Secure Cloud Computing Algorithm Using Homomorphic Encryption And Multi-Party Computation

Debasis Das
Department of Computer Science and Information Systems
BITS Pilani, K.K. Birla Goa Campus,
Zuarinagar, Goa-403726, India.
Email: debasisd@goa.bits-pilani.ac.in

***Abstract*—Cloud computing is a developing technology that is yet unclear to many security issues. Data in the untrusted clouds can be encrypted using encryption algorithm. Randomizing this data provides more security which can be achieved by padding concept in the cloud. In this paper, the user's data is encrypted using padding scheme, called Optimal Asymmetric Encryption Padding (OAEP) together with Hybrid Encryption algorithm that is based on RSA (i.e., HE-RSA), in order to allow multiple parties to compute a function on their inputs while preserving Integrity and Confidentiality. The Homomorphic Encryption(HE) is performed on the encrypted data without decrypting it in computationally powerful clouds and the Secure Multi-Party Computation (SMPC) can be used in the cloud to ensure security and privacy of the users. In this paper, we have proposed a scheme that integrates the multi-party computation with homomorphic encryption to allow calculations of encrypted data without decryption. The cryptographic techniques used in our cloud model are described and the overheads are compared with Homomorphic Encryption and Multi-Party Computation.**

***Index Terms*—Cloud Computing; Optimal Asymmetric Encryption Padding; Homomorphic Encryption; Multiparty Computation.**

I. Introduction

There is a need for an appropriate or more suitable big data infrastructure [1] that supports the storage and processing on a high scale. Now a days the world is data centric, hence the big data processing and analysis have become the most important chore for any large establishment. The cloud computing is a model to provide convenient, on-demand access to share the computing resources. Organizations can simply connect to the cloud and use the available resources on the proper usage basis. The cloud computing has become a tool for analyzing big data using shared computing resources while easily handling changes in the volume and variety of the data[1]. The cloud provides many advantages such as more fault tolerance and multi-factor authentication to secure the information in the cloud. However, the cloud computing also comes with risks in maintaining the confidentiality and integrity of data due to these properties. In the last few years, there have been increasing the number of data breaches in the cloud as a result of malicious and intrusive actions. Encryption keeps the data at rest secure but data is lost if we lose the encryption key. Thus, to prevent the malicious attacks on the cloud, it is necessary to develop efficient cryptographic techniques [3], [4] which is resistant to active attacks as well as performing calculations of encrypted data without decryption. The cloud computing-based solutions have become increasingly popular in the past few years. The cloud computing platform analyzes and extracts useful information from the Big data cloud. One of the main concerns with cloud computing has been the privacy and confidentiality [5], [6], [7] of the data in the cloud. One solution is to send the data encrypted to the cloud. However, we still need to support useful computations on the encrypted data and Fully Homomorphic Encryption (FHE) [8], [9], [10], [11] is a way of supporting such computations on encrypted data. We note that while other mechanisms exist for secure computation, they generally require the different data providers to exchange information. Because FHE schemes are public key schemes, FHE is much better suited for the scenario where we have many sources of data.

The Secure Multi-Party Computation (MPC) [12], [13] guarantees that everyone learns the correct output of a joint computation but nothing else about anyone else's inputs, even when some of the user performing the computation might be actively or passively malicious. Secure MPC can be done for arbitrary computations and for any number of parties. Hence, we can view secure MPC protocols as compilers that take as input a specification of a function and output a protocol that computes the function securely. Hence, we can view secure MPC protocols as compilers that require as input a specification of a mapping and output is a protocol that computes the function securely. The Secure MPC [14], [15] offers both confidentialities as well as integrity which is much better than fully homomorphic encryption and verifiable computation. Dependable MPC can be performed for arbitrary computations and for any number of parties in Cloud Environment.
The benefits of our model:

- Integrates multi-party computation with homomorphic

978-1-5386-2290-2/18/$31.00 ©2018 IEEE

encryption

- Proposes a plan to use Optimal Asymmetric Encryption Padding(OAEP)-Homomorphic Encryption(HE)-RSA for encryption

In Section II, we describe related works. In Section III, we define the problem statement for secure cloud computing. In Section IV, we describe the Objective. In Section V, we describe the our contribution. Then, in Section V and VII, we describe the cryptographic techniques and give a detailed description of our scheme. In Section VIII, we describe the results. Finally, in Section IX, we conclude and describe goals achieved by our approach.

