Proxy re-encryption in cloud using ALBC (adaptive lattice based cryptography)

Chandrakala B M[1], S C Lingareddy[2]
[1]ISE Department, Dayanandasagar College of Engineering, Bangalore, India
[2]Department of CSE, Sri Venkateshwara College of Engineering, Bangalore, India

Article Info

Article history:

Received Jan 31, 2019
Revised Jul 2, 2019
Accepted Jul 28, 2019

Keywords:

Encryption
Proxy re-encryption
Re-encryption

ABSTRACT

In recent days, data sharing has provided the flexibility to share the data, store the data, and perform operation on data virtually as well as cost effectively. Data sharing in cloud is one of the feature, which is being popular and widely accepted. However, the concern here is to ensure the data security and this has led the researcher to research in this area. To provide the security several Proxy re-encryption scheme has been introduced, however all these method lacks of efficiency. Hence In this paper, we propose a scheme known as ALBC (Adaptive Lattice Based Cryptography), this scheme follows the two phase i.e. encryption and Re-encryption. Encryption phase has few algorithms such as Key_Gen, Enc, Dec. Similarly ALBC Re-Enc has five algorithm i.e. Key_Gen, Key_ReGen, Enc, Re-Enc, Dec. our algorithm not only provides the security but also solves the problem of RL (Ring-learning) with errors problems. In order to evaluate, our algorithm is compared with the existing model in terms of encryption time, decryption time, re-encryption time, key generation and key regeneration by varying the various key size. When we observe the comparative analysis, it is observed that our algorithm outperforms the existing algorithm.

Corresponding Author:

Chandrakala B M,
ISE Department, Dayanandasagar College of Engineering,
Bangalore, India.
Email: chandrakalabm2016@gmail.com

1. INTRODUCTION

Nowadays cloud computing is accepted widely and it is growing day by day, this has helped and provide flexibility for the user to store the data and share the data [1]. For an example, a firm enables its employees of the same group to share the files in given public cloud, with the help of cloud computing the employees of the same group can easily access the data that are uploaded by the data owner of that particular group with low investment . Moreover, data stored are shared in the cloud and it can be accessed by given member of the group, this process can take place at any time from any place with the availability of internet. Moreover, the cloud has several benefits; however, it raises the security challenges and concern [2]. The main concern here is that only the accessed user should be authorized to access the data. Hence to ensure this encryption has been introduced, encryption is a particular kind of technology which provides the control over the encrypted data.

Encryption helps in protecting the sensitive data, which has been outsourced in given cloud server [3], this is protected until the data is encrypted. This is said to be one of the essential approach for protecting the data that are stored in the cloud. The data in the cloud is encrypted with general asymmetric encryption. In order to share the data storage with the other group members in the group, it is required that data owner should download the data and decrypt the requested data and re-encrypt the data by using the public key [4]. Moreover, this alone cannot guarantee the security goal since the communication overhead as well as computation cost, this in terms contradicts the cloud computing motivation. With the help of cloud computing, data sharing has become easier than ever, however, one of the main concern is the security how to ensure the data security while sharing the data. This concern has raised the issue for not adopting the cloud services in many field. Hence it is essential to take the step forward for protecting the data privacy of the data that are stored in the cloud.

The Figure 1 shows the general proxy Re-Encryption in cloud computing environment [5]. The above diagram has four module namely data owner, cloud storage, data users, proxy server. In every data field, the data should have data owner who has the authority for deciding the access of the data, usage of the data. Data owner is the one who is original generator of the data. Data owner can update and upload the file. Cloud storage is one of the data storage model, where the data are stored in the various logical pools. Cloud storage are solely responsible for storing the data and keeping it accessible and available at any time at any place over the internet [6]. Data users are the users, they can access the data use the data. However, they cannot modify the data. Moreover, only the authorized user can access the data. In this system, users can access the files from cloud, which are uploaded by data owner. However, for accessing these files the utilizer should be gratifies the access control policy which is provided by the data owner. When utilizer is slake the policy then utilizing of Re-Encryption key, which is engendered by data owner, utilizer can be decrypt the Re- Encrypted data.

