On Multi-user based Efficient Computation Outsourcing Scheme and its Application to Cloud

V. Sudarsan Rao[1] and N. Satyanarayana[2]
(Corresponding author: V. Sudarsan Rao)
Department of CSE, Khammam Institute of Technology and Sciences (KITS)[1]
Telangana, India
Department of CSE, Nagole Institute of Technology and Sciences (NITS)[2]
Telangana, India
(Email: sudharshan.cse2008@gmail.com)
(Received Oct. 9, 2017; revised and accepted Apr. 13, 2018)

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, a new kind of outsourcing computing protocol is proposed which utilizes multi cloud servers view framework. This paper mainly adopts the Fully Homomorphic Encryption technique (FHE). 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; Circuit Computation; Cloud Computing; Privacy; Secret Information Parameters; Secure Outsourcing

1 Introduction

Beside the tremendous advantages of outsourcing, client faces some challenges by outsourcing the computational task to cloud [2, 3]. 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 [6]. 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,

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

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

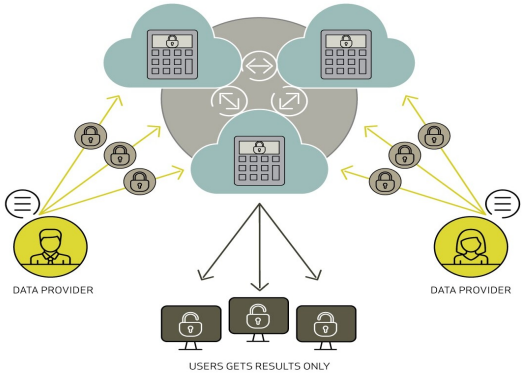

Figure 1: General computational outsourcing scenario

Recently, as the development of cloud computing [33], 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 [31], which is maintained and operated by untrusted cloud service provider (CSP) [21, 29]. 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.

Remaining paper organized as - Section 2 provides a general nomenclature for various secure outsourcing algorithms. Significant state-of-the-art protocols along with the motivation towards the problem and our contribution in this paper is summarized. Preliminaries are given in Section 3. Secure outsourcing using FHE scheme is given in Section 4. Experimental analysis are presented in Section 5. Section 6 presents our proposed scheme along

with correctness, security analysis and our experimental simulation results. Section 7 concludes the paper.

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 Figure 2.

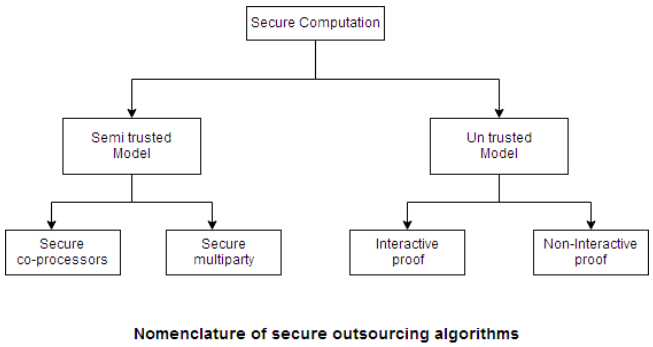

Figure 2: Secure outsourcing algorithms nomenclature

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 [16], 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 by [1], authors have 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.* [18] in 2011, that was a non-interactive method for secure outsourcing [8]. [23] 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 [25], was firstly dealt with by Goyal *et al.* [15]. Two distinct and interrelated notions of ABE were determined in [15]. Accordingly, several constructions supporting for any kinds of access structures were provided [13,24] for practical applications [19, 32]. Atallah *et al.* [2] 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 [3] given an efficient protocol to outsource sequence comparison with two servers in secure manner. Furthermore, Benjamin and Atallah [6] addressed the problem of secure outsourcing for widely applicable linear algebraic computations. Atallah and Frikken [4] further studied this problem and gave improved protocols based on the so-called weak secret hiding assumption. Recently, Wang *et al.* [28] presented efficient mechanisms for secure outsourcing of linear programming computation.

