Volume 4, Issue 12, December 2014 ISSN: 2277 128X

International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

A Novel Homomorphic Encryption with Behavioural Attribute Based Fine Grained Data Access in Cloud Computing

Amit Kanungo
Dep. of CSE, LKCT,
Indore (M.P), India

Prateek Nahar
Prof. Dep. of CSE, LKCT,
Indore (M.P), India

Dr. Sanjay Thakur
Vice Principal, LKCT,
Indore (M.P), India

Abstract- Cloud computing is the recent innovation in the field of internet technologies. It supports heterogeneous computing without any managerial loads for the user. In this effective computation is provided in terms of services through policies regarding resource allocation, processing, load sharing, fault tolerance etc. Normally the service ranges from infrastructure, platform and software counted as pay per use basis so as supporting cost is not burdened to the user. Here the number of users accessing these services and their data is large so as it suffers from several issues. Among those issues security is taken to be critical one for providing the isolation and privacy to the user. During the last few years several policies are newly created along with traditional security standards but such mechanism are unable to satisfy the users and providers need. In such environment the users data is placed at third party locations and while securely accessing the computation and overhead loads are increased. This load is due to iterative encryption standards applied while frequently accessing and saving the users changes to the files. Thus some novel standard is suggested in the literature using homomorphism characteristics from which some mathematical operations are performed directly on the ciphertext without decrypting the data. By this the load of the system gets reduced. But its practical implementation is always questioned. This work proposes a novel practically feasible HEBA (Homomorphic Encryption with Behavioural Attribute) schemes for overcoming the above issues. At the analytical level of result evaluation, the suggested approach seems to be providing effective results is near future.

Index Terms- Cloud Computing, Security Service, Homomorphic Encryption with Behavioural Attribute (HEBA), Authentication, User Attributes, Monitoring Service;

I. INTRODUCTION

Outsourcing based services is getting users attentions due to their reduced management and maintenance burdens. Out of those the most famous technology is cloud computing. It is the new era of computing which evolves from some traditional computing paradigms such as distributed, utility, grid and autonomic fault tolerance. According to it, various services can be offered to consumers with some improved controls and less management. One and all can be delivered as a service through browser based programming using networks. It is found that the focus is always towards the data and the service is used to provide the way of doing this. Cloud computing also aims towards scalable and fault tolerance with secure accessibility of user owned data. The service model of the cloud era works towards credibility and control which sometimes might suffers from unauthorized access and data theft or loss. Hence it must provide the way to make the service and data more reliable and secure.

Traditional security mechanism can be used to achieve the goals but somewhere it is not service the complete users and creators need. Identifying the most optimal security technique is always questioned because of their black box detection and measurement phenomenon's. Data confidentiality can be achieved easily at the consumer's location but for storage at third party locations needs to be monitored. It suffers from several open issues needs to be resolved from increasing the users trust over the system. Consumer actually requires the complete isolation of its private information even from the provider. In some of the practical scenarios of service usages, data security and authenticity comes under judicial boundaries and hence requires some robust and improved system. For example, release of medical diagnose information is come under a legal act [1]. Existing mechanism provides the security up to these levels but makes the heavy loads of computational calculations on the servers and machine itself.

In some cases cloud shares the user's information in restricted manner but requires additional protections with control over the provider view of the data. Such controls always are in hands of providers and for further improvements such controls needs to be shift to users. To increase the security with transferred controls to users, a new mechanism needs to be adopted with traditional options. There are numerous issues associated with access controls & data isolations for cloud consumers about cloud security such as: loss of control over cloud hosted assets, lack of security guarantees in the SLAs between the cloud provider and cloud consumer; sharing of resources with competitors or malicious users.

Increasing the security at third party location will also affects the normal actions of the data access because each time a user changes its data then encryption is performed with a decryption and if the modifications are very often then this process will seems to make the system overloaded. Also the cloud service will be reached end users through various intermediate brokers and if traditional encryption is performed, lot of effectors and dimensions are wasted. This migration follows multitenant model and cloud computing is bringing remarkable impact on information security fields.

Such issues generated because of dynamic scalability, service abstraction and location transparency [2]. Also some times is quite complex to perform the operations on the encrypted data even with a small change. The solution is to use a homomorphic encryption rather than traditional methods. By using this mathematical operations are performed on ciphered data. But it is still a feasibility issue with practically achieving the homomorphism characteristics. To achieve the security goals as mentioned in the paper several other potions with variable encryptions are available.