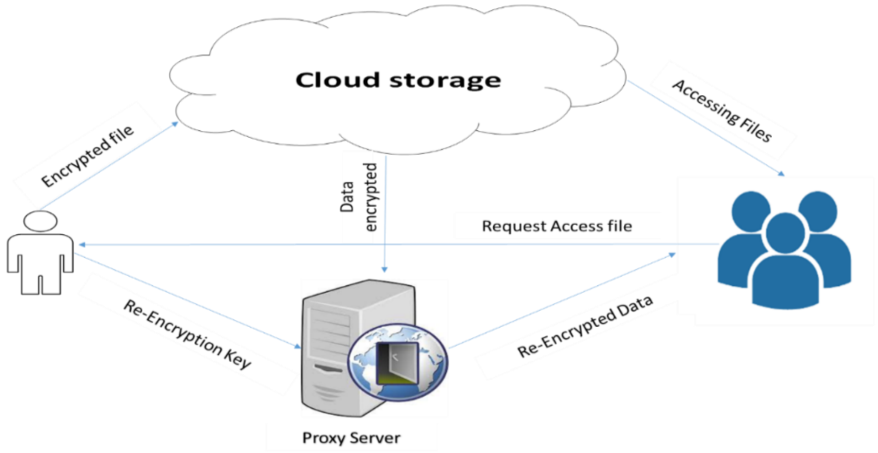

Figure 1. Proxy Re-Encryption in cloud computing

Proxy Re-Encryption is one of the technique which helps in ensuring the data security in the given cloud platform. Moreover, several scheme has been proposed in the past, which ensures the security of data such as identity based state-of art technique. However, all these technique fails miserably in one or the other parameter such as some of the technique produces more communication overhead, other methods requires more time for generation of key, encryption and decryption time is more. They are discussed in the next section of our research. Apart from all these issue there is one more issue known as RL (Ring learning) with errors problem also known as RLWE. This problem arises when the encryption is done over and over, hence in order to solve this issue we introduce a technique known as ALBC (Adaptive Lattice Based Cryptography).This not only helps in solving the problem of RLWE, but also gives the satisfactory results in terms of security.

Our work is organized such that the section 2 presents brief discussion over the existing method along with their shortcomings. Section 3 depicts the proposed methodology including the pictorial representation of methodology. In section 4 we evaluate our algorithm by comparing ALBC with existing technique. In the last section, we conclude our research work.

2. LITERATURE SURVEY

At first the single use and pairing free bi-directional PRE scheme has been proposed for securing the data against the attacks of ciphertext in the given random oracle model [7], this is achieved by minimizing its security to the CDH (Computational Diffie Hellman) problem. Moreover, [8] proposed the first unidirectional Scheme (PRE), this was packed with the RCC (Repay able Chosen Ciphertext) security, this was implemented in adaptive corruption model, this in terms helped in solving the problem that arises in the previous paper [9]. In [10] tries to solve the problem of the two paper, first unidirectional PRE scheme is constructed along with the collusion-resistance and CCA security. Moreover, the bilinear pairing was removed from this. In [11], attribute based PRE is presented by combining attribute-based encryption and PRE, in this method the specified attributes is allowed to re-encrypt the ciphertext with certain policy to the other encryption under the various access policy a methodology where the delegator itself owns the fine-grained control. Moreover, the transformation of ciphertext is done only by the proxy and decrypted by delegate; this is done only if the ciphertext satisfies the particular condition. Meanwhile based on the multi-linear maps, multi-use and PRE scheme is used.

In [12], unidirectional as well as multi-use PRE scheme has been proposed, these scheme are based on the multi-linear maps. This helps in knowing how PRE scheme designing is done simultaneously based on multi-hop and unidirectional. In [13] existing security of the existing model is determined and a novel nomenclature is proposed which can access the both decryption as well as re-encryption oracles, this model is constructed under the theoretical assumption. [14, 15] presented re-encryption based on the cross-cryptosystem, this is done by allowing the given authorized proxy for converting the complicated broadcast encryption based on the ID. And the ciphertext is deployed in the provided server to the general ID-based encryption. [16] introduced a scheme based on the IB (Identity Based)-PRE, this scheme was based on the idea of ID based and Pre. This provided the complete deployment of ID-based Pre scheme which is based on the bilinear paring this scheme is non-interactive, multi-use and unidirectional when compared with the basic PRE scheme, ID based method gives the advantage of ignoring the tedious certificate [17].