II. Related Work

M. Tebba et al. [2], proposed a technique to execute operations on encrypted data in the cloud which will provide us with the similar results after calculations as if we have worked directly on the raw data. Z. Wang et al. [3] gave a fresh definition of homomorphic signature for identity management in mobile cloud computing. S. Yakoubov et al. [4] conducted a survey of Cryptographic Approaches to Securing Big-Data Analytics in the Cloud. C. Rong et al. [5] conducted a survey on different security challenges in Cloud Computing. C. Gentry [6] computed arbitrary functions of encrypted data which describes a fully homomorphic encryption technique that keeps information private, but that leaves a worker that does not possess the private decryption key to compute any result of the data, even when the purpose of the data is really complex. C. Wang et al. [7] proposed an effective scheme with two salient features to ensure the correctness of the user's data in the cloud. Y. Yu et al. [8] investigated the active adversary attack in three auditing mechanisms for shared data in the cloud and also proposed a solution to remedy the weakness without sacrificing any desirable features of these mechanisms. L. Wei et al. [9] proposed a privacy cheating discouragement and secure computation auditing protocol, or SecCloud, which is a first protocol bridging secure storage and secure computation auditing in cloud. A. Lopez-Alt et al. [10] showed a new type of encryption scheme which they called multi key FHE. F. F. Moghaddam et al. [11] proposed a hybrid encryption algorithm based on RSA Small-e and Efficient-RSA for cloud computing environments.E. Shen et al. [13] proposed a scheme which is called Cryptographically Secure Computation [14] in the cloud using the concept of secure multi-party computation. This is a cryptographic approach that enables information sharing and analysis while keeping sensitive inputs secret faster and easier to use for application software developers. M. Bellare et al. [12] proposed Optimal Asymmetric Encryption with RSA. This work aimed to use the cryptography concepts in cloud computing communications and to increase the security.

D. Zissis et al. [18] addressed the details of cloud computing security issues and they proposed Public Key Infrastructure operating based on SSO and LDAP, to ensure the authentication, integrity and confidentiality of involved data and communications. The solution, presents a horizontal level of service, available to all implicated entities, that realizes a security mesh, within which essential trust is maintained. C. Hongbing et al. [20] presented an alternative approach called secure Big Data Storage and Sharing Scheme for Cloud Environment. Tenants which divides big data into sequenced parts and stores them among multiple Cloud storage service providers. Instead of protecting the big data itself, the proposed scheme protects the mapping of the various data elements to each provider using a trapdoor function. The cloud computing reviewed was presented in the paper [15]. In this paper [15] authors have discussed the relationship between big data and cloud computing, big data storage systems, and Hadoop technology [17], [19]. Furthermore, research challenges are investigated, with focus on scalability, availability, data integrity, data transformation, data quality, data heterogeneity, privacy, legal and regulatory issues, and governance. Lastly, open research issues that require substantial research efforts are summarized.

J. Zhou et al. [16] proposed a scheme called Secure and Privacy Preserving Protocol for Cloud-based Vehicular DTNs which solved the open problem of resisting layer-adding attack by outsourcing the privacy-preserving aggregated transmission evidence generation for multiple resource constrained vehicles to the cloud side from performing any one-way trapdoor function only once. The vehicle privacy is well protected from both the cloud and transportation manager.

III. Problem Formulation for Secure Cloud Computing

Consider three parties (shown in Fig. 1): a user Alice that stores her data in the cloud; a user Bob with whom Alice wants to share data; and a cloud service provider that stores Alice's data. To use the service, Alice and Bob begin by downloading a client application that consists of a data processor, a data verifier and a token generator. Upon its first execution, Alice's application generates a cryptographic key. We will refer to this key as a master key and assume it is stored locally on Alice's system and that it is kept secret from the cloud service provider. Cloud security is measured in terms of Availability, Integrity, and Confidentiality and encryption techniques are prone to a number of attacks like:

A. Availability

In this scenario cloud service providers have multiple servers. When one server fails, there is no security issue as another server is ready to provide services.