This paper proposes a novel HEBA policy based work to provide higher security with less managerial concern. The work is an extension of security approach which uses homomorphic encryptions. The work will also consider the event of a security breach, which overcomes from data isolation issues. The work also analysed the existing service delivery models of cloud computing and identifies that the resources of cloud services based on may be owned by multiple providers.

II. BACKGROUND

Information system requires effective way of processing and transferring the desired knowledge to the end user through some verification means by which user's assurance towards the data security is increased. The approach must satisfy the growing user's needs and elastic service demands. Security can be overlooked by compliances and audits verifications as mentioned in [3]. The information loss can be planted or natural. If the losses occur due to some planted operations by which creditability of the provider is decreased and user's confidentiality goes down. Such process of known drops are comes under the categories of attacks. Here the aim of attacker is to steal some confidential information of the user or organization on which major financial decisions depends. To guide desired data access by authorized users, several authentication mechanism is developed. But for cloud environment such mechanism needs to be made sounder. Such process is known as fine grained access control systems which assist conceding differential access rights to a group of users and allow flexible operational services in specifying the access rights of several users. Several mechanisms are known for practically implementing the concept of fine grained data access control process.

Along with fine grained access control some of the control needs to be transferred to the end user for more robust and trust based security. Security of any standard depends mainly on the key size and type. It could be generated from the several mechanism of public and private key. In third party storage, the users lost control over its data and the dependency is generated over the cloud provider or service provider. But as the users key needs to be from user side, existing key generation policies must be modified. Setting key with the user's properties in cloud is controlled by an intermediate function in virtual environment known as data obfuscation [4]. Thus a user is equipped with some controls of key generation policies. So for applying the encryption the user holds the key and not the cloud provider to increase the protection over the cloud data. But for providers view such keys are inconsistence with their existing business models. This architecture limits a cloud provider's ability to data mine or otherwise exploits the users' data. If a provider does not have access to the keys, they lose access to the data for their own use.

Also along with access control, confidentiality can be provided by encryption mechanism. Traditional mechanism is not suitable for cloud environment because of their overloaded computational operations and restricted options of performing some modifications directly through ciphertext without decrypting the data. During the last few years several research papers are published which satisfy the requirements. Among them, homomorphic encryption is the most suitable option.

Understanding Homomorphic Encryption (HE)

Encryption is used to protect data against eavesdroppers who would otherwise intercept private communication. One party encrypts a message and sends the corresponding cipher text to a second party, who then decrypts the ciphertext to recover the message. To prevent an un-trusted third party from eavesdropping, the problem of recovering any information about the message from the ciphertext should be reasonably hard; in addition, the cipher text should itself reveal no information about the message. Increasingly, data storage and computation is outsourced to these un-trusted parties, which gives rise to the need for an encryption scheme that allows computation on the ciphertexts. Homomorphic encryption schemes are malleable by design. The homomorphic property of various cryptosystems can be used to create secure voting systems, collision-resistant hash functions, and private information retrieval and enable widespread use of cloud computing by ensuring the confidentiality of processed data. There are several efficient, partially homomorphic cryptosystems, and a number of fully homomorphic, but less efficient cryptosystems. Although a cryptosystem which is unintentionally homomorphic can be subject to attacks on this basis, if treated carefully homomorphism can also be used to perform computations securely.

Informally speaking, a homomorphic cryptosystem is a cryptosystem with the additional property that there exists an efficient algorithm to compute an encryption of the sum or the product, of two messages given the public key and the encryptions of the messages but not the messages themselves. The homomorphic properties of various encryption schemes have been a fascination of the cryptographic community for decades. With the rise of cloud computing and decentralized processing, the need for security in such applications is increasing. Only recently, however, has the construction of a fully homomorphic encryption scheme been realized.

Some of the improved security framework suggested by industry using SLAs and regular assessment and monitoring is achieved by NIST-FISMA. It provides trust based computing with improved privacy and security. A detailed description of the approach is given in [5]. Some of the cloud security requirements are mentioned here are:

(i) Data safe storage by Cloud providers
(ii) Managed privacy adequately

(iii) Following laws and regulations by Cloud providers [6]
(iv) Minimum disruption of business or outage
(v) Effective protection against cyber attacks by the cloud provider [7]

Getting effective solutions enables researchers and security professionals to know about users and vendors' concerns and critical analysis about the different security models and tools proposed [8]. Thus to clearly understand the above defined requirements, this paper comprises of an aggregated study over existing encryptions and security schemes from which some of the unsolved issues can be identified. After a clear analysis, the paper also proposes a solution to overcome the issue.