In [18], introduced one more scheme which is known as the key private PRE, this gives the flexibility to keep the key as a private such that even a given proxy cannot find the difference among the involved users. This was one of the interesting method for securing the method, however the work is still under the progress. Moreover, once proxy communicates with several users the data should not be revealed from one user to the other. In this prepare method known as ABPRE which was based on the ciphertext policy is a combination of traditional PRE scheme and ABE (Attribute Based Encryption) ,this methodology was secured against the CPA. In this scheme the given key is associated with the structure given, this helps in solving the problem of key distribution and multiple users over the huge data. Many times overhead situation takes place and it is caused by the key management. Several algorithm has been proven against the ciphertext under the BDH assumption [19]. In this fine grained-AC (Access Control) is given to the user for specifying that who can decipher the given message or the data, this takes by setting with the given set of attribute [20]. Later to avoid the collusion resistant scheme based on the unidirectional along with the monatomic access structure is introduced [21]. However it lacks with several issue such as key management, overhead and others.

Through the literature survey, we observe that various method has been presented in the past for re-encryption. However all these scheme lacks in one or other criteria such as communication overhead, collusion resistant and other. Hence in the next section we propose a methodology which overcomes the above discussed problem of existing scheme.

3. PROPOSED METHODOLOGY

In this section a scheme known as ALBC (Adaptive Lattice Based Cryptography) is presented, our methodology has two phase namely encryption phase and decryption phase. Encryption phase has mainly three algorithm first algorithm where the key is generated, in second algorithm encryption takes place, last but not the least in algorithm of encryption phase decryption is done.

Our scheme ALBC has several properties which helps in providing more flexibility to secure the data and they are:

a. Multiple Encryption
Our algorithm supports the multiple encryption; here the error term is included such that there is a growth in noise on each loop.

b. Bidirectional
Our scheme supports the bidirectional properties, i.e. given the $nu_{z\to k} = s_z s_k^{-1},\ nu_{k\to z} = CIP_k CIP_Z^{-1}$.

c. Collusion Safe
In order to avoid the collision secret keys are extracted from the Re-Encryption if the proxy collides with each other. Our algorithm is elaborated over the two rings i.e. A and A_f, parameters used is c and f. The data (plain text) along with the parameter is correspond to the ring i.e. A/e, here $e \in A_f^B$ has been considered as parameter. Moreover, the characterization of the family distribution is done by using the α parameter. Henceforth the global parameter used are $(c, f, e, \alpha, \sigma)$. This algorithm is similar to the MLBC (Modified Lattice Based Cryptography) along with that our methodology also generates the keys and the noise terms is included, our algorithm also provides the security over the Ring-LWE problem.

3.1. ALBC-Enc Phase (Encryption Phases)

The Figure 2 shows the Encryption phase of our algorithm.

a. Generation of Key

This algorithm gives the output of secret key and public key $(\text{seck}_Y, pubk_Y). \epsilon A_q^B \text{X} A_q^B$. Gaussian Distribution is considered as $Dist_{p^c}$. Moreover, the keys are computed as given below:

Step1: using the $Dist_{p^c}, \sigma$, s' is sampled, let $s_A = 1 + e . s'$

Step2: In case if $(s_A \; mod \; q) \notin A_q^B$

Step3: Sampling given l_Z from $Dist_{p^c}$

Step4: In case if $(l_Z \; mod \; q) \notin A_f^B$ resampling is done.

Step5: Computation of $M_Z = e. l_Z \; .s_Z^{-1}$

Step6: secret key is returned i.e. $sk_A = f_A$ and $ei_Z = M_Z$

b. Encryption

Here the input given is message$M\epsilon$ɰ, noise polynomials s, e are sampled from the given distribution $\psi_\alpha$output expected is in the form of ciphertext $CIP_Z = M_Z O + er + Msg\epsilon A_f$

c. Decryption

Here, the ciphertext CIP_Z and the secret key seck_Y= s_Z , this algorithm helps in computing the $CIP_Z{}'$ = $CIP_Z . s_Z$ and output generated in terms of message i.e. $Msg = (CIP_Z{}' \; mod \; e) \in n$.