B. Integrity

The data integrity means the correctness and trustworthiness of the data. It ensures that the computation on sensitive data is correct. The data cannot be altered by the unauthorised user.

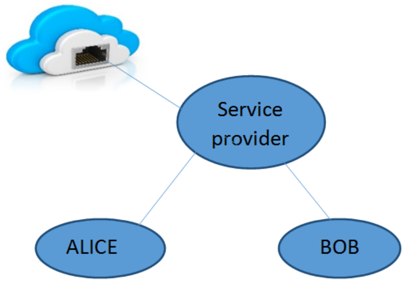

Fig. 1. *Cloud Architecture under various Cryptographic Techniques*

C. *Confidentiality*

Confidentiality is to prevent sensitive information from the reach of the attacker, while making sure that the authorised user have access to it. Services require user to trust the cloud with their data. But in the untrusted cloud Data owners do not trust the cloud. Thus user side protection is necessary. Users encrypt their data before storing into the cloud with the help of a public key .

D. *Cycle attack*

In this attack, the cipher text is encrypted repeatedly and the no of iterations are counted until the original text appears. It can decrypt any cipher text.

E. *Cipher text attack*

In this attack, both the plaintext and the cipher text is known to the attacker and he can use this to discover private exponent and once it discovered it is easy to find then. Multiple parties wish to perform operations on their inputs. This requires decryption of their data. This poses security problems in the case of untrusted Clouds.

Can multiple parties store their data with efficient cryptographic techniques that are resistant to attacks and perform the computations without decrypting their data?

IV. Objective

The Cloud environment requires protection and confidentiality of user data while leveraging computation ability of entities in the cloud network directly on encrypted data. This paper focuses on an issue that is attractive to many types of research, which is a data encryption for cloud computing. Cloud environment requires security and confidentiality of user data while leveraging the computational ability of entities in the cloud network directly on encrypted data. In this paper, we have proposed a scheme that integrates the multi-party computation with homomorphic encryption to allow calculations of encrypted data without decryption.

V. Our Contribution

In our paper, we have proposed an efficient cryptographic technique by padding the multiple party data before encrypting it. The user's data is encrypted using padding scheme Optimal Asymmetric Encryption Padding (OAEP) together with Hybrid Encryption algorithm that is based on RSA Small-e and Efficient RSA (HE-RSA). In order to allow multiple parties to compute a function on their inputs while preserving Integrity and Confidentiality. The homomorphic encryption is performed on the encrypted data without decrypting it in computationally powerful clouds. The proposed scheme integrates the multi-party computation with homomorphic encryption to allow calculations of encrypted data without decryption. The output of this nature allows maintaining confidentiality and integrity in the cloud environment.

VI. Cryptographic Technique for Secure Cloud Computing

A. *Preliminaries And Notations*

$$G : \{0,1\}^{K0} \longrightarrow \{0,1\}^{K0} \tag{1}$$

G is a mask generation function based off a hash function with SHA1 as defined by RFC 3447. G expands the K0 bits of r to K-K0 bits.

$$H : \{0,1\}^{K-K0} \longrightarrow \{0,1\}^{K-K0} \tag{2}$$

H is SHA-256 hash function. H reduces the K-K0 bit to K0 bits. r is a random seed of size K0.

B. *The OAEP Cryptosystem*

It is a padding scheme, proposed by Bellare and Rogaway ([12]), which prevents partial decryption of ciphertexts by adding an element of randomness.

Encode Operation

Select random integer r such that $1 < r < n$.

$$r \longleftarrow \{0,1\}^{K0} \tag{3}$$

$$S = (M \parallel 0^{K1}) \oplus G(r) \tag{4}$$

$$t = r \oplus H(s) \tag{5}$$

Return $s \parallel t$

Decode Operation

$$r = t \oplus H(s) \tag{6}$$

$$S = (M \parallel 0^{K1}) \oplus G(r), \tag{7}$$

r is a random seed of size K1.