In [17], 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, V.Sudarshan *et al.* [27] proposed an Attribute-Based Encryption mechanism, applied for cloud security. Recently Lai *et al.* [20] 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. A.K.Chattopadhyay *et al.* [12] proposed the scheme which uses simple Boolean based encryption and decryption of the data files, which is low in computational cost. Yongjian L. *et al.* [22] given the ABE based construction of scheme where some invalid ciphertexts checking is performed in the decryption algorithm. Kai F. *et al.* [14] designed an efficient user and attribute revocation method. Along with this, analysis and simulation results for their scheme showed that it is both secure and highly efficient. Z. Cao *et al.* [11] remarked that Yu *et al.*'s scheme [30] has two inherent weaknesses:

1) It does not truly mitigate the client's computational burden for key updates.

2) It does not ensure confidentiality since the files uploaded to the cloud by the client are eventually not encrypted at all.

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.

3 Preliminaries

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

3.1 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 [9,10] 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 Miklos Ajtai, in 1996. In the same year, first lattice based public key encryption scheme (NTRU) was proposed. Later, much work and improvement [16] was carried out towards this direction involving some additional cryptographic primitives LWE (learning with errors).

3.2 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. □

Definition 1. *A matrix $M \in R^{n,n}$ can be called as orthogonal if it is satisfying one of the equivalent conditions*

1) $M.M^T = M^T.M = I_n$;

2) M is invertible and $M^{-1} = M^T$.

4 Secure Outsourcing Using FHE

This section summarizes Sudarshan *et al.* scheme [26] for secure outsourcing of large matrix multiplication computations on cloud. First, the key space is being generated at client side, which will be utilized in further steps. Here, we have considered the scientific computation as 'large matrix multiplication', which the client needs to outsource to cloud server. Here the assumption taken is that the third party or cloud server is untrusted. Client needs to perform problem transformation step for secure outsourcing. Further, the computation inside cloud is performed. After getting the computed result, client will retransform it and get the original result for matrix multiplication problem. The complete description and steps involved in this scheme are summarized as below:

Algorithm 1 Secure Outsourcing using FHE

1: Begin
2: Generate secret key pair: $\{H, Y\}$
where, H: is a Hadamard matrix [34] and
Y: is a diagonal matrix selected randomly.
3: 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.
4: Client computes,
$$M'_1 = H \times M_1 \times Y$$
$$M'_2 = Y^{-1} \times M_2$$
5: Client sends M'_1 and M'_2 to cloud server.
6: $Result' \leftarrow M'_1 \times M'_2$
7: The cloud server sends back the computed result to client side.
8: After getting the computed result, client will retransform it and get the original result for MM problem. The procedure is given as below Algorithm
9: $Result \leftarrow H^{-1} \times Result'$
10: End

5 Experimental Analysis

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

We performed the experiments on varying sized secret key pair, arbitrary large sized matrices M_1 and M_2 as input. Problem parameters transformation/encryption, decryption and entire average execution time for executing the protocol is analysed. The graph for encryption phase for various sized input parameters is Figure 1.

The graph for decryption phase for various sized input parameters is in Figure 2.