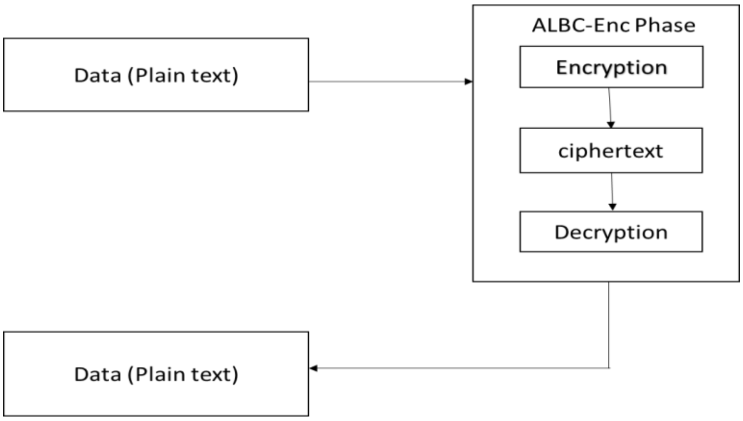

Figure 2. ALBC-Encryption phase

3.2. ALBC (Adaptive Lattice Based Cryptography) Re-encryption

This phase uses the tuple $(c, f, e, \alpha, \sigma)$ and global parameter, and in this phase following algorithm are included. The Figure 3 shows the re-encryption process of our proposed methodology, here the global parameters are used and this phase consist of five algorithm. First algorithm i.e. in Key generation public as well as secret keys are generated, In second algorithm with the help of secret keys the Re-Encryption between the two given user are calculated. Third algorithm the input given is message as well as given public key, this in terms helps in achieving the ciphertext. Similarly, in the Re-encryption algorithm the input given is Re-Encryption key and cipher text , outcomes of this algorithm is expected to be the ciphertext. In the Decryption algorithm the ciphertext along with the secret key is given input and this in terms gives the original message.

a. Generation of key

Here, the output generated is secret key and public key i.e. $(\text{seck}_Y, pubk_Y). \epsilon A_q^B \text{X} A_q^B$, the key is generated by using the following steps involved.

Step1: using the $Dist_{p^c}$, σ, s' is sampled, let $s_z = 1 + p . s'$

Step2: In case if $(s_A \; mod \; q) \notin A_q^B$

Step3: Sampling given l_Z from $Dist_{p^c}$

Step4: In case if $(l_Z \; mod \; q) \notin A_q^B$ resampling is done.

Step5: Computation of $M_Z = e. l_Z \; .s_Z^{-1}$

Step6: secret key is returned i.e. $\text{seck}_Y = s_z$ and $pk_A = pubk_Y$

b. Regeneration of key
This algorithm takes the input of two secret keys i.e. $\mathrm{seck}_Y = s_z$ and $\mathrm{seck}_k = O_K$. This algorithm computes the re-encryption key between the two given users A and B as $nu_{Z\to K} = \mathrm{seck}_Y.seck_k^{-1}$ $=s_z.s_k^{-1}$.

c. Encryption
In this algorithm, input taken is message $Msg \in n$ and the public key $pubk_Y$. In this algorithm noise polynomials s is sampled and e from the given distribution φ_α. The output expected is in the form of ciphertext $CIP_z = m_z O + er + Msg + A_f$.

d. Re-Encryption
In this algorithm, input taken is ciphertext C_A and re-encryption key$rk_{A\to B}$. In this noise polynomials p' from the given distribution φ_α and the output expected is $CIp_k = CIP_Z.nu_{z\to k} + er' + A_f$.

e. Decryption
In this algorithm, the input taken is ciphertext CIP_Z and given secret key $\mathrm{seck}_Y = s_z$. This algorithm helps in computing the $C_A^{"} = C_A.f_A$ and the output is expected in the form of message i.e. $Msg = \left(CIp_z^{"}\ mod\ e\right) \in Msg$. From the algorithm it is clear that the process of re-encryption results in increase in error terms and this can be cause for the failure of decryption algorithm. However, our proposed algorithm is capable enough to handle the error terms.