C. HE-RSA

Hybrid Encryption algorithm that is based on RSA Small-e and Efficient RSA (HE-RSA) [11]. Efficient RSA was introduced as a scheme that employs the general linear group of order h with values that was intentionally selected randomly from the ring of integer mod n. n is used as the modulus for both the public and private keys. Its length, usually expressed in bits, is the key length.

$$p, q \in prime, n = p.q \tag{8}$$

$$\emptyset(n) = (p-1)(q-1) \tag{9}$$

$$\Upsilon(n,h) = (p^h - p^0)(p^h - p^1)..(p^h - p^{h-1})(q^h - q^0)..(q^h - q^{h-1}) \tag{10}$$

Select random integer r such that $1 < r < n$ and $gcd(r, \Phi) = 1$ and $gcd(r, \Upsilon) = 1$

Compute e such that $r.e = 1 mod \Phi, 1 < e < \Phi(n)$

Compute d such that $d.e = 1 mod \Upsilon, 1 < d < \Upsilon(n)$

Public key: (e, n)

Secret key: (r, d, n)

D. Homomorphic Encryption

Operations can be performed on encrypted data using Homomorphic Encryption [2], [11], [20]. The result of these operations when decrypted using a secret key is same as if we had performed operations on the original data. Multiplicative Homomorphic encryption which allows only products on the original data.

$E(x.y) = E(x).E(y)$

VII. Proposed Algorithm for Secure Cloud Computing

The proposed secure cloud computing algorithm is ensuring the security and privacy [21], [22] of individual data in the cloud along with the enhancement of the security mechanism like Homomorphic Encryption and Multi Party Computation (MPC).

The Proposed Algorithm is based along the four phases: Key Generation, Encryption, Homomorphic Encryption (HE) and Multi Party Computation (MPC), and Deception. The main goal is to minimize the running time, cost, and the overhead during these four phases. In the proposed Algorithm, the number of exponents during key generation (in the step 1) has been enlarged in comparison to the some existing Algorithms (i.e., RSA). In addition to that, a dual encryption process (in the step 2) has been implemented in this Algorithm to prevent the general attacks against some existing techniques. In step 3, we have integrated the fully homomorphic encryption and multi-party computation to allow the calculations of encrypted data without decryption (in the step 4) in the cloud.

To interact with various services in the cloud and to store the data generated/processed by those services, several security capabilities are required. Suppose Alice and Bob send data to M1 and M2 respectively to the cloud and Alice's data (M1) is padded using Optimal Asymmetric Encryption Padding (OAEP) scheme before being encrypted using HE-RSA resulting in ciphertext C1 which is shown in Fig. 2.

Alices data (M1) is padded using Optimal Asymmetric Encryption Padding (OAEP) padding scheme before being encrypted using HE-RSA resulting in ciphertext C1 (shown

Algorithm 1 Algorithm for Secure Cloud Computing

```
  Step 1: Key generation algorithm: keygen(p,q)
  Randomly choose two large primes p ,q and compute n=p.q
.
$\emptyset(n) = (p-1)(q-1)$
$\Upsilon(n,h) = (p^h - p^0)(p^h - p^1) \ldots\ldots\ldots\ldots\ldots (p^h - p^{h-1})(q^h - q^0)(q^h - q^1) \ldots\ldots\ldots (q^h - q^{h-1})$
  Select random integer r such that $1 < r < n$ and $gcd(R, \Phi) = 1$ and $gcd(R, \Upsilon) = 1$
  Compute e such that $r.e = 1 \bmod \Phi, 1 < e < \Phi(n)$
  Compute d such that $d.e = 1 \bmod \Upsilon, 1 < d < \Upsilon(n)$
  Public key(pk): (e,n)
  Secret key(sk): (r,d,n)
  Step 2: : Encryption: Enc(M,pk)
  Suppose Sender and Receiver send data to M1 and M2 respectively to the cloud
$G : \{0,1\}^{K0} \longrightarrow \{0,1\}^{K0}$
$H : \{0,1\}^{K-K0} \longrightarrow \{0,1\}^{K-K0}$
$r \longleftarrow \{0,1\}^{K0}$
$S = (M \parallel 0^{K1}) \oplus G(r)$
$t = r \oplus H(s)$
$C \longleftarrow \ll S \gg^e \bmod n \gg^e \bmod n$
  Return C
  Step 3: Homomorphism and Multi-party computation
  Homomorphic computations are performed on Sender and Receiver encrypted data C1 and C2 respectively.
$C1 = ((M1)^e \bmod \mathrm{n})^e \bmod n$
$C2 = ((M2)^e \mathrm{mod}\ n)^e \bmod n$
$C1.C2 = [((M1)^e \bmod n)^e \bmod n][((M2)^e mod\ \mathrm{n})^e \bmod n]$
$= ((M1)^e \bmod n)^e((M2)^e \bmod n)^e \bmod n$
$= ((M1M2)^e \bmod n)^e \bmod n$
$LetC = C1.C2, M = M1M2$
$C = (M^e \bmod n)^e \bmod n$
  Step 4: Decryption: Dec(C,sk)
  Sender and Receiver decrypt the computed data C using their respective private keys
$W \longleftarrow (C^r \bmod n)^d \bmod n$
  Parse W as $s \| t$
$r \longleftarrow H(s) \oplus t$
$M^1 \longleftarrow s \oplus G(r), parseM^1asM \| Z$
```