III. LITERATURE SURVEY

Considering outsourced data storage environments with cloud computing, there are several approaches which had been proposed to resolve the existing security breaches. While optimal solutions searching also leads the researcher towards homomorphic encryption. The single security method cannot solve the cloud computing security problem. The combinations of various existing and new technological strategies must be used together for protecting the total cloud computing system. Cloud computing should provide a strong user access control which powers the licensing, certification, quarantine and other aspects of data management [9]. But still there are some issues and things which have to be opened and presented here as literature survey.

In the articles [10], NIST suggested some of the guidelines for security enhancement over the existing and newer applications of cloud and other secure computing. It gives various aspects of making a cloud application more secure. The guideline distributes their activities in the form of recommendation to provide & users. It identifies security, privacy, and other organizational requirements for cloud services to meet, as a criterion for selecting a cloud provider. It will also analyze privacy controls by providers and assess the level of risk associated with commitment to deliver cloud services over the target time frame.

Some of the authors worked with improving the key management structure for improving the security of the outsourced data as mentioned in [11]. It gives life cycle to the cryptographic operations using assigned keys of user's behaviours. Such features are based on some of the networked characteristics and follow some generation principles. Their applications are from cloud oriented systems to modern healthcare by which protection on data can be assured even at distributed locations. The organization used to provide such functionalities are termed as OASIS key management interoperability protocol.

While working towards making the user's personal information secure from the third party executor, data access needs to be defined properly with all the desired controls. These controls are of type authentications and authorizations. In traditional systems dynamically changing environment are not handled effectively due to the nature of cloud computing. Even, the systems are always susceptible to lots of impersonation attacks and server spoofing and sniffing. For overcoming such issues the paper [12] proposes a novel identity based approach through various phases of registration, login key based access and password protection from some unauthorized user with the same behaviour. At the analytical level of evaluation the approach seems to be effective then its competitors.

Further elaboration of security stacks and service is achieved with improved performance homomorphic encryption (HE). The HE security is enabled by some networked locations with confidentiality to perform operations on encrypted data. Hence the meaningful computation is applied on arbitrary encrypted data. The suggested approach in [13] is the extension of the existing ring based learning encryption with some the major functions. Practical analysis and evaluation of the approach gives effective results in the form of short ciphertext. The developed approach is termed as RLWE and the tool used to achieve the goal is MAGMA. It is somewhat homomorphic encryption but for implementing the overall characteristics fully homomorphic encryption is best suited. The approach keeps data private, but also permits a worker that does not have the secret decryption key to compute any (still encrypted) result of the data, even when the function of the data is very complex [14].

It is known to be complex, achieving the complete property of homomorphism. But still various paper are trying to implement the same using several options such as ElGamal encryption, pailliers methods etc. In a way to do so, secure data sharing (SDS) is a proxy based re-encryption which prevents the data leaks through unauthorized access [15]. The approach provides security definitions in term of secure multi-party computation (SMC) using generic HE based sub routines. The re-encryption here mans reducing the encryption load by applying the computational operations of intermediate cipher. Also, the results of the approach show that it avoids the uncertainty and collusion between the cloud provider and user.

In the article [16] a novel protocol KMIP (Key Management Interoperability Protocol) is proposed using behavioral element based key generation and management. The aim is to make the distribution secure between the various cloud elements including the user. Here the client is equipped with a wide range of applicable keys on the data. The provider is not aware about the key selected with explicitly or implicitly defined policies by the client application and the server. It replaces the existing redundant and unmanaged key management protocols by a new trusted service of key generation and distribution using KMIP.

Various organizations are offering the cloud based service for providing the services following the structure of web 2.0. It satisfies the user's demands of scalability with security. Amazon Web Services (AWS) is one of those providers which offers improved computing with optimal resource consumption and management. It delivers a secure cloud computing platform with high availability and dependability [17]. It also offers the flexibility to enable customers to

build a wide range of applications with high confidentiality and integrity. Specifically, AWS physical and operational security processes are described for network and server infrastructure under AWS's management, as well as service-specific security implementations.

