A Secure and Efficient Computation Outsourcing Scheme for Multi-users

V. Sudarsan Rao[1(✉)] and N. Satyanarayana[2]

[1] Department of CSE, Khammam Institute of Technology and Sciences (KITS), Khammam, (T.S), India
sudharshan.cse2008@gmail.com

[2] Department of CSE, Nagole Institute of Technology and Sciences (NITS), Hyderabad, (T.S), India
nsn2008@gmail.com

Abstract. The outsourcing process is computationally secure if it is performed without unveiling to the other external agent or cloud, either the original data or the actual solution to the computations. Secure multiparty computation computes a certain function without revealing their private secret information.

In this paper, we presented a new secure and computationally efficient protocol utilizing multi cloud servers view. In our proposed protocol, encrypted data by different users is transformed to cloud. The protocol being non-interactive between users, gives the comparatively lesser computational and communication complexity. The analysis of our proposed protocol is also presented at the end of the paper.

Keywords: Access control · Lattice Based Encryption
Secure outsourcing · Cloud computing · Key issuing · Privacy

1 Introduction

Beside the tremendous advantages of outsourcing, client faces some challenges by outsourcing the computational task to cloud [8,9]. These are security, input-output privacy and verification of result. Consider a scenario where some mutually distrusted members are present, and they want to compute a complex function, which involves their own private inputs [10]. This scenario may be termed as secure multi-party computation. Suppose, $U_1, U_2, \cdots, U_m$ are m users, and each posses a private number $n_1, n_2, \cdots, n_m$. Consider function is,

$$\mathbb{FUNC} = f(n_1, n_2, \cdots, n_m), \tag{1}$$

which they want to co-operatively compute, but they don't want to expose n_i of corresponding U_i to other users U_j, $i \neq j$ & $i, j \in (1, 2, \cdots, m)$. Also they should guarantee that $\mathbb{FUNC}$ should not be known by any of the unauthorized user. Its observable that the computation and communication complexities are

© Springer Nature Singapore Pte Ltd. 2018
M. Singh et al. (Eds.): ICACDS 2018, CCIS 905, pp. 12–24, 2018.
https://doi.org/10.1007/978-981-13-1810-8_2

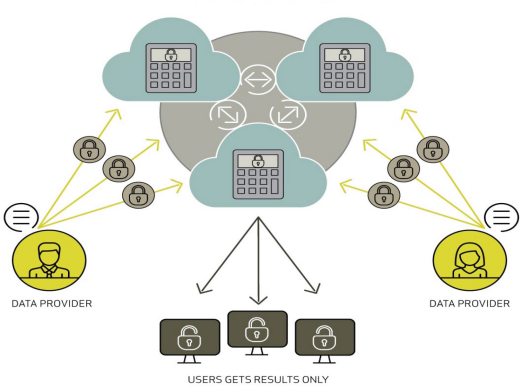


Fig. 1. General computational outsourcing scenario

mostly dependant on the complex nature of computation function. The scenario is shown as Fig. 1.

Recently, as the development of cloud computing [25], users' concerns about data security are the main obstacles that impedes cloud computing from wide adoption. These concerns are originated from the fact that sensitive data resides in public cloud [18], which is maintained and operated by untrusted cloud service provider (CSP) [20,22]. The expectation of users is that the cloud should compute the function having the inputs as private parameters of users in the encrypted/transformed form.

2 Secure Outsourcing Algorithms Classification

Increasing no. of smart equipments and their growing need to execute computationally large task resulting the outsourcing of any scientific computation to the cloud server an encouraging solution. The general nomenclature is represented as Fig. 2 below:-