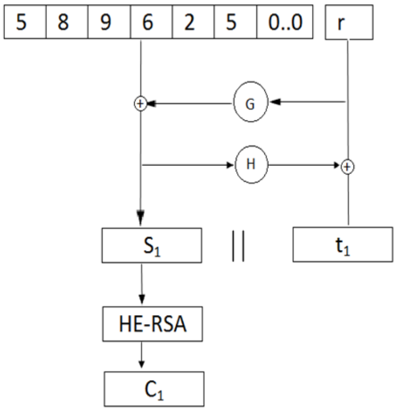

Fig. 2. *Sender's data M1(Encryption Process)*

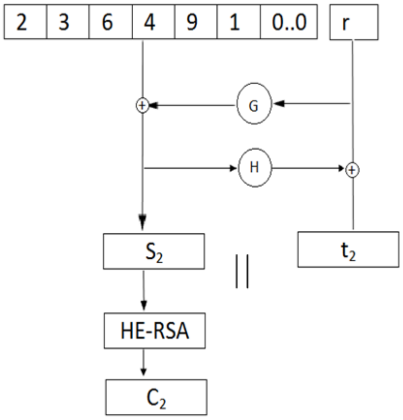

Fig. 3. *Receiver's data M2(encryption process)*

in Fig. 2). Receiver's data (M2) is padded using Optimal Asymmetric Encryption Padding(OAEP) scheme before being encrypted using HE-RSA resulting in ciphertext C2 (shown in Fig. 3). The data C resulting from homomorphic encryption is decrypted using Sender's private key and is then decoded using OAEP(shown in Fig. 4).

The data C resulting from homomorphic encryption is decrypted using Receiver's private key and is then decoded using OAEP(shown in Fig. 5). For instance, homomorphic encryption requires that all users and the eventual recipients of the results share a key to encrypt the inputs and decrypt the results, which may be difficult to arrange if they belong to different organizations. Also, homomorphic encryption does not allow for computation on data encrypted using different keys (without incurring additional significant overhead), thus making it impossible for users allow different access to data they contribute to the computation.

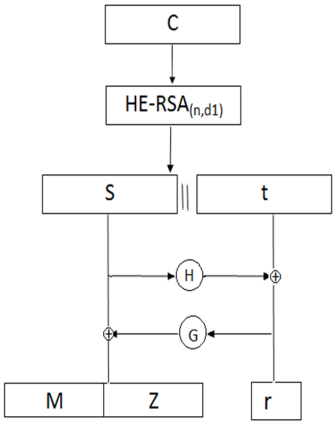

Fig. 4. *Sender's data(decryption process)*

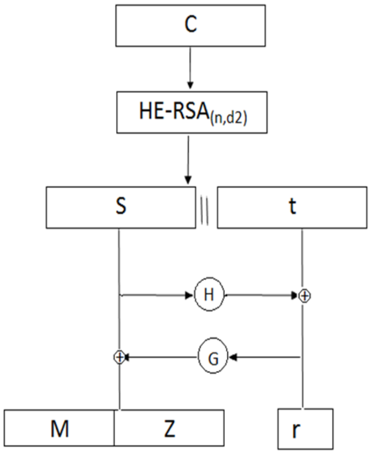

Fig. 5. *Receiver's data(decryption process)*

Secure multi-party computation (MPC) is suited to take advantage of the semi-trusted cloud setting. MPC leverage is the presence of honest parties, without necessarily knowing which parties are honest, to achieve confidentiality and integrity of the data and computation. Multi-party computation offers weaker security guarantees than FHE, but can be much more efficient. In MPC, no single party learns anything about the data, but if sufficiently many parties are corrupted by an adversary and pool their information, they can break confidentiality. The relative efficiency of MPC, as well as the applicability of the semi-trusted cloud model to the real world, make it a promising candidate for use in more practical secure cloud computation.

VIII. Results

After combining Homomorphic encryption and Multi Party Computation(HE +MPC), the confidentiality and integrity of the data is maintained and the overhead is less than Homomorphic Encryption but more than Multi Party Computation. So, we have received moderate overhead based on the Homomorphic Encryption and Multi Party Computation (shown in Table 1). Fig. 6 summarizes the approximate

TABLE I
COMPARISON OF CRYPTOGRAPHIC APPROACH

Cryptographic Technique	Confidentiality	Integrity	Interaction	Overheads
HE	YES	NO	NO	More Overheads
MPC	YES	YES	YES	Less Overheads
HE+MPC	YES	YES	YES	Moderate Overheads

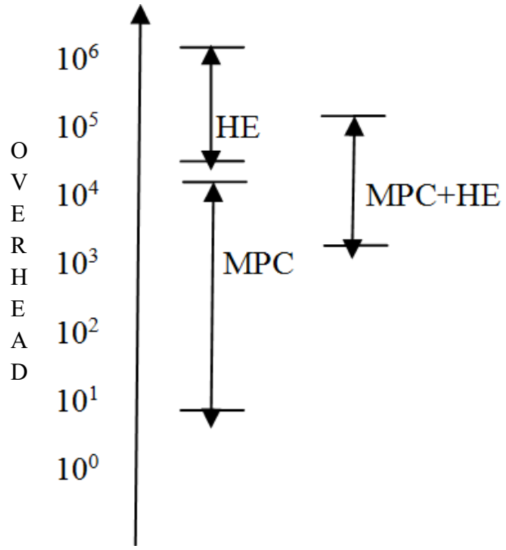

Fig. 6. *A graphical depiction of the multiplicative performance overheads over unsecured computation incurred by Homomorphic Encryption (HE), Multi Party Computation (MPC) and Homomorphic Encryption + Multi Party Computation (HE + MPC).*

efficiency, and cost of each of the techniques across a wide range of computations, depicting the multiplicative performance overhead incurred over unsecured computation. In addition to its inefficiency, homomorphic encryption has other limitations.

IX. CONCLUSION

In this paper, we proposed a secure cloud computing model in which efficient cryptographic technique Based on Homomorphic Encryption (HE) And Multi-party Computation(MPC) was used to encrypt user's data followed by operations on their data while maintaining integrity and confidentiality. The output is same as if the operations have been carried on raw data. A party is able to jointly perform computations without revealing their data to the other party. Here, we designed and developed secure homomorphic encryption and multi-party computation techniques tailored specifically for a private semi-trusted cloud setting. This setting allows developers to design the private cloud together with the cryptographic techniques (i.e., HE+MPC) necessary to protect it.

X. ACKNOWLEDGEMENT

This work is partially supported by Early Career Research Award from Science and Engineering Research Board (SERB), Department of Science and Technology (DST), Govt. of India, New Delhi, India (Project Number : ECR/2015/000256) and BITS-Pilani K.K Birla Goa Campus Research Initiation Grant (RIG) Award 2017.

REFERENCES

[1] Mell P, Grace T. *The NIST definition of cloud computing*, NIST Special Publication, 2009, pp. 800–145.

[2] Tebaa M, Hajji S.E, Ghazi A.E. *Homomorphic Encryption Applied to the Cloud Computing Security*, Proceedings of the World Congress on Engineering, London, U.K., Vol.1, No.1, 2014, pp. 4-6.

[3] Wang Z, Sun G, Chen D. *A new definition of homomorphic signature for identity management in mobile cloud computing*, Journal of Computer and System Sciences, Vol. 80, N0. 3, 2014, pp. 546-553.

[4] Yakoubov S, Gadepally V, Schear N, Shen E, Yerukhimovich A. *A Survey of Cryptographic Approaches to Securing Big-Data Analytics in the Cloud*, IEEE High Performance Extreme Computing Conference (HPEC), 2014, pp. 1–6.

[5] Rong C , Nguyen ST, Jaatun MG. *Beyond lightning: A survey on security challenges in cloud computing*, Computers and Electrical Engineering, Vol. 39, No. 1, 2013, pp. 47-54.

[6] Gentry C.*Computing Arbitrary Functions of Encrypted Data*, Communications of the ACM, Vol. 53, No. 3, 2010, pp. 97-105.

[7] Wang C, Wang Q, Ren K, Lou W. *Ensuring Data Storage Security in Cloud Computing*, Quality of Service, 2009, pp. 1–9.

[8] Yu Y, Niua L, Yang, G, Mu Y, Susilo W. *On the security of auditing mechanisms for secure cloud storage*, Future Generation Computer Systems, Vol. 30, 2014 pp. 127-132.

[9] Wei L, Zhu H, Cao Z, Dong X, Jia W, Chen Y, Vasilakos AV. *Security and privacy for storage and computation in cloud computing*, Information Sciences, Vol. 258, 2014, pp. 371-386.

[10] Lopez-Alt A, Tromer V, Vaikuntanathan E. *On-the-Fly Multiparty Computation on the Cloud via Multikey Fully Homomorphic Encryption*, Proceedings of the forty-fourth annual ACM symposium on Theory of computing, 2012, pp. 1219–1234.

[11] Brakerski Z,and Vaikuntanathan E. *Efficient fully homomorphic encryption from (standard) LWE*, SIAM Journal on Computing, Vol.43, No.2, 2011, pp. 831–871.

[12] Bellare M, and Rogawayy P. *Optimal Asymmetric Encryption How to Encrypt with RSA*, Advances in Cryptology Eurocrypt 94 Proceedings, Vol. 950, 1995, pp. 1–19.

[13] Shen E, Varia M, Cunningham RK, Vesey WK. *Cryptographically Secure Computation*, IEEE Computer Society, Vol. 48, No.4, 2015, pp. 78–81.

[14] Zissis D, Lekkas D. *Addressing cloud computing security issues*, Future Generation Computer Systems, Vol. 28, No. 3, 2012, pp. 583–592.

[15] Hashem IAT, Yaqoob I, Anuar NB, Mokhtar N, Gani A, Khan S.U. *The rise of 'big data' on cloud computing: Review and open research issues*, Information Systems, 2015, Vol. 47, pp. 98-115.

[16] Zhou J, Dong X, Cao Z, Vasilakos AV. *Secure and Privacy Preserving Protocol for Cloud-based Vehicular DTNs*, IEEE Transactions on Information Forensics and Security,Information Systems,Vo. 10, No. 6, 2015, pp. 1299 - 1314.

[17] Zhao J, Wang L, Tao J, Chen J, Sun W, Ranjan R, Kolodziej J, Streit A, Georgakopoulos D. *A security framework in G-Hadoop for big data computing across distributed Cloud data centres*, *Journal of Computer and System Sciences*, Vol. 80, No. 5, 2014, pp. 994-1007.

[18] Zuech R, Khoshgoftaar TM, Wald R. *Intrusion detection and Big Heterogeneous Data: a Survey*, Journal of Big Data, Springer, Vol. 2, No. 3, 2015, pp. 1-40.

[19] Hongbing C, Chunming R, Kai H, Weihong W, Yanyan L. *Secure Big Data Storage and Sharing Scheme for Cloud Tenants*, China Communications, 2015, pp. 106–115.

[20] Jajodia S, Kant K, Samarati P, Singhal A, Swarup V, Wang C. *Secure Cloud Computing*, Springer Science+Business Media, 2014, pp. 1-350.

[21] D. Das, R. Misra, A. Raj *Approximating Geographic Routing using Coverage Tree Heuristics for Wireless Network*, Wireless Networks (WINE), Springer US, Vol. 21. No. 4, 2015, pp. 1109-1118.

[22] T. Limbasiya, D. Das, *Secure Message Transmission Algorithm for Vehicle to Vehicle (V2V) Communication*, IEEE Region 10 Conference (TENCON)2016, Technologies for Smart Nation, Singapore, 22-25 Nov. 2016, Singapore, pp. 2507-2512.