IV. PROBLEM STATEMENT

Cloud computing is the recent are of the work because of its frequent growth in number of users demanding scalability with security. As the number of options gets integrated with cloud, handling of security becomes more complex. User wants maximum availability with minimum dependability in highly secured environment. Such requirement can be made feasible with cryptographic primitives. Traditionally these options are more applicable for single end or dual end security, but for cloud environment only cryptography is not sufficient. Security and availability is not fully guaranteed with existing algorithms. Some of the identified issues that is unable to be achieved with existing security options are:

Issues 1: In cloud computing users is not having any control at third party locations of data storage. Thus, somewhere the users trust over the system is not that much good? Users have to be in some control of making the data secure.

Issue 2: Normally the user access data many times and same number of time security algorithms work with data. Thus, traditional algorithm for making it secure is quite complicated which increases heavy computational load on the network and servers.

Issue 3: User is not allowed to deal with encrypted data, and each time the data is demanded decryption and re-encryption is applied. Thus it is required to provide some control to the user by which mathematical operations are directly applied on ciphered data.

Issue 4: One of the obligatory aspects of data security is to create models of requirements intensity of security. It gives the degree up to which it needs to be protected. Protection level are taken to be as transmission of the file using encrypted protocols, access control of the file itself, but without encryption of the content, access control (including encryption of the content of a data object) and access control (including encryption of the content of a data object) also including rights management options (for example, no copying content, no printing content, date restrictions, etc.).

Thus various approaches are attempted to provide multiple cloud security control and some for them achieved in their goal. But still practical development of solution and evaluations of such approach is not drafted with proper rules. This work proposes a novel HEBA based improved cloud security.

V. PROPOSED HEBA APPROACH

This work proposes a novel HEBA (Homomorphic Encryption with Behavioural Attribute) based security scheme for cloud computing. It gives optimal computational load for dynamic data encryption using some of the user's behaviour elements, guided key generation supports. The approach also verifies integrity of the data throughout the access. Traditionally the user store data on at third party location where the security is handled by the provider without any user's interaction. In such system users trust over the system gets reduced sometimes when the data is not accessible or due to some leakage of the data. For making it secure the provider encrypt the data by using traditional encryption standards. Robustness of the encryption algorithms depends on the key size and with existing system its generation is also in control of the provider. So this work suggested modifications and providing some control to user in key generation process. Here the wok uses the users behavioural properties and generate the key from that elements implicitly or explicitly.

For cloud computing handling of multi-tenancy model is a critical task because of its fine grained data access control property. The overall process of users data access and verification of user and service provider is performed by auditor whose aim is towards monitoring activity. The auditor also monitors the resource consumption by different process initiated by user or provider on normal reading. If the consumption is higher than normal range then some attack event initiation is confirmed. Such monitoring will reduces the computation load with improved security.

While taking the performance and ease of accessibility, traditional encryption makes the process more complex. User access the data from its stored location and dynamically does changes as required. Each time a change occurs re-encryption is performed by which the load on machine is increases and the resource consumption raises the uses bill. For overcoming these, the work uses a property of homomorphism by which arbitrary computation can be performed directly of encrypted data. It assures data confidentiality without decrypting the data for smaller mathematical operations. It drastically reduces the storage and computational load. The approach is also capable of verifying the stored and accessed data by providers and hence the error localization can be easily measured. The provided mechanism can be verified after implementing the approach using the suggested algorithms

Steps of Implementing Solution through HEBA

(i) Start Approach At cloud Storage
(ii) Cerate user login for Credential based key Generation
(iii) Convert the credential information to hash code using MD5 algorithm
(iv) Fetch user behaviour attributes from its profiles.
(v) Convert attributes to digest again by using MD5
(vi) Concatenate intermediate compositions for Comprehensive key
(vii) Measure the time of key generation and its size along with complexity
(viii) Select the cloud services

(ix) Upload the file and encrypt using the above comprehensive key
(x) Check resource consumption before and after the service usage
(xi) Load data for modifications
(xii) Start Paillier Homomorphic Encryption Algorithm
(xiii) Perform intermediate computation on ciphertext directly without decrypting the data.
(xiv) Retrieve the data when the key is same for the users attributes
(xv) At third party monitoring the overall process is monitored through memory and CPU cycles.
(xvi) Exit