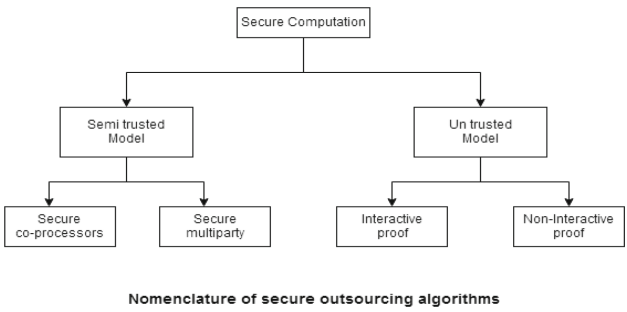


Fig. 2. Secure outsourcing algorithms nomenclature

2.1 Related Work

While outsourcing the private data functions to the cloud, there exist many problems and challenges. In past years, much research have been carried out to come up with various solutions for secure computational outsourcing. One solution was proposed by Gentry [2], in 2009 where a joint public key is used to encrypt their private input data and accordingly the notion was termed as Homomorphic encryption, which successively used in the secure outsourcing of practical complex problems. In the work presented by [7] has given a scheme where for encryption purpose, users' public keys are utilized, and cloud will be able to compute the function having their private inputs. A more secure outsourcing was given by Halevi et al. [13] in 2011, that was a non-interactive method for secure outsourcing [15]. [16] given a new fully homomorphic scheme, multikey FHE, which applied bootstrapping concept for secure outsourcing of computations. ABE, introduced as fuzzy identity-based encryption in [24], was firstly dealt with by Goyal et al. [1]. Two distinct and interrelated notions of ABE were determined in [1]. Accordingly, several constructions supporting for any kinds of access structures were provided [3,4] for practical applications [5,6]. Atallah et al. [8] offered an structure for secure outsourcing of scientific computations e.g. multiplication of matrices. Although, the solution used the disguise technique and thus leaded to leakage of private information. Atallah and Li [9] given an efficient protocol to outsource sequence comparison with two servers in secure manner. Furthermore, Benjamin and Atallah [10] addressed the problem of secure outsourcing for widely applicable linear algebraic computations. Atallah and Frikken [11] further studied this problem and gave improved protocols based on the so-called weak secret hiding assumption. Recently, Wang et al. [12] presented efficient mechanisms for secure outsourcing of linear programming computation.

In [14], a novel paradigm for outsourcing the decryption of ABE is given. Compared with our work, the two lack of the consideration on the eliminating the overhead computation at attribute authority. In 2014, Sudarshan et al. [27] proposed an Attribute-Based Encryption mechanism, applied for cloud security. Recently Lai et al. [17] given a construction with verifiable decryption, which achieves both security and verifiability without random oracles. Their task supplements a redundancy with ciphertext and uses this redundancy for correctness checking.

2.2 Motivation and Contribution

In the scenario of outsourcing private inputs or computational function to cloud, There exist hurdles in following two aspects - One is in the users' or customers point of view, where they want to ensure the privacy of its input parameters and results. Another is to cloud servers point of view, where cloud entity is worried about feasibleness of encrypted/transformed inputs and operating on them. In computational outsourcing, users are not participating in the computational function, rather than they outsource the private problem along with parameters

to the cloud, but users and cloud servers are not mutually trusted entities. Thus, users would not like to submit their private problem data inputs to the cloud. Thus, encrypting/transforming the private data prior to submission to cloud is a usual solution. Our contribution in this paper is as -

- We have proposed protocol for secure and an efficient computational outsourcing to cloud. The protocol is completely non-interactive between users.
- We have performed the computational security analysis for our proposed system.

2.3 Organization of the Paper

Remaining paper organized as - Preliminaries are given in Sect. 3. Secure outsourcing using FHE scheme is given in Sect. 4. Experimental results are presented in Sect. 5. Section 6 presents our proposed scheme along with correctness, security analysis and our experimental simulation results. Section 7 concludes the paper.

3 Preliminaries

This section discusses some of the significant preliminaries required for secure computational outsourcing.

3.1 Computational Verifiability

Various different solutions exist for secure computational outsourcing. Homomorphic encryption(HE) can be assumed as a better solution to secure outsourcing of scientific computations, but it is useful when the returned result can be trusted.

Lemma 1. *It is infeasible to factorizing the N in polynomial time if integer factorization in large scale in infeasible.*

Proof. Assume x is an adversary who is able to factorize a number N into primes p and q of probable same bit length in polynomial time. Suppose this operations probability as p'. Each factor $fact_i$ of a number N will at least posses two prime factors. So the probability p''_r that the attacker can factorize it is - almost lesser than p'. Thus the resultant probability that attacker can factorize N is $\prod_{i=1}^{m} p''_r \leq (p')^m$. Now if p' is negligible, the resultant probability is also negligible.

3.2 Lattice-Based Encryption

As we know that the computational complexity as well as the input parameters' privacy is mostly dependant on the encryption procedure adopted by user. Lattice-Based Encryption [16,28] is considered as secure against quantum

computer attacks and much efficient as well as potent than RSA and Elliptic curve cryptosystems.

Lattice based cryptosystem, whose security is based on core lattice theory problems, was introduced by Mikls Ajtai, in 1996. In the same year, first lattice based public key encryption scheme (NTRU) was proposed. Later, much work and improvement [2] was carried out towards this direction involving some additional cryptographic primitives LWE(learning with errors).

4 Secure Outsourcing Using FHE

This section summarizes the scheme [26] for secure outsourcing of large matrix multiplication computations on cloud. The complete description and steps involved in this scheme are summarized as below:-

Algorithm 1

1: Generate secret key pair: $\{H, Y\}$
where, H: is a Hadamard matrix [23] and
Y: is a diagonal matrix selected randomly.
2: Consider, M_1 and M_2 are two large matrices, for which the multiplication needs to be computed, thus client will outsource this computation problem to cloud side.
3: Client computes,

$$M_1' = H \times M_1 \times Y$$
$$M_2' = Y^{-1} \times M_2$$

4: Client sends M_1' and M_2' to cloud server.
5: $Result' \leftarrow M_1' \times M_2'$
6: The cloud server sends back the computed result to client side.
7: After getting the computed result, client will retransform it and get the original result for MM problem. The procedure is given as below Algorithm -
8: $Result \leftarrow H^{-1} \times Result'$

5 Experiment Results

This section presents our experimental analysis.

5.1 System Specifications

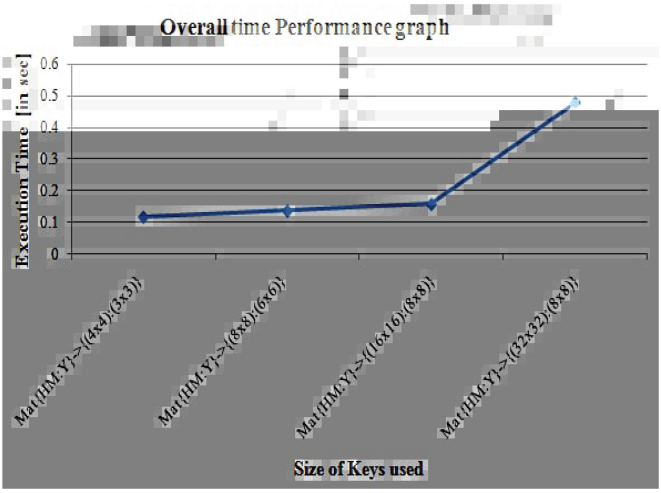
Our system specifications are as below:-

– *Software Specifications* -
OS - Ubuntu 16.04 LTS, 64 bit
Python version - 'Python 3.6.0'

– *Hardware Specifications* -
RAM size - 4 GB
Processor - Intel core i3 4030U CPU @1.90$GHz \times 4$

5.2 Our Results

The graph for overall algorithm execution for various sized input parameters is given below:-



The end results of execution performance for varying key sizes is presented as Table 1 below:-

Table 1. Execution performance

S.No	Dimensions				Exec Performance		
	HM	M1	M2	Y	T[encry](in sec)	T[dec](in sec)	T[Overall](in sec)
1	4×4	4×3	3×4	3×3	0.0994174	0.115151	0.1187498
2	8×8	8×6	6×4	6×6	0.1260472	0.1349868	0.1377818
3	16×16	16×8	8×8	8×8	0.1321644	0.1473488	0.1589264
4	32×32	32×8	8×8	8×8	0.165747	0.146771	0.4791004
Tabular form							

6 Proposed Scheme

In this section, we have proposed an efficient secure computational outsourcing mechanism applicable for multi-users. The system model and proposed mechanism steps are given in subsections below:-

6.1 System Model

The proposed system model is represented as diagram below (Fig. 3):- Notations used are given in Table 2.