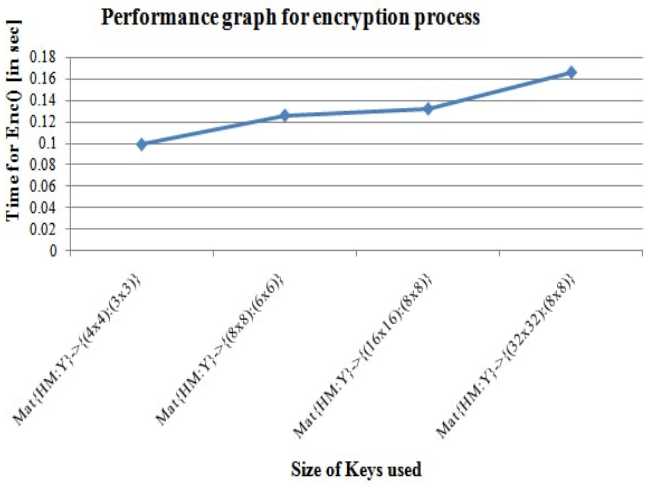

Figure 1: The performance for encryption phase

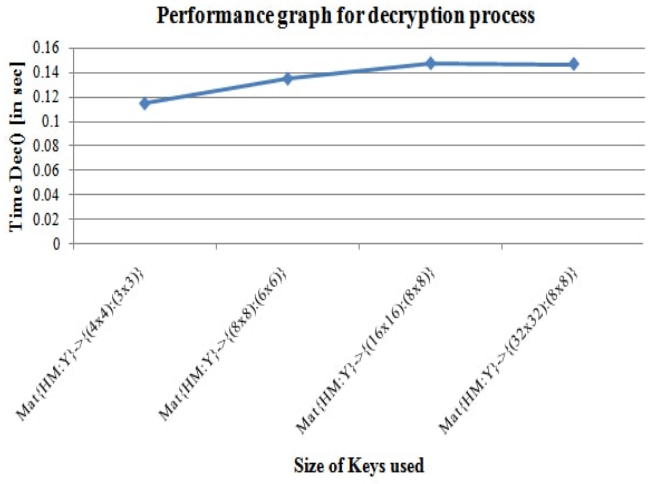

Figure 2: The performance for decryption phase

The graph for overall algorithm execution for various sized input parameters is in Figure 3.

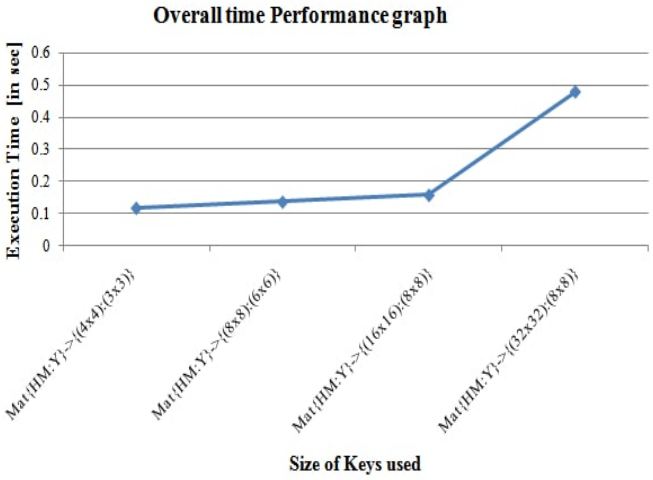

Figure 3: The performance for overall algorithm

In above Performance graphs 1-3, the encryption, decryption and overall execution time (in seconds) for varying experimental instances of secret key matrix pair size dimensions is shown. The end results of execution performance for varying key sizes is presented as Table 1.

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.	4x4	4x3	3x4	3x3	0.0994174	0.115151	0.1187498
2.	8x8	8x6	6x4	6x6	0.1260472	0.1349868	0.1377818
3.	16x16	16x8	8x8	8x8	0.1321644	0.1473488	0.1589264
4.	32x32	32x8	8x8	8x8	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 in Figure 4.

Notations used are given in Table 2.

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

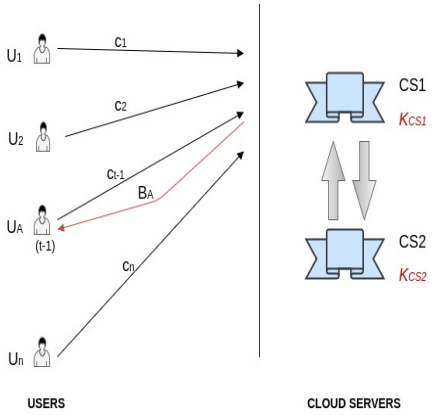

Figure 4: Proposed model

6.2 Protocol Steps

The proposed secure computational outsourcing protocol executes in the below phases

Algorithm 2 Key Gen() and Set up

1: Begin
2: 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
$$\psi = \int_{-\infty}^{x} P(\xi) d\xi$$
3: Key pairs of U_i: Public key $(a_i.SK_i + 2\alpha_i) \in R_q^N$; Private key SK_i.
4: $CS1$ has its private no. as K_{CS1} & $CS2$ has its private no. as K_{CS2}.
5: U_i shares a random no. $RAND_i$ with $CS1$.
6: Each user U_i initiates protocol and sends $RAND_i.SK_i$ to $CS2$.
7: $CS2$ reckons $K_{CS2}.RAND_i.SK_i$ and sends back to $CS1$.
8: $CS1$ can get $K_{CS2}.SK_i$ by extracting $RAND_i$.
9: End

$\forall i \in (1, 2, \cdots, n)$, U_i uses Lattice based encryption method to encrypt its own problem input α_i. The sub-steps involved in this are as Algorithm-3.