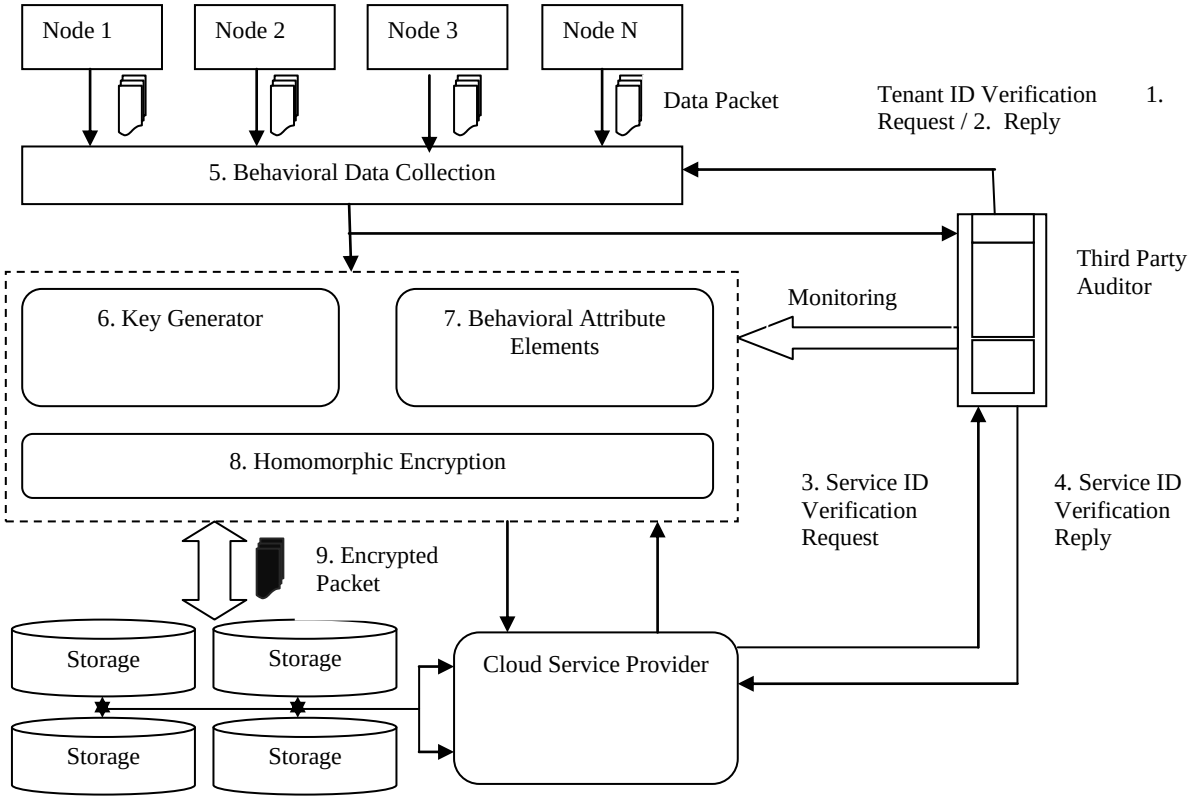

FIGURE 1:- PROPOSED HEBA CLOUD SECURITY SERVICE ARCHITECTURE

Components: The approach mainly deals with authentication, key generation, service orientation, data obfuscation and monitoring. Here the authentication deals with converting the formal credential code to have formats for integrity based verifications. These codes are stored at trusted third party locations from which only authorized user can access the data. Even the provider is unaware about the content of the storage location. Second is the key generation using some of the user detected properties of its behaviour or attributes of its profile is used. Form this step the user is provided some controls on encryption by key generation. The normal size of the key is 128 bits and is in hexadecimal format. For evaluating this generation is measure on various parameters such as generation time, type, size etc. After this key is generated it is applied to homomorphic based Pailliers encryption.

In the Paillier is a partially homomorphic cryptosystem, if the public key is the modulus m and the base g, then the encryption of a message x is $\varepsilon(x) = g^x r^m \bmod m^2$, for some random $r \in \{0, \ldots, m-1\}$. The homomorphic property is then

$$\in (x1). \in (x2) = (gx1r1m)(gx2r2m) = gx1 + x2(r1r2)m = \in (x1 + x2modm)$$

Using homomorphic cryptosystems the encrypted function can be evaluated which guarantees its privacy. Homomorphic schemes also work on encrypted data to compute publicly while maintaining the privacy of the secret data.

This can be done encrypting the data in advance and then exploiting the homomorphic property to compute with encrypted data.