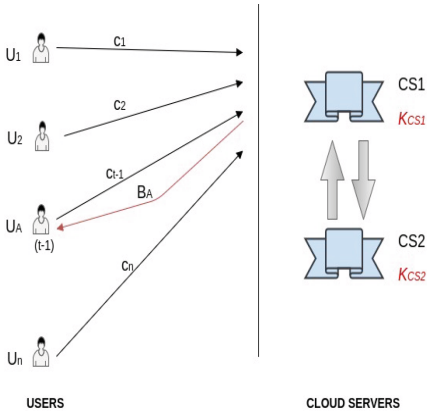


Fig. 3. Proposed model

Table 2. Notations used in proposed system

$CS1$:	First cloud server
$CS2$:	Second cloud server
c_i:	Ciphertexts (encrypted data of each customer/user U_i)
n:	No. of users
α_i:	Private input corresponding to U_i
ψ:	Probability density function
q:	Prime order
$RAND_i$:	Random number for i^{th} user
$\mathscr{C}_{FUN}$:	Function circuit
R:	Ring structure space
β:	Final computed result

6.2 Protocol Steps

The proposed secure computational outsourcing protocol executes in the below phases -

Key Gen() and Set up -

- Perform sampling for ring element space vector $a_i \leftarrow R_q^N$, $\forall\ i = (1, 2, \cdots, n)$; Ring element $SK_i \leftarrow \psi$; $\alpha_i \leftarrow \psi^N$ (ψ represents: probability density function), where -

$$\psi = \int_{-\infty}^{x} P(\xi) d\xi$$

- Key pairs of U_i: Public key - $(a_i.SK_i + 2\alpha_i) \in R_q^N$; Private key - SK_i.
- $CS1$ has its private no. as K_{CS1} & $CS2$ has its private no. as K_{CS2}.
- U_i shares a random no. $RAND_i$ with $CS1$.
- Each user U_i initiates protocol and sends $RAND_i.SK_i$ to $CS2$.
- $CS2$ reckons $K_{CS2}.RAND_i.SK_i$ and sends back to $CS1$.
- $CS1$ can get $K_{CS2}.SK_i$ by extracting $RAND_i$.

$\forall i \in (1, 2, \cdots, n)$,
U_i uses Lattice based encryption method to encrypt its own problem input α_i. The sub-steps involved in this are as below:-

Lattice based Encryption -

- First U_i perform sampling as: $e_i \leftarrow \psi^N$. where,
  ψ is: probability density function(PDF), defined as -

$$\psi = \int_{-\infty}^{x} P(\xi) d\xi$$

- Next, each user U_i computes -

$$c_0^i \leftarrow < u_i, e_i > + \alpha_i \in R_q$$
$$c_1^i \leftarrow < a_i, e_i > \in R_q$$

- Further, it gives output as ciphertext,

$$c_i = (c_0^i, c_1^i) \in R_q^N;\ (N = 2)$$

$CS1$ stores all ciphertexts coming from user $U_i (1 \leq i \leq n)$, then further steps are as below:-

Circuit Computation on Outsourcing -

- First, $CS1$ transforms the ciphertexts as $c_i \rightarrow c_i^{TR_1}$
  where, $c_i^{TR_1} = (c_0^{iTR_1}, c_1^{iTR_1}) = (K_{CS1}.c_0^i, K_{CS1}.(K_{CS2}.SK_i).c_1^i)$.
- $CS1$ sends above $c_i^{TR_1}$ to $CS2$.
- After receiving $c_i^{TR_1}$, $CS2$ again transforms $c_i^{TR_1}$ into

$$c_i^{TR_2} = (K_{CS2}.K_{CS1}.c_0^i, K_{CS1}.(K_{CS2}.SK_i).c_1^i)$$

  take, $K = K_{CS1}.K_{CS2}$
  then, $c_i^{TR_2} = (c_0^{iTR_2}, c_1^{iTR_2}) = (K.c_0^i, K.SK_i.c_1^i)$
- $CS2$ then reckons the ciphertext of result by transformed ciphertext of every user's private i/p.
  - Additive oprn. for each add. gate -
    $\Rightarrow c_i^{TR_2} \bigoplus c_j^{TR_2}$
    $\Rightarrow (c_1^{iTR_2} - c_0^{iTR_2}) \bigoplus (c_1^{jTR_2} - c_0^{jTR_2})$
    $\Rightarrow (K.SK_i.c_1^i - K.c_0^i) \bigoplus (K.SK_j.c_1^j - K.c_0^j)$
    $\Rightarrow (K.(SK_i.c_1^i - c_0^i) \bigoplus K.(SK_j.c_1^j - c_0^j))$
    $\Rightarrow K.[(SK_i.c_1^i - c_0^i) \bigoplus (SK_j.c_1^j - c_0^j)]$
    $\Rightarrow K.[\alpha_i + \alpha_j]$
  - Multiplicative oprn. for every mul. gate -
    $\Rightarrow c_i^{TR_2} \bigotimes c_j^{TR_2}$
    $\Rightarrow (c_1^{iTR_2} - c_0^{iTR_2}) \bigotimes (c_1^{jTR_2} - c_0^{jTR_2})$
    $\Rightarrow (K.SK_i.c_1^i - K.c_0^i) \bigotimes (K.SK_j.c_1^j - K.c_0^j)$
    $\Rightarrow (K.(SK_i.c_1^i - c_0^i) \bigotimes K.(SK_j.c_1^j - c_0^j))$
    $\Rightarrow K^2.[(SK_i.c_1^i - c_0^i) \bigotimes (SK_j.c_1^j - c_0^j)]$
    $\Rightarrow K^2.[\alpha_i \times \alpha_j]$

Production of the result by cloud servers will follow as steps below:-

Production of Result -

- When $CS2$ performed gate by gate computation on circuit $\mathscr{C}_{FUN}$, it gets some intermediate meta result, which is encrypted by K_{CS1} and K_{CS2} of the cloud servers $CS1$ and $CS2$.
  If $\beta = FUN(\alpha_1, \alpha_2, \cdots, \alpha_n)$ and let's θ is the no. of multiplicative gates of $\mathscr{C}_{FUN}$.
  then, $\beta' = K^{\theta+1}.\beta = (K_{CS1}^{\theta+1}.K_{CS2}^{\theta+1}).\beta$
- To provide results for each user, and ensure that only authorized user set must get final result [*Assume,* $U_{\mathbb{A}}$, $\mathbb{A} \in (1, 2, 3, \cdots, n)$ *is authorized user set to access result*], $CS2$ first sends β' to $CS1$.
- $CS1$ removes $K_{CS1}^{\theta+1}$ and ties $RAND_{\mathbb{A}}$ to compute $\beta'_{\mathbb{A}} = RAND_{\mathbb{A}}.K_{CS2}^{\theta+1}.\beta$
- Then $CS1$ sends $\beta'_{\mathbb{A}}$ to $CS2$.
- $CS2$ finally removes $K_{CS2}^{\theta+1}$ and gets $\beta_{\mathbb{A}} = RAND_{\mathbb{A}}.\beta$
- Further $CS2$ sends it to authorized users set $U_{\mathbb{A}}$, $\mathbb{A} \in (1, 2, 3, \cdots, n)$.

Secure Results Reconstruction at Users' side -

- For each $U_{\mathbb{A}}$, $\mathbb{A} \in (1, 2, 3, \cdots, n)$, it successfully gets the final result β by deposing $RAND_{\mathbb{A}}$.

6.3 Analysis of Proposed Scheme

Here, we have presented the correctness and security analysis of our proposed scheme.

- Correctness analysis -

The correctness analysis of given scheme is as follows:-

Theorem 1. *Due to Homomorphic properties of the transformed ciphertexts, the given scheme is correct.*

Let, P and Q are rings a function $f : P \rightarrow Q$ will be ring homomorphism if $\forall x_1, x_2 \in P$.

- $f(x_1 + x_2) = f(x_1) + f(x_2)$
- $f(x_1 * x_2) = f(x_1) * f(x_2)$

- Security analysis -

The security analysis of proposed scheme can be analysed as below:-

Theorem 2. *As long as Lattice based encryption is secure and cloud servers CS1 and CS2 are noncolluding, the given protocol is secure enough.*

In proposed protocol, each user U_i encrypts its private input α_i with the help of its own public key, which is being produced by triggering lattice based encryption scheme. Further, U_i sends $RAND_i.SK_i$ to $CS2$. Then, $CS2$ reckons $K_{CS2}.RAND_i.SK_i$ and sends back to $CS1$. Here, $U_i's$ private key is SK_i, which is protected by $RAND_i$. In the entire process, the user's private keys are not being revealed.

After transferring computed results, cloud ensures in the protocol that only authorized user set must get final result; (Assume, $U_{\mathbb{A}}$, $\mathbb{A} \in (1, 2, 3, \cdots, n)$ is authorized user set to access result.)

6.4 Comparative Analysis

This section presents the comparison of our scheme with existing schemes on several factors/parameters. The representation is given in Table 3.

Table 3. Comparison with related work

Schemes	Feasible data size	Encry() technique adopted	Download result and decry()	Users	Speed-up	Cloud-Efficiency
Wang et al. (2015)	Low and medium sized	Parameters transformation	Slow on large size data	Single user	Good for medium sized problem	Moderate
Li et al. (2015)	Medium sized	Identity based encryption	Slow on large size data	Single user	Good upto medium sized problem	Moderate
Our construction	Medium to large sized	Lattice based encryption	Comparatively faster	Multi user supported	Better for large sized problem	Good

7 Conclusion and Future Work

When users have to compute some complex function, which involves their private inputs then to perform outsourcing is the possible scenario from user side. There exist hurdles in following two aspects - One is in the users' or customers point of view, where they want to ensure the privacy of its input parameters and results. Another is to cloud servers point of view, where cloud entity is worried about feasibleness of encrypted/transformed inputs and operating on them.

In this paper, we have constructed a scheme for secure outsourcing based on multi cloud servers. The computational complexity and security analysis is also given for our proposed system. Finding an efficient, practical and computationally secure outsourcing solution for various specific scientific problems will be our further research work.

References

1. Goyal, V., Pandey, O., Sahai, A., Waters, B.: Attribute-based encryption for fine-grained access control of encrypted data. In: Proceedings of 13th ACM Conference on Computer and Communications Security, pp. 89–98 (2006)
2. Gentry, C.: A fully homomorphic encryption scheme [Doctoral dissertation], Stanford University (2009)
3. Cheung, L., Newport, C.: Provably secure ciphertext policy ABE. In: Proceedings of 14th ACM Conference on CCS, pp. 456–465 (2007)
4. Nishide, T., Yoneyama, K., Ohta, K.: Attribute-based encryption with partially hidden encryptor-specified access structures. In: Bellovin, S.M., Gennaro, R., Keromytis, A., Yung, M. (eds.) ACNS 2008. LNCS, vol. 5037, pp. 111–129. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-68914-0_7
5. Han, F., Qin, J., Zhao, H., Hu, J.: A general transformation from KP-ABE to searchable encryption. Future Gen. Comput. Syst. **30**, 107–115 (2014)
6. Zhao, H., Qin, J., Hu, J.: Energy efficient key management scheme for body sensor networks. IEEE Trans. Parallel Distrib. Syst. **24**(11), 2202–2210 (2013)

7. Asharov, G., Jain, A., López-Alt, A., Tromer, E., Vaikuntanathan, V., Wichs, D.: Multiparty computation with low communication, computation and interaction via threshold FHE. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. LNCS, vol. 7237, pp. 483–501. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-29011-4_29
8. Atallah, M.J., Pantazopoulos, K., Rice, J.R., Spafford, E.E.: Secure outsourcing of scientific computations. In: Zelkowitz, M.V. (ed.) Trends in Software Engineering, vol. 54, pp. 215–272. Elsevier, Amsterdam (2002)
9. Atallah, M.J., Li, J.: Secure outsourcing of sequence comparisons. Intl. J. Inf. Secur. **4**(4), 277–287 (2005)
10. Benjamin, D., Atallah, M.J.: Private and cheating-free outsourcing of algebraic computations. In: Proceedings of 6th Annual Conference on PST, pp. 240–245 (2008)
11. Atallah, M.J., Frikken, K.B.: Securely outsourcing linear algebra computations. In: Proceedings of 5th ACM Symposium on ASIACCS, pp. 48–59 (2010)
12. Wang, C., Ren, K., Wang, J.: Secure and practical outsourcing of linear programming in Cloud Computing. In: Proceedings of IEEE INFOCOM, pp. 820–828 (2011)
13. Halevi, S., Lindell, Y., Pinkas, B.: Secure computation on the web: computing without simultaneous interaction. In: Rogaway, P. (ed.) CRYPTO 2011. LNCS, vol. 6841, pp. 132–150. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-22792-9_8
14. Green, M., Hohenberger, S., Waters, B.: Outsourcing the decryption of ABE ciphertexts. In: Proceedings of 20th USENIX Conference on SEC, p. 34 (2011)
15. Brakerski, Z., Vaikuntanathan, V.: Efficient fully homomorphic encryption from (standard) LWE. In: Proceedings of the IEEE 52nd Annual Symposium on Foundations of Computer Science (FOCS 2011), pp. 97–106 (2011)
16. López-Alt, A., Tromer, E., Vaikuntanathan, V.: Cloud-assisted multiparty computation from fully homomorphic encryption. IACR Cryptology ePrint Archive, vol. 2011, Article 663 (2011)
17. Lai, J., Deng, R., Guan, C., Weng, J.: Attribute-based encryption with verifiable outsourced decryption. IEEE Trans. Inf. Forensics Secur. **8**(8), 1343–1354 (2013)
18. Zhang, Y., Blanton, M.: Efficient secure and verifiable outsourcing of matrix multiplications. In: Chow, S.S.M., Camenisch, J., Hui, L.C.K., Yiu, S.M. (eds.) ISC 2014. LNCS, vol. 8783, pp. 158–178. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-13257-0_10
19. Atallah, M.J., Frikken, K.B.: Securely outsourcing linear algebra computations. In: ASLACCS, 13–16 April 2010, Beijing, China (2010)
20. Lei, X., Liao, X., Huang, T., Li, H., Hu, C.: Outsourcing large matrix inversion computation to a public cloud. IEEE Trans. Cloud Comput. **1**(1), 1 (2013)
21. Benjamin, D., Atallah, M.J.: Private and cheating-free outsourcing of algebraic computations. In: Sixth Annual Conference on Privacy, Security and Trust, PST 2008. IEEE (2008)
22. Xiang, C., Tang, C.: Securely verifiable outsourcing schemes of matrix calculation. Int. J. High Perform. Comput. Netw. **8**(2), 93–101 (2015)
23. http://homepages.math.uic.edu/leon/mcs425-s08/handouts/Hadamard_codes.pdf
24. Sahai, A., Waters, B.: Fuzzy identity-based encryption. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 457–473. Springer, Heidelberg (2005). https://doi.org/10.1007/11426639_27

25. Zeng, D., Guo, S., Hu, J.: Reliable bulk-data dissemination in delay tolerant networks. IEEE Trans. Parallel Distrib. Syst. doi.ieeecomputersociety.org/10.1109-TPDS.2013.221
26. Sudarshan, V., Satyanarayana, N.: An efficient protocol for secure outsourcing of scientific computations to an untrusted cloud. In: International Conference on Intelligent Computing and Control (I2C2), Karpagam College of Engineering, Tamilnadu (2017)
27. Sudarshan, V., Satyanarayana, N., Dileep Kumar, A.: Lock-in to the meta cloud with attribute based encryption without outsourced decryption. IJCST **5**(4) (2014)
28. Brakerski, Z., Vaikuntanathan, V.: Fully homomorphic encryption from ring-LWE and security for key dependent messages. In: Rogaway, P. (ed.) CRYPTO 2011. LNCS, vol. 6841, pp. 505–524. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-22792-9_29