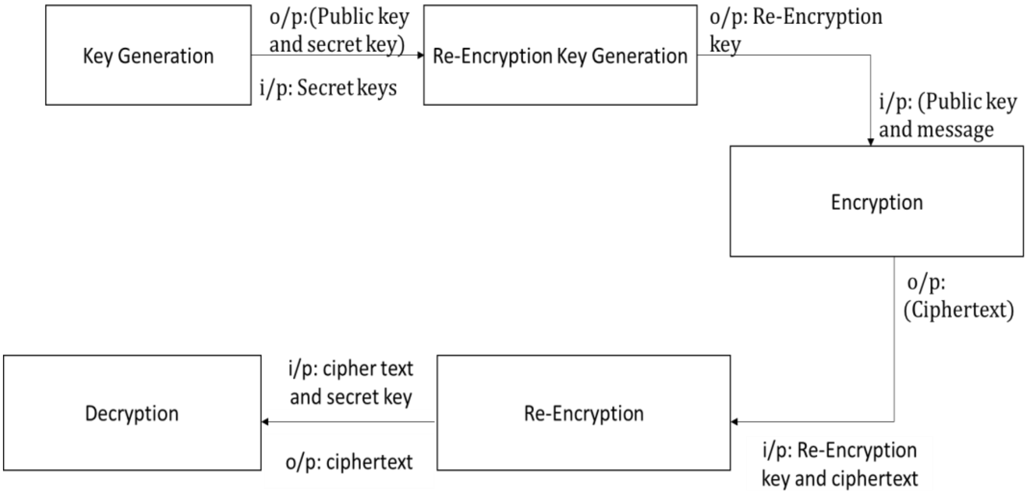
Key Generation
o/p:(Public key and secret key)
i/p: Secret keys
Re-Encryption Key Generation
o/p: Re-Encryption key
i/p: (Public key and message
Encryption
o/p: (Ciphertext)
i/p: Re-Encryption key and ciphertext
Re-Encryption
i/p: cipher text and secret key
o/p: ciphertext
Decryption

Figure 3. ALBC RE-Encryption

4. RESULTS AND DISCUSSION

This section of research shows the performance evaluation and comparative analysis of our re-encryption algorithm. We have performed this particular research on windows 10 operating system loaded with i5 processor and 3.2 GHZ quad core. The system also has 8 GB of Ram and dedicated graphic of NVIDIA. Our algorithm is evaluated using the libraries of java cryptography in eclipse. Simulation is conducted by varying the key size such as (256-512, 512-4094, and 512-4094) for the various parameter such as encryption time, decryption time, re-encryption and total computation time. For each of this parameter the computation time is noted and compared with the existing system as given below.

4.1. Comparative analysis

4.1.1. Encryption time

In order to prove the benchmark of our algorithm against the existing system, we have compared the results based on several parameters. The first parameter considered is the encryption time. Encryption time is the time taken to encrypt the given data. In the Figure 4 we have compared the existing system with the proposed one by taking the various key size. N the Figure 4, for the key size (256-512) in case of existing system is 0.015625 whereas proposed system takes 0.013020833 which is less than the existing one. Similarly, in case of key size (512-4094), the existing system requires 0.015625ms, whereas proposed system requires 0.013586957. For the key size (1024-4094), to encrypt the data (plain text) time required is 0.109375ms whereas proposed system encrypts the data in 0.0093482906.

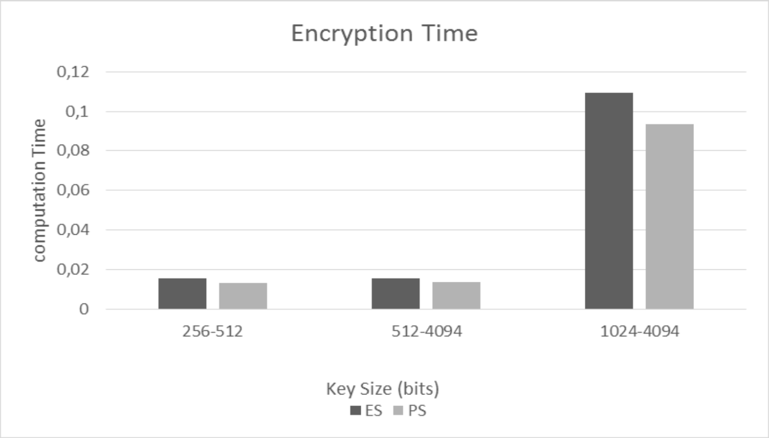

Figure 4. The comparative analysis of existing system with the proposed one by taking the various key size

4.1.2. Re-Encryption Time

The Figure 5 shows the comparative analysis of existing and proposed method based on the Re-Encryption time. We observe that for key size (256-512) time taken to Re-Encrypt the data is 0.03125 where as our proposed methodology takes 0.002469705. In case of key size (512-4094), existing system takes 0.03125ms whereas proposed algorithm takes the 0.00326712 which is comparatively less. At last with the key size (1024-4094), the time taken to re-encrypt the data (plain text) is 0.002831899ms.

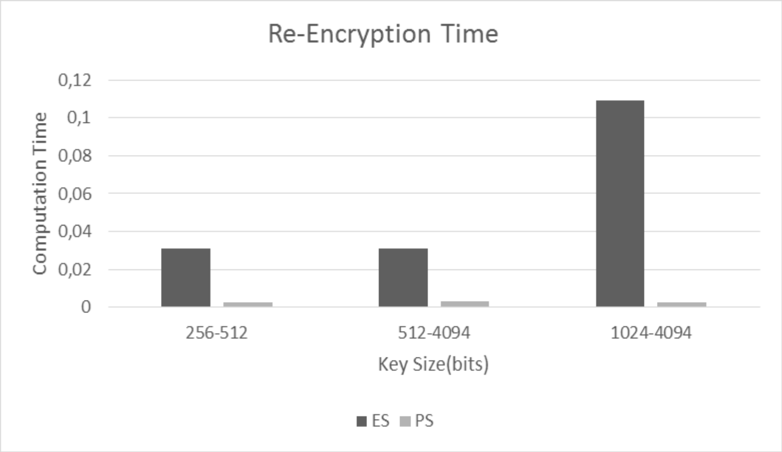

Figure 5. The comparative analysis of existing and proposed method based on the Re-Encryption time

4.1.3. Decryption Time

Decryption time is one of the parameter which has been considered for the comparative analysis, it is time which takes for decrypting the data when key is provides. In the Figure 6, we observe that for the key size (256-512), the decryption time of existing methodology is 0.0625 whereas ALBC (Proposed algorithm) takes only 0.050813008ms, similarly for the key size (512-4094), the time taken to decrypt the data is 0.046875 and ALBC takes 0.036 (approximately) to decrypt the data. For the key size (1024-4094) the decryption time is 0.0625ms and ALBC takes 0.049603175 ms.

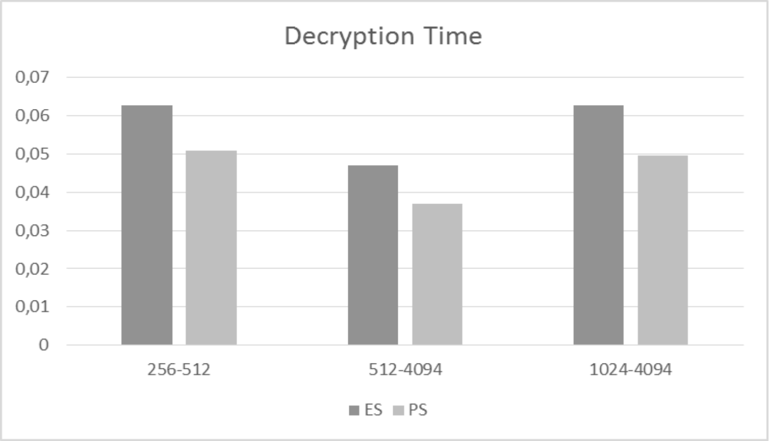

Figure 6. The comparative analysis of existing and proposed method based on the Decryption time

4.1.4. Key-Generation Time

Key Generation time is the time taken for generating the key in order to protect the data, this is considered to be one of the parameter to evaluate the performance of our algorithm. Through the Figure 7, we observe that in case of key size (256-512) the time taken to generate the key is 0.0625 whereas ALBC takes 0.04166, similarly in case of key size (512-4094), the time taken to generate the key for existing system and proposed system is 0.034375 and 0.05208. In case of the key size (1024-4094), the time taken to generate the key by existing algorithm is 0.06875 whereas the time taken by ALBC is 0.04166. From the result we see that in case of the two key size our algorithm performs better than the existing one.

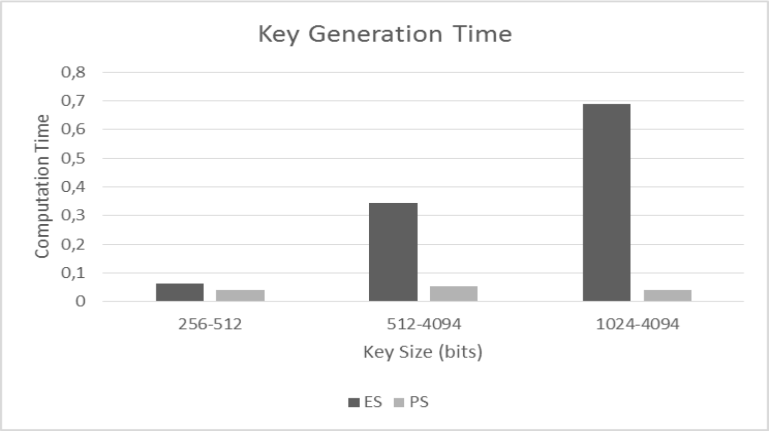

Figure 7. The comparative analysis of existing and proposed method based on the Key Generation time

4.1.5. Proxy Key generation

Proxy Key generation time is the time taken to generate the proxy key, the less time taken to generate the proxy key, the more efficient the algorithm is. In Figure 8, for the key size (256-512) is 0.065625 and the time taken for generation of proxy key by ALBC is 0.04166. In case of key size (512-4094), the time taken to generate the proxy key is 0.0625 and the time taken to generate the proxy key is 0.05208 and for the key size (1024-4094) the time taken to generate the proxy key for existing and proposed methodology is 0.140625 and 0.04166 respectively.

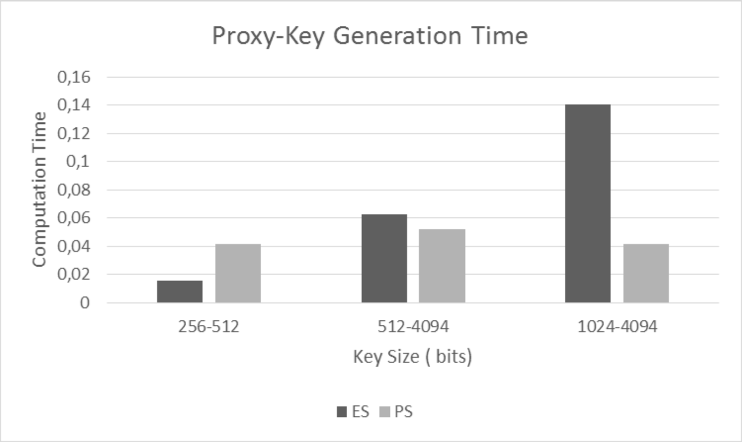

Figure 8. The comparative analysis of existing and proposed method based on the Proxy Key generation time

4.1.6. Total computation time

Total computation time is the time taken to complete the task; in Figure 9, it shows the time taken to complete the re-encryption process. For the key size (256-512), the time taken is 0.53125ms and whereas the total time taken for ALBC is 0.02604, in case of key size (512-4094), the time taken is 1.109375 whereas our proposed algorithm takes only 0.04076 . Similarly in case of the key size (1024-4094), the existing algorithm takes 1.171875 and the proposed algorithm takes only 0.02670ms. In terms of total computation time we see that as the key size increases the computation capability of the existing algorithm reduces, moreover our algorithm either reduces or maintains the similar computation time. When compared with the existing system, we found the marginal difference. Hence, our algorithm excels.

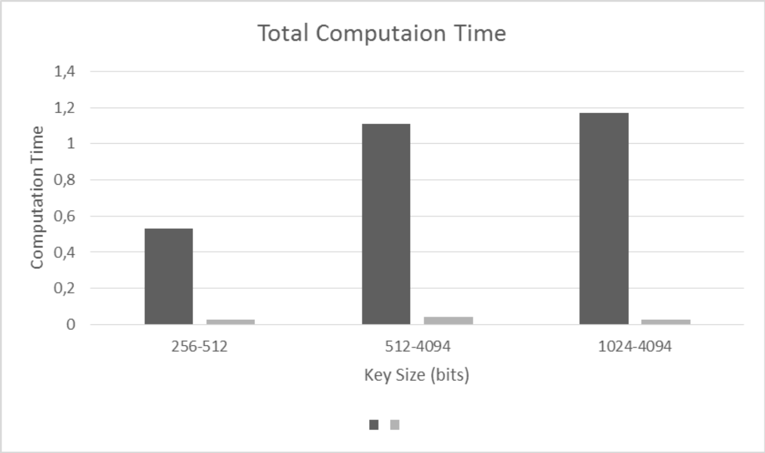

Figure 9. The comparative analysis of existing and proposed method based on the Total computation time

5. CONCLUSION

As the demand of data security in cloud computing increases, in this paper we propose a re-encryption scheme namely ALBC (Adaptive based Lattice Cryptography) which not only helps in securing the data but also solves the RLWE problem. This scheme follows mainly two phases i.e. ALBC_ENC and ALBC_RENC for the encryption and re-encryption, which has several algorithm. For evaluation of algorithm we have considered various size such as (256-512), (512-1094) and (1024-4094), for more proof our method is compared with the existing methodology , Result is compared based on the several parameters such as key generation time, encryption time, re-encryption time, decryption time, key regeneration time and the total computation . Less the time taken to perform on these parameter the better the model is and our scheme i.e. ALBC completely outperforms the existing methodology. However, there are various things that has to be looked in a future such as considering more parameter and reduce the same.

REFERENCES

[1] J. Weinman, "Toward a Theoretical Model of Cloud Computing," *IEEE Cloud Computing*, vol. 5, pp. 92-101, 2018.

[2] S. Basu, et al., "Cloud computing security challenges &amp; solutions-A survey," *2018 IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC)*, Las Vegas, NV, pp. 347-356, 2018.

[3] J. P. Singh, et al., "Authentication and encryption in Cloud Computing," *2015 International Conference on Smart Technologies and Management for Computing, Communication, Controls, Energy and Materials (ICSTM)*, Chennai, pp. 216-219, 2015.

[4] K. Sakurai, et al., "Improved proxy re-encryption scheme for symmetric key cryptography," *2017 International Workshop on Big Data and Information Security (IWBIS)*, Jakarta, pp. 105-111, 2017.

[5] B. Libert and D. Vergnaud, "Unidirectional Chosen-Ciphertext Secure Proxy Re-Encryption," *IEEE Transactions on Information Theory*, vol. 57, pp. 1786-1802, 2011.

[6] D. Zhe, et al., "Study on Data Security Policy Based on Cloud Storage," *IEEE 3rd international conference on big data security on cloud (bigdatasecurity), ieee international conference on high performance and smart computing (hpsc), and ieee international conference on intelligent data and security (ids)*, Beijing, pp. 145-149, 2017.

[7] B. Libert and D. Vergnaud, "Unidirectional chosen ciphertext secure proxy re-encryption," *Public Key Cryptography–PKC'08*, Springer, pp. 360-379, 2008.

[8] R. H. Deng, et al., "Chosen ciphertext secure proxy re-encryption without pairings," *Proceedings of the 7th International Conference on Cryptology and Network Security*. Springer, pp. 1-17, 2008.

[9] R. Canetti and S. Hohenberger, "Chosen-ciphertext secure proxy re-encryption," *Proceedings of the 14th ACM conference on Computer and communications security*, ACM, pp. 185-194, 2007.

[10] J. Shao and Z. Cao, "Cca-secure proxy re-encryption without pairings," *Public Key Cryptography–PKC'09*, Springer, pp. 357-376, 2009.

[11] X. Liang, et al., "Attribute based proxy re-encryption with delegating capabilities," *Proceedings of the 4th International Symposium on Information, Computer, and Communications Security,* ACM, pp. 276-286, 2009.

[12] T. Fei, et al., "Multi-hop unidirectional proxy re-encryption from multilinear maps," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. 98, pp. 762-766, 2015.

[13] S. Garg, et al., "Candidate multilinear maps from ideal lattices," *Advances in Cryptology–Eurocrypt'13,* Springer, vol. 7881, pp. 1-17, 2013.

[14] D. Hua, et al., "Asymmetric cross-cryptosystem reencryption applicable to efficient and secure mobile access to outsourced data," *Proceedings of the 10thACM Symposium on Information, Computer and Communications Security (ASIA CCS '15),* ACM, pp. 393-404, 2015.

[15] Z. Yunya, et al., "Identity-based proxy re-encryption version 2: Making mobile access easy in cloud," *Future Generation Computer Systems*, 2015.

[16] M. Green and G. Ateniese, "Identity-based proxy reencryption," *Proceedings of the 5th International Conference on Applied Cryptography and Network Security*, Springer pp. 288-306, 2007.

[17] W. K. Koo, et al., "Security vulnerability in a non-interactive id-based proxy reencryption scheme," *Information Processing Letters*, vol. 109, pp. 1260-1262, 2009.

[18] G. Ateniese, et al., "Key-Private Proxy Re-Encryption," *Topics in Cryptology, Springer*, 2009.

[19] K. Liang, et al., "A Ciphertext-Policy Attribute-Based Proxy Re-Encryption with Chosen-Ciphertext Security," *5th International Conference on Intelligent Networking and Collaborative Systems (INCoS),* 2013.

[20] A. Sahai and B. Waters, "Fuzzy Identity Based Encryption," *Springer,* pp. 457-473, 2005.

[21] K. Liang, et al., "An Adaptively CCA-Secure Ciphertext-Policy Attribute-Based Proxy Re-Encryption for Cloud Data Sharing," *10th International Conference, ISPEC 2014, Fuzhou, China*, 2014.