After converting the Pailliers code the data is stored at different locations. While retrieving the same data, this code is not decrypted and instead of that direct arbitrary operation is performed on it. The uploaded data is not accesses without the user based key. Generation of this key is not possible by any of the mechanism because it depends on the users properties. Until all the properties matches, the key is not generated and the data is not downloaded. The system has various log information like system status, failed login attempts, integrity checks & verification ID. The system monitors this regularly to detect uneven behaviour for fault localization. It uses adversary monitoring through a weak or strong bond. Weak adversary is interested in corrupting the user's data files stored on individual servers. Once a server is comprised, an adversary can pollute the original data files by modifying or introducing its own fraudulent data to prevent the original data from being retrieved by the user. While strong adversary is the worst case scenario, in which we assume that the adversary can compromise all the storage servers so that he can intentionally modify the data files as long as they

are internally consistent. In fact, this is equivalent to the case where all servers are colluding together to hide a data loss or corruption incident. This security is increases with reduced computational loads and overheads.

At the analytical level of evaluation, approach seems to be effective and well performed than existing mechanism. The approach also reduces the computation load with better monitoring and resource consumption analysis.

VI. APPLICATIONS & BENEFITS

Application of HEBA and Homomorphic Encryption

- *Protection of mobile agents:* The protection of mobile agents by homomorphic encryption can be used in two ways: Computing with encrypted functions and Computing with encrypted data. Computation with encrypted functions is a special case of protection of mobile agents.
- *Multiparty computation:* In multiparty computation schemes, several parties are interested in computing a common, public function on their inputs while keeping their individual inputs private. This problem belongs to the area of computing with encrypted data. It can be solved with HE.
- *Threshold schemes:* Both secret sharing schemes and the multiparty computation schemes are examples of threshold schemes. Threshold schemes can be implemented using homomorphic encryption techniques.
- *Zero-knowledge proofs:* This is a fundamental primitive of cryptographic protocols and serves as an example of a theoretical application of homomorphic cryptosystems. Zero knowledge proofs are used to prove knowledge of some private information. Zero-knowledge proofs guarantee that the protocol communicates exactly the knowledge that was intended, and no (zero) extra knowledge.
- *Election schemes:* In election schemes, the homomorphic property provides a tool to obtain the tally given the encrypted votes without decrypting the individual votes. Watermarking and fingerprinting schemes:
- *Oblivious transfer:* It is an interesting cryptographic primitive. Usually in a two-party 1-out-of-2 oblivious transfer protocol, the first party sends a bit to the second party in such as way that the second party receives it with probability ½, without the first party knowing whether or not the second party received the bit.
- *Commitment schemes:* Commitment schemes are some fundamental cryptographic primitives. In a commitment scheme, a player makes a commitment. Some commitment schemes can be efficiently implemented using homomorphic property.
- *Lottery protocols:* Usually in a cryptographic lottery, a number pointing to the winning ticket has to be jointly and randomly chosen by all participants. Using a homomorphic encryption scheme this can be realized as follows: Each player chooses a random number which she encrypts. Then using the homomorphic property the encryption of the sum of the random values can be efficiently computed. The combination of this and a threshold decryption scheme leads to the desired functionality.

Security Benefits

In order to measure and compare the performances of the proposed HEBA scheme, the work continues to adopt the various comparison metrics, First is key size which is very less in case of homomorphic encryption. Second is secure data access mechanism. The work makes the following observations about the proposed work:

Continue Operations with Data confidentiality: from Un-authorized user's identification at early stages matches the user attributes and gives access in accordance to that. Also the homomorphism characteristics lets the user works on encrypted data without decrypting it always even for a small changes.

On-demand revocation with Less Response Time: can be possible by using user HE and provides fine grained data access mechanism with any third party servers.

Effective Write access control from unauthorized user different from data owner group can be given to enhance further security though revocable homomorphic encryption.

Demand Based Elastically Scalable and efficient Usability based domains to provide the user isolation of work area and data separation. It should handle all the complexity issues to make the approach more effective.

VII. CONCLUSION

Cloud computing involves various issues related to dynamic data handling and security because of its outsourced service based distributed architecture. As, the number of user is increasing in cloud computing, security is becoming more crucial aspect. Thus, the purpose of suggested HEBA is to identify the security breaches and develop an improved approach by which protection is increased but the computational load with its respect is decreased. The approach provides various polices related to user trust increments by controlled key generation, monitored services, allowed computation on encrypted data etc. It is made feasible by practically applying the homomorphism characteristics. Unlike most prior works for ensuring remote data integrity, the new scheme supports secure and efