3}^{\mathrm{T}}\gamma = (\mathbf{Q_3}\mathbf{x_s})^{\mathrm{T}}\gamma. \tag{24}$$

As a result, after receiving the result $\mathbf{x_s}$ returned by the cloud, the investor must be able to obtain the real answer $\mathbf{x} = \mathbf{Q_3}\mathbf{x_s}$. In addition, it is noted that $\sum_{i=1}^{n} x_i = \sum_{i=1}^{n} x_{s,i} = 1$ obviously holds. □

5.3. Privacy analysis

From the cloud perspective, the attacked objectives are $\mathbf{R'}$, $\mathbf{C_s}$, $\mathbf{C'}$, $\boldsymbol{\gamma'}$, $\mathbf{x_s}$ and their input privacy or output privacy is cryptanalyzed under the mode of ciphertext-only attack. In other words, the keys K_1 and K are randomly updated in each outsourcing process and are never used twice. First, for the input privacy of $\mathbf{R'}$, which is encrypted by full-rank matrices, the cloud attempts to know $K_1 = \{\mathbf{U}, \mathbf{Q}_1, \mathbf{Q}_2\}$. If the precision of u_i is q, then the probability of guessing K_1 is $\left(\frac{1}{q^n(n!)^2}\right)$, where $n!$ represents the factorial of n. Such a tiny probability is reliable to prevent the cloud's brute-force attack. Likewise, the output privacy of $\mathbf{C_s}$ is guaranteed by K_1, which is secure against the exhaustive analysis.

Furthermore, $K = \{\mathbf{Q_3}\}$ provides the safeguard for the input privacy of $\mathbf{C'}$, $\boldsymbol{\gamma'}$ and the output privacy of $\mathbf{x_s}$. The probability of success of guessing K is $\frac{1}{n!}$. If n is too small, then this probability might not efficiently ensure the cloud's violent analysis of $\mathbf{C'}$, $\boldsymbol{\gamma'}$ and $\mathbf{x_s}$. However, too small n means a small number of risky assets, which is unworthy of outsourcing, as the low computation workload can be done locally without effort. The greater n is worth outsourcing and the security will be increased. In the proposed scheme, we design a more secure encryption method for the output privacy of $\mathbf{R}$ than $\mathbf{x}$. This is because the sample return matrix belongs to the raw data, which contains more sensitive information from the point view of the investor. For the portfolio proportion $\mathbf{x}$, the investor frequently adjusts it according to the market fluctuation in practice. If one can seek a more secure encryption mode, then many schemes like homomorphic encryption [9] can be considered. Nevertheless, a stronger security level implies higher complexity, which results in greater losses than gains. Thus, the proposed method is suitable for the encryption candidate.

Table 1
Computation Complexity.

	Investor					Cloud	
Algorithm	KeyGen	SarEnc	CMVtD	QPEnc	PIPVtD	CMCal	QPSol
Complexity	$O(n)$	$O(n^2)$	$O(n)$	$O(n^2)$	$O(n^2)$	$O(n^3)$	$O(n^3)$

Table 2
Symbols and corresponding physical significance.

Symbol	Physical significance
$t_{original}$	the time of computing the original MM locally by the investor
t_{cloud}	the time of computing the outsourced MM by the cloud
$t_{investor1}$	the time of generating the secret key and encrypting the original MM by the investor
$t_{investor2}$	the time of verifying and decrypting the returned result by the investor
$t_{investor}$	$t_{investor1} + t_{investor2}$
$\frac{t_{original}}{t_{investor}}$	the investor speedup
$\frac{t_{original}}{t_{cloud}}$	the cloud efficiency
$\frac{t_{investor}+t_{cloud}}{t_{original}} - 1$	the relative extra cost

5.4. Computation complexity analysis

We discuss the computational complexity of the seven algorithms (KeyGen, SarEnc, CMCal, CMVtD, QPEnc, QPSol, PIPVtD) theoretically. In KeyGen, generating four matrices $\mathbf{U}$, $\mathbf{Q}_1$, $\mathbf{Q}_2$, and $\mathbf{Q}_3$, one of which requires generating n numbers, consumes time $O(n)$. In CMEnc, it seems to be the matrix–matrix multiplications, but observing $\mathbf{U}$, $\mathbf{Q}_1$ and $\mathbf{Q}_2$ reveals that they are matrix–vector multiplications. Therefore, the complexity is $O(n^2)$. In CMCal, calculating a covariance of two vectors takes time $O(n)$, and then n^2 covariance values lead to $O(n^3)$. In CMVtD, randomly calculating a certain number of covariance values consumes time $O(n^2)$ at most. In QPEnc, the complexity can be easily found to be $O(n)$. In QPSol, the complexity of QP is asymptotically the same as that of solving a normal linear programming problem, which usually requires more than $O(n^3)$ time [22]. In PIPVtD, the first method takes time $O(n)$, whereas the second one takes time $O(n^2)$ due to the matrix–vector multiplication involved; thus, the complexity is no more than $O(n^2)$. These analytic results are summarized in Table 1. In summary, the investor running five algorithms (KeyGen, SarEnc, CMVtD, QPEnc, PIPVtD) consumes time $O(n^2)$ in total, and the cloud carries out two algorithms (CMCal, QPSol) whose complexity is $O(n^3)$. As we can see, the investor can gain substantial computation savings by outsourcing MM to the cloud.

6. Experiment analysis

This section experimentally illustrates the efficiency of the proposed MM outsourcing scheme, although the theoretical analysis in terms of the computational complexity is assessed above. The experiment is performed on a PC with an Intel Core CPU i7-4510U and 8 GB RAM. The cloud server is simulated in the same PC, and both the investor's and cloud's computations are conducted by MATLAB software. The size of MM is chosen as a large range from 50 to 1600 and some randomly generated data is used for testing. The communication and storage costs are neglected.

Let $t_{original}$ and t_{cloud} denote the time that the investor computes the original problem locally and the time that the cloud computes the outsourced problem, respectively. In the outsourcing scheme, the time of the investor, denoted by $t_{investor}$, consists of the time during which the investor generates the secret key and encrypts the original problem and the time that the investor verifies and decrypts the returned result, which are denoted by $t_{investor1}$ and $t_{investor2}$, respectively. In spite of the substantial computation savings indicated by the theoretical analysis, we leverage some experimental data to validate that the outsourcing indeed benefits the investor. Three indicator ratios are defined to better reflect the efficiency performance. The first one is to measure the efficiency gain of the investor, $\frac{t_{original}}{t_{investor}}$, which is referred to as the investor speedup. Naturally, this value is a positive number greater than 1. The greater the value, the greater the efficiency gain. The second one is the metric of cloud efficiency, $\frac{t_{original}}{t_{cloud}}$, which signifies the degree of the time increase induced by the encryption operations. This value should be close to 1 as soon as possible, meaning that the computation cost does not increase too much after encryption. The last one is the relative extra cost, $\frac{t_{investor}+t_{cloud}-t_{original}}{t_{original}}$ $(= \frac{t_{investor}+t_{cloud}}{t_{original}} - 1)$, in which $t_{investor} + t_{cloud} - t_{original}$ stands for the extra cost due to outsourcing in comparison with the original problem. This value should approach 1, which indicates that the overall computation cost in the outsourcing scheme basically remains consistent with that of the original problem. Table 2 lists these symbols and corresponding physical significance for convenience.

The proposed MM outsourcing consists of two outsourcing problems, including the CM outsourcing and QP outsourcing. Table 3 lists the efficiency performance result of the CM outsourcing. Table 4 is about the efficiency performance of the QP outsourcing. The overall efficiency performance results are shown in Table 5. As can be seen from the three tables, the investor speedup increases rapidly with the number of assets, although the entire execution time, including $t_{original}$, t_{cloud},

Table 3
The efficiency performance of outsourcing CM. (Seconds).