Algorithm 3 Lattice based Encryption

1: Begin
2: First U_i perform sampling as: $e_i \leftarrow \psi^N$. where, ψ is: probability density function(PDF), defined as
$$\psi = \int_{-\infty}^{x} P(\xi) d\xi$$
3: Next, each user U_i computes
$$c_0^i \leftarrow < u_i, e_i > + \alpha_i \in R_q$$
$$c_1^i \leftarrow < a_i, e_i > \in R_q$$
4: Further, it gives output as ciphertext,
$$c_i = (c_0^i, c_1^i) \in R_q^N;\ (N = 2)$$
5: End

$CS1$ stores all ciphertexts coming from user $U_i (1 \leq i \leq n)$, then further steps are given in Algorithm-4.

Production of the result by cloud servers will follow as steps of Algorithm-5.

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 non-colluding, the given protocol is secure enough.*

Algorithm 4 Circuit Computation on Outsourcing

1: Begin
2: First, $CS1$ transforms the ciphertexts as $c_i \rightarrow c_i^{TR_1}$ where, $c_i^{TR_1} = (c_0^{i_{TR_1}}, c_1^{i_{TR_1}}) = (K_{CS1}.c_0^i, K_{CS1}.(K_{CS2}.SK_i).c_1^i)$.
3: $CS1$ sends above $c_i^{TR_1}$ to $CS2$.
4: 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, $c_i^{TR_2} = (c_0^{i_{TR_2}}, c_1^{i_{TR_2}}) = (K.c_0^i, K.SK_i.c_1^i)$
5: $CS2$ then reckons the ciphertext of result by transformed ciphertext of every user's private i/p.
6: Additive oprn. for each add. gate
$\Rightarrow c_i^{TR_2} \bigoplus c_j^{TR_2}$
$\Rightarrow (c_1^{i_{TR_2}} - c_0^{i_{TR_2}}) \bigoplus (c_1^{j_{TR_2}} - c_0^{j_{TR_2}})$
$\Rightarrow (K.SK_i.c_1^i - K.c_0^i) \bigoplus (K.SK_j.c_1^j - K.c_0^j)$
$\Rightarrow (K.(SK_i.c_1^i - c_0^i) \bigoplus K.(SK_j.c_1^j - c_0^j))$
$\Rightarrow K.[(SK_i.c_1^i - c_0^i) \bigoplus (SK_j.c_1^j - c_0^j)]$
$\Rightarrow K.[\alpha_i + \alpha_j]$
7: Multiplicative oprn. for every mul. gate -
$\Rightarrow c_i^{TR_2} \bigotimes c_j^{TR_2}$
$\Rightarrow (c_1^{i_{TR_2}} - c_0^{i_{TR_2}}) \bigotimes (c_1^{j_{TR_2}} - c_0^{j_{TR_2}})$
$\Rightarrow (K.SK_i.c_1^i - K.c_0^i) \bigotimes (K.SK_j.c_1^j - K.c_0^j)$
$\Rightarrow (K.(SK_i.c_1^i - c_0^i) \bigotimes K.(SK_j.c_1^j - c_0^j))$
$\Rightarrow K^2.[(SK_i.c_1^i - c_0^i) \bigotimes (SK_j.c_1^j - c_0^j)]$
$\Rightarrow K^2.[\alpha_i \times \alpha_j]$
8: End

Algorithm 5 Production of Result

1: Begin
2: 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, $\beta' = K^{\theta+1}.\beta = (K_{CS1}^{\theta+1}.K_{CS2}^{\theta+1}).\beta$
3: 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 $CS1$.
4: $CS1$ removes $K_{CS1}^{\theta+1}$ and ties $RAND_{\mathbb{A}}$ to compute $\beta'_{\mathbb{A}}$ $= RAND_{\mathbb{A}}.K_{CS2}^{\theta+1}.\beta$
5: Then $CS1$ sends $\beta'_{\mathbb{A}}$ to $CS2$.
6: $CS2$ finally removes $K_{CS2}^{\theta+1}$ and gets $\beta_{\mathbb{A}} = RAND_{\mathbb{A}}.\beta$
7: Further $CS2$ sends it to authorized users set $U_{\mathbb{A}}$, $\mathbb{A} \in (1, 2, 3, \cdots, n)$.
8: End

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 transfering 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.)

Algorithm 6 Secure Results Reconstruction at Users' side

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

6.4 Experimental Simulation Results

This section presents the simulation results on virtual cloud server. We used *Microsoft Azure cloud server* along with *Android Studio 3.0.1 software platform SDK* to perform simulation on client-cloud outsourcing model. *JDK 9.0.1* is used for backend interface development. The machine used for simulation has specifications as *OS: Windows 10*; *RAM size: 16 GB*; *NVIDIA GeForce GTX 1080 GPU* with *octa core i7 CPU*.

The experimental simulation results obtained are given in Table 3.

In Table 3, we tested our scheme through various client's private data size and key size parameters etc. The client's private data can be in any form like integer, image, text, pdf file, video file etc. We obtained the time taken for encrypt private data, the time taken for upload operation on Azure cloud server and time taken to download and decrypt data on client side. Inside cloud scientific computational operations are being performed on transformed parameters. The overall representation is given in Figure 5.

6.5 Comparative Analysis

This section presents the comparison of our scheme with existing schemes on several factors/parameters. The representation is given in Table 4.

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.

Table 3: Experimental simulation

Client data size	Key size	$T.T._{Encry()}$	$T.T._{Upload-on-Cloud-Server}$	$T.T._{Download-res-and-Decry()}$
500 kB	16 bits	2 sec.	2 sec.	2.4 sec.
1 mB	32 bits	2.6 sec.	2.4 sec.	2.8 sec.
2 mB	32 bits	3 sec.	3.2 sec.	3.2 sec.
50 mB	64 bits	6 sec.	6.4 sec.	5.8 sec.
100 mB	64 bits	9.4 sec.	8.8 sec.	9 sec.
500 mB	128 bits	19 sec.	19.2 sec.	19 sec.
1 GB	128 bits	32.2 sec.	32 sec.	32.4 sec.

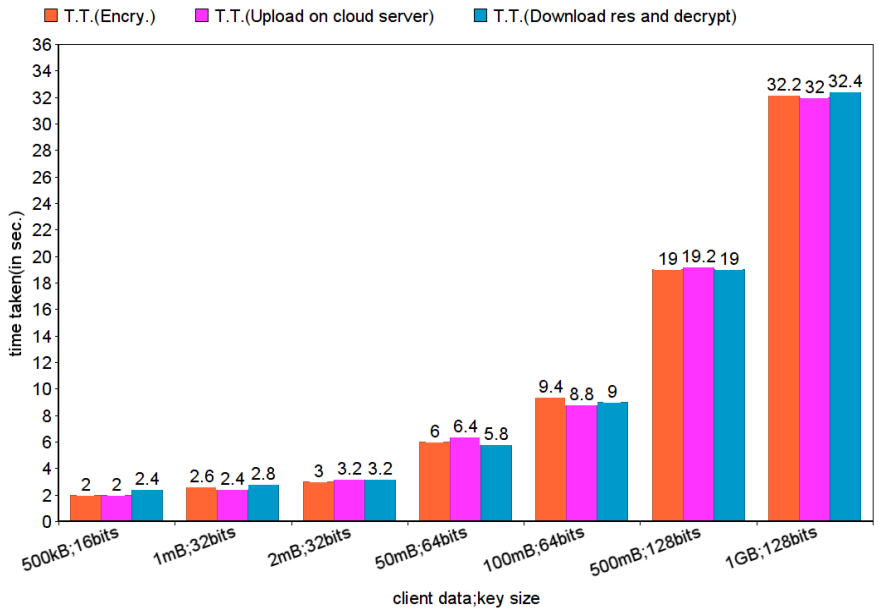

Figure 5: The overall performance

Table 4: Comparison with related work

Schemes	**Feasible data size**	**Encry() technique adopted**	**Download result and decry()**	**Users**	**Speed-up**	**Cloud - Efficiency**
C. Wang et.al. (2015)	Low and medium sized	Parameters transformation	Slow on large size data	Single user	Good for medium sized problem	Moderate
Jin 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

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] G. Asharov, A. Jain, A. Lopez-Alt, E. Tromer, V. Vaikuntanathan, and D. Wichs, "Multiparty computation with low communication, computation and interaction via threshold fhe," in *Advances in Cryptology (EUROCRYPT'12)*, pp. 483-501, 2012.

[2] M. J. Atallah, K. Pantazopoulos, J. R. Rice, and E. E. Spafford, "Secure outsourcing of scientific computations," in *Trends in Software Engineering*, vol. 54, pp. 215-272, 2002.

[3] M. J. Atallah and J. Li, "Secure outsourcing of sequence comparisons," *Internatinal Journal Informations Security*, vol. 4, no. 4, pp. 277-287, Oct. 2005.

[4] M. J. Atallah and K. B. Frikken, "Securely outsourcing linear algebra computations," in *Proceedings 5th ACM Symposium*, pp. 48-59, 2010.

[5] M. J. Atallah and K. B. Frikken, "Securely outsourcing linear algebra computations", in *Proceedings 5th ACM Symposium*, pp. 48-59, 2010.

[6] D. Benjamin and M. J. Atallah, "Private and cheating-free outsourcing of algebraic computations," in *Proceedings 6th Annual Conference (PST'08)*, pp. 240-245, 2008.

[7] D. Benjamin, M. J. Atallah, "Private and cheating-free outsourcing of algebraic computations," in *Proceedings 6th Annual Conference (PST'08)*, pp. 240-245, 2008.

[8] Z. Brakerski and V. Vaikuntanathan, "Efficient fully homomorphic encryption from (standard) LWE," in *Proceedings of the IEEE 52nd Annual Symposium on Foundations of Computer Science (FOCS'11)*, pp. 97-106, 2011.

[9] Z. Brakerski and V. Vaikuntanathan, "Fully homomorphic encryption from ring-lwe and security for key dependent messages," in *Advances in Cryptology (CRYPTO'11)*, vol. 6841, pp. 505-524, 2011.

[10] Z. Brakerski and V. Vaikuntanathan, "Fully homomorphic encryption from ring-lwe and security for key dependent messages," in Advances in Cryptology (CRYPTO'11), vol. 6841, pp. 505-524, 2011.

[11] Z. Cao, L. Liu, O. Markowitch, "Analysis of one scheme for enabling cloud storage auditing with verifiable outsourcing of key updates", *International Journal of Network Security*, vol. 19, no. 6, pp. 950-954, Nov. 2017.

[12] A. K. Chattopadhyay, A. Nag and K. Majumder, "Secure data outsourcing on cloud using secret sharing scheme", *International Journal of Network Security*, vol.19, no.6, pp. 912-921, Nov. 2017.

[13] L. Cheung and C. Newport, "Provably secure ciphertext policy ABE," in *Proceedings 14th ACM Conference (CCS'07)*, pp. 456-465, 2007.

[14] K. Fan, J. Wang, X. Wang, H. Li and Y. Yang, "A secure and verifiable outsourced access control scheme in fog-cloud computing", *Sensors*, vol. 17, no. 7, 2017.

[15] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in *Proceedings 13th ACM Conference Computer Communications Security*, pp. 89-98, 2006.

[16] C. Gentry, "A fully homomorphic encryption scheme," Stanford University, 2009. (`https://crypto.stanford.edu/craig/craig-thesis.pdf`)

[17] M. Green, S. Hohenberger, and B. Waters, "Outsourcing the decryption of ABE ciphertexts," in *Proceedings 20th USENIX Conference (SEC'11)*, pp. 34, 2011.

[18] S. Halevi, Y. Lindell, and B. Pinkas, "Secure computation on the web: computing without simultaneous interaction," in *Advances in Cryptology*, pp. 132-150, 2011.

[19] F. Han, J. Qin, H. Zhao, and J. Hu, "A general transformation from KP-ABE to searchable encryption," *Future Generation Computer Systems*, vol. 30, pp. 107-115, Jan. 2014.

[20] J. Lai, R. Deng, C. Guan, and J. Weng, "Attribute-based encryption with verifiable outsourced decryption," *IEEE Transactions on Information Forensics Security*, vol. 8, no. 8, pp. 1343-1354, Aug. 2013.

[21] X. Lei, X. Liao, T. Huang, H. Li, and C. Hu, "Outsourcing large matrix inversion computation to a public cloud", *IEEE Transactions on Cloud Computing*, vol. 1, no. 1, 2013.

[22] Y. Liao, Y. He, F. Li, S. Jiang, S. Zhou, "Analysis of an ABE scheme with verifiable outsourced decryption", *Sensors*, vol. 18, no. 176, 2018.

[23] A. Lopez-Alt, E. Tromer, and V. Vaikuntanathan, "Cloud-assisted multiparty computation from fully homomorphic encryption," *IACR Cryptology ePrint Archive*, vol. 2011, no. 663, 2011.

[24] T. Nishide, K. Yoneyama, and K. Ohta, "Attribute-based encryption with partially hidden encryptor-specified access structures," in *Proceedings Applications Cryptography Networks Security*, pp. 111-129, 2008.

[25] A. Sahai and B. Waters, "Fuzzy identity-based encryption," in *Proceedings Advanced Cryptology*, pp. 457-473, 2005.

[26] V. Sudarshan, N. Satyanarayana. "An efficient protocol for secure outsourcing of scientific computations to an untrusted Cloud", *International Conference on Intelligent Computing and Control (I2C2)*, 2017.

[27] V. Sudarshan, N. Satyanarayana, A. Dileep Kumar. "Lock-In to the meta cloud with attribute based encryption without outsourced decryption", *IJCST*, vol. 5, no. 4, 2014.

[28] C. Wang, K. Ren, and J. Wang, "Secure and practical outsourcing of linear programming in cloud computing," in *Proceedings of IEEE Infoaom*, pp. 820-828, 2011.

[29] C. Xiang and C. Tang, "Securely verifiable outsourcing schemes of matrix calculation", *International Journal High Performance Computing and Networking*, vol. 8, no. 2, 2015.

[30] J. Yu, K. Ren, and C. Wang, "Enabling cloud storage auditing with verifiable outsourcing of key updates," *IEEE Transactions on Information Forensics Security*, vol. 11, no. 6, pp. 1362-1375, 2016.

[31] Y. Zhang and M. Blanton, "Efficient secure and verifiable outsourcing of matrix multiplications", *Department of Computer Science and Engineering*, pp. 158-178, 2014.

[32] H. Zhao, J. Qin, and J. Hu, "Energy efficient key management scheme for body sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 11, pp. 2202-2210, Nov. 2013.

[33] D. Zeng, S. Guo, and J. Hu, "Reliable bulk-data dissemination in delay tolerant networks," *IEEE Transactions Parallel Distributed Systems*, vol. 25, no. 8, pp. 2180-2189, 2014.

[34] J. S. Leon, *Hadamard Matrices and Hadamard Codes*, Dec. 27, 2018. (`http://homepages.math.uic.edu/~leon/mcs425-s08/handouts/Hadamard_codes.pdf`)

Biography

V. Sudarsan Rao received his M.Tech in Computer Science and Engineering from JNTUH,Hyderabad , India, in 2010. Currently, he is a Researcher in the Department of Computer Science and Engineering of JNTUH, Hyderabad. His research interests include Cloud Computing and Security of provably secure symmetric encryption schemes, efficient software implementations of cryptographic primitives, pattern recognition, and Network Security.

Dr. N. Satyanarayana, M. Tech (CS), Ph.D (CSE): The Nagole Institute of Science and Technologies headed by Dr. N.Satyanarayana who possessed the highest qualification in engineering. His qualifications are M.Phil, AMIE(ET), M.Tech (CS), Ph.D(CSE), MISTE, MCSI. He obtained his Ph.D in Computer Science and had a remarkable record of merit in pursuit his engineering studies. He is young and dynamic. He has a disposition to inspire both students and staff to achieve their right goals. With his extensive and rich research experience he is able to run the institution on sound academic ground.