No.	n	Original CM		Outsourced CM			Investor speedup	Cloud efficiency	Relative extra cost
		$t_{original}$	t_{cloud}	$t_{investor1}$	$t_{investor2}$	$t_{investor}$	$\frac{t_{original}}{t_{investor}}$	$\frac{t_{original}}{t_{cloud}}$	$\frac{t_{investor}+t_{cloud}}{t_{original}}-1$
1	50	0.246	0.279	0.052	0.029	0.081	3.012x	0.8926	0.4605
2	100	0.995	1.083	0.067	0.037	0.104	9.598x	0.9266	0.2028
3	200	3.976	4.182	0.070	0.056	0.126	31.652x	0.9510	0.0857
4	400	17.440	17.891	0.135	0.132	0.226	65.614x	0.9751	0.0424
5	800	76.948	76.095	0.447	0.775	1.222	63.019x	1.0126	0.0046
6	1600	370.947	368.125	1.431	4.753	6.184	59.986x	1.0077	0.0091

Table 4
The efficiency performance of outsourcing QP. (Seconds).

No.	n	Original QP		Outsourced QP			Investor speedup	Cloud efficiency	Relative extra cost
		$t_{original}$	t_{cloud}	$t_{investor1}$	$t_{investor2}$	$t_{investor}$	$\frac{t_{original}}{t_{investor}}$	$\frac{t_{original}}{t_{cloud}}$	$\frac{t_{investor}+t_{cloud}}{t_{original}}-1$
1	50	0.107	0.125	0.048	0.040	0.088	1.227x	0.8651	1.0025
2	100	0.307	0.352	0.060	0.052	0.111	2.754x	0.8815	0.5101
3	200	1.605	1.461	0.060	0.092	0.152	10.525x	1.1003	0.0070
4	400	13.240	9.337	0.079	0.169	0.248	53.945x	1.4716	−0.2729
5	800	98.466	100.126	0.205	0.889	1.094	91.764x	0.9833	0.0317
6	1600	1119.898	1103.112	0.947	4.948	5.895	188.986x	1.0152	−0.0097

Table 5
The efficiency performance of the whole MM outsourcing. (Seconds).

No.	n	Original MM		Outsourced MM			Investor speedup	Cloud efficiency	Relative extra cost
		$t_{original}$	t_{cloud}	$t_{investor1}$	$t_{investor2}$	$t_{investor}$	$\frac{t_{original}}{t_{investor}}$	$\frac{t_{original}}{t_{cloud}}$	$\frac{t_{investor}+t_{cloud}}{t_{original}}-1$
1	50	0.353	0.404	0.063	0.069	0.132	2.693x	0.8842	0.5106
2	100	1.302	1.435	0.081	0.089	0.170	7.703x	0.9132	0.2398
3	200	5.582	5.642	0.088	0.148	0.236	23.747x	0.9887	0.0554
4	400	30.680	27.228	0.168	0.301	0.469	65.767x	1.1325	−0.0952
5	800	175.414	176.222	0.595	1.664	2.259	78.071x	0.9953	0.0189
6	1600	1490.845	1471.237	2.317	9.701	12.018	124.056x	1.0133	−0.0051

$t_{investor1}$, $t_{investor2}$, and $t_{investor}$, increases. Three cases of the investor speedup are $3.012\times$, $1.227\times$, and $2.693\times$ when n is equal to 50, but they achieve $59.986\times$, $188.986\times$, and $124.056\times$ when n is 1600. The efficiency gain is readily apparent from these results. In particular, the greater the value of n, the more worthwhile the outsourcing and the greater the efficiency gain. With respect to the cloud efficiency, the values in all three tables stay close to 1, which indicates that the cloud's computation over the encrypted domain does not cause an increase in the computation time and the proposed encryption methods can maintain a consistent computation time with the original problem. Regarding the relative extra cost, it will decrease as the size increases. When n approaches 1600, the outsourcing hardly effects any extra cost. It is again demonstrated that the advantage of the proposed outsourcing is more evident when the number of assets is greater.

7. Concluding remarks

This work investigated the privacy-preserving outsourcing of MM. Because a high-complexity covariance matrix calculation is contained in MM, the MM problem is decomposed into two outsourcing tasks, including a CM calculation task and a QP solving task. The input/output privacy is protected by some scrambling matrices and diagonal matrices in order to change the locations and values, respectively. It has been demonstrated that the calculation of CM and the solving of QP in the cloud side are correctly decrypted and effectively verified by the investor. In terms of the complexity, it has been validated by both the theoretical and experimental analyses that the investor can save a large amount of local computation resources through outsourcing, and does not introduce extra overload to the cloud, even though the cloud processes two tasks over an encrypted domain. It is hoped that this study can guide more research in cloud-assisted privacy-conscious computation models with respect to financial analysis, forecasting, and investment, which benefits investors and enterprises to make fast and efficient financial decisions.

Acknowledgments

The authors would like to thank the editor and anonymous reviewers for their valuable comments and suggestions to improve the quality of this manuscript. The work was funded by the Research Start-Up Fund of NUAA (Grant No. 1015-YAH19001) and the National Natural Science Foundation of China (Grant Nos. 61572089, 61602158, 61672038, U1536204).

References

[1] S. Boyd, L. Vandenberghe, Convex optimization, Cambridge University press, 2004.
[2] J. Camenisch, S. Hohenberger, M.O. Pedersen, Batch verification of short signatures, in: Eurocrypt, 4515, Springer, 2007, pp. 246–263.
[3] C.P. Chen, C.-Y. Zhang, Data-intensive applications, challenges, techniques and technologies: a survey on big data, Inf. Sci. 275 (2014) 314–347.
[4] F. Chen, T. Xiang, X. Lei, J. Chen, Highly efficient linear regression outsourcing to a cloud, IEEE Trans. Cloud Comput. 2 (4) (2014) 499–508.
[5] F. Chen, T. Xiang, Y. Yang, Privacy-preserving and verifiable protocols for scientific computation outsourcing to the cloud, J. Parallel Distrib. Comput. 74 (3) (Mar. 2014) 2141–2151.
[6] X. Chen, X. Huang, J. Li, J. Ma, W. Lou, D. Wong, New algorithms for secure outsourcing of large-scale systems of linear equations, IEEE Trans. Inf. Forensics Security 10 (1) (2015) 69–78.
[7] J. Domingo-Ferrer, S. Ricci, C. Domingo-Enrich, Outsourcing scalar products and matrix products on privacy-protected unencrypted data stored in untrusted clouds., Inf. Sci. 436 (2018) 320–342.
[8] H. Fang, S.A. Vorobyov, H. Jiang, O. Taheri, Permutation meets parallel compressed sensing: how to relax restricted isometry property for 2D sparse signals, IEEE Trans. Signal Process. 62 (1) (2014) 196–210.
[9] C. Gentry, et al., Fully homomorphic encryption using ideal lattices., in: STOC, 9, 2009, pp. 169–178.
[10] G. Hu, D. Xiao, T. Xiang, S. Bai, Y. Zhang, A compressive sensing based privacy preserving outsourcing of image storage and identity authentication service in cloud, Inf. Sci. 387 (2017) 132–145.
[11] X. Lei, X. Liao, T. Huang, F. Heriniaina, Achieving security, robust cheating resistance, and high-efficiency for outsourcing large matrix multiplication computation to a malicious cloud, Inf. Sci. 280 (2014) 205–217.
[12] X. Lei, X. Liao, T. Huang, H. Li, Cloud computing service: the case of large matrix determinant computation, IEEE Trans. Serv. Comput. 8 (5) (2015) 688–700.
[13] X. Lei, X. Liao, T. Huang, H. Li, C. Hu, Outsourcing large matrix inversion computation to a public cloud, IEEE Trans. Cloud Comput. 1 (1) (2013) 1–1.
[14] X. Lei, X. Liao, X. Ma, L. Feng, Securely and efficiently perform large matrix rank decomposition computation via cloud computing, Cluster Comput. 18 (2) (2015) 989–997.
[15] D. Li, W.-L. Ng, Optimal dynamic portfolio selection: multiperiod mean-variance formulation, Math. Finance 10 (3) (2000) 387–406.
[16] W. Liao, C. Luo, S. Salinas, P. Li, Efficient secure outsourcing of large-scale convex separable programming for big data, IEEE Trans. Big Data, in press (2017), doi:10.1109/TBDATA.2017.2787198.
[17] X. Liu, R. Choo, R. Deng, R. Lu, J. Weng, Efficient and privacy-preserving outsourced calculation of rational numbers, IEEE Trans. Depend. Secure Comput. 15 (1) (2018) 27–39.
[18] X. Liu, R. Deng, K.-K.R. Choo, Y. Yang, Privacy-preserving outsourced support vector machine design for secure drug discovery, IEEE Trans. Cloud Comput., in press (2018), doi:10.1109/TCC.2018.2799219.
[19] X. Liu, R. Deng, K.-K.R. Choo, Y. Yang, H. Pang, Privacy-preserving outsourced calculation toolkit in the cloud, IEEE Trans. Depend. Secure Comput., in press (2018), doi:10.1109/TDSC.2018.2816656.
[20] X. Liu, R.H. Deng, W. Ding, R. Lu, B. Qin, Privacy-preserving outsourced calculation on floating point numbers, IEEE Trans. Inf. Forensics Security 11 (11) (2016) 2513–2527.
[21] X. Liu, B. Qin, R. Deng, Y. Li, An efficient privacy-preserving outsourced computation over public data, IEEE Trans. Serv. Comput. 10 (5) (2017) 756–770.
[22] D.G. Luenberger, Y. Ye, et al., Linear and nonlinear programming, 2, Springer, 1984.
[23] C. Luo, K. Zhang, S. Salinas, P. Li, Efficient privacy-preserving outsourcing of large-scale QR factorization, in: Proc. IEEE Trustcom, IEEE, 2017, pp. 917–924.
[24] C. Luo, K. Zhang, S. Salinas, P. Li, Secfact: secure large-scale QR and LU factorizations, IEEE Trans. Big Data, in press (2017), doi:10.1109/TBDATA.2017.2782809.
[25] R.C. Lyndon, P.E. Schupp, Combinatorial group theory, Springer, 2015.
[26] H. Markowitz, Portfolio selection, J. Finance 7 (1) (1952) 77–91.
[27] M.M. Najafabadi, F. Villanustre, T.M. Khoshgoftaar, N. Seliya, R. Wald, E. Muharemagic, Deep learning applications and challenges in big data analytics, J. Big Data 2 (1) (2015) 1.
[28] S. Salinas, C. Luo, X. Chen, W. Liao, P. Li, Efficient secure outsourcing of large-scale sparse linear systems of equations, IEEE Trans. Big Data 4 (1) (2018) 26–39.
[29] M.R. Sarker, J. Wang, Z. Li, K. Ren, Security and cloud outsourcing framework for economic dispatch, IEEE Trans. Smart Grid 9 (6) (2018) 5810–5819.
[30] W. Shen, B. Yin, X. Cao, Y. Cheng, X.S. Shen, A distributed secure outsourcing scheme for solving linear algebraic equations in ad hoc clouds, IEEE Trans. Cloud Comput., in press (2017), doi:10.1109/TCC.2016.2647718.
[31] M.C. Steinbach, Markowitz revisited: mean-variance models in financial portfolio analysis, SIAM Rev. 43 (1) (2001) 31–85.
[32] C. Wang, K. Ren, J. Wang, Secure and practical outsourcing of linear programming in cloud computing, in: Proc. IEEE Comput. Commun. (INFOCOM), IEEE, 2011, pp. 820–828.
[33] C. Wang, K. Ren, J. Wang, Secure optimization computation outsourcing in cloud computing: a case study of linear programming, IEEE Trans. Comput. 65 (1) (2016) 216–229.
[34] C. Wang, K. Ren, J. Wang, K.M.R. Urs, Harnessing the cloud for securely solving large-scale systems of linear equations, in: Proc. 31st Int. Conf. Distr. Comput. Syst. (ICDCS), IEEE, 2011, pp. 549–558.
[35] C. Wang, B. Zhang, K. Ren, J. Wang, Harnessing the cloud for securely outsourcing large-scale systems of linear equations, IEEE Trans. Parallel Distrib. Syst. 24 (6) (2013) 1172–1181.
[36] C. Wang, B. Zhang, K. Ren, J. Wang, Privacy-assured outsourcing of image reconstruction service in cloud, IEEE Trans. Emerg. Top. Comput. 1 (1) (2013) 166–177.
[37] W. Wang, J. Liu, F. Xia, I. King, H. Tong, Shifu: Deep learning based advisor-advisee relationship mining in scholarly big data, in: Proc. 26th Int. Conf. World Wide Web Companion, International World Wide Web Conferences Steering Committee, 2017, pp. 303–310.
[38] W. Wang, J. Liu, S. Yu, C. Zhang, Z. Xu, F. Xia, Mining advisor-advisee relationships in scholarly big data: A deep learning approach, in: IEEE/ACM Joint Conf. Digital Libraries (JCDL), IEEE, 2016, pp. 209–210.
[39] F. Xia, W. Wang, T.M. Bekele, H. Liu, Big scholarly data: a survey, IEEE Trans. Big Data 3 (1) (2017) 18–35.
[40] Z. Xu, C. Wang, K. Ren, L. Wang, B. Zhang, Proof-carrying cloud computation: the case of convex optimization, IEEE Trans. Inf. Forensics Security 9 (11) (2014) 1790–1803.
[41] L.Y. Zhang, K.-W. Wong, Y. Zhang, J. Zhou, Bi-level protected compressive sampling, IEEE Trans. Multimed. 18 (9) (2016) 1720–1732.
[42] Q. Zhang, L.T. Yang, Z. Chen, Deep computation model for unsupervised feature learning on big data, IEEE Trans. Serv. Comput. 9 (1) (2016) 161–171.
[43] Q. Zhang, L.T. Yang, Z. Chen, Privacy preserving deep computation model on cloud for big data feature learning, IEEE Trans. Comput. 65 (5) (2016) 1351–1362.

[44] Q. Zhang, L.T. Yang, Z. Chen, P. Li, M.J. Deen, Privacy-preserving double-projection deep computation model with crowdsourcing on cloud for big data feature learning, IEEE Internet Things J. 5 (4) (2018) 2896–2903.
[45] Y. Zhang, J. Zhou, F. Chen, L.Y. Zhang, K.-W. Wong, X. He, D. Xiao, Embedding cryptographic features in compressive sensing, Neurocomputing 205 (2016) 472–480.
[46] Y. Zhang, J. Zhou, Y. Xiang, L.Y. Zhang, F. Chen, S. Pang, X. Liao, Computation outsourcing meets lossy channel: secure sparse robustness decoding service in multi-clouds, IEEE Trans. Big Data, in press (2017), doi:10.1109/TBDATA.2017.2711040.
[47] Y. Zhang, J. Zhou, L.Y. Zhang, F. Chen, X. Lei, Support-set-assured parallel outsourcing of sparse reconstruction service for compressive sensing in multi-clouds, in: Proc. Int. Symp. Security Privacy Social Netw. Big Data (SocialSec), 2015, pp. 1–6.
[48] L. Zhou, C. Li, Outsourcing large-scale quadratic programming to a public cloud, IEEE Access 3 (2015) 2581–2589.
[49] L. Zhou, C. Li, Outsourcing eigen-decomposition and singular value decomposition of large matrix to a public cloud, IEEE Access 4 (2016) 869–879.
[50] X.Y. Zhou, G. Yin, Markowitz'S mean-variance portfolio selection with regime switching: a continuous-time model, SIAM J. Control Optim. 42 (4) (2003) 1466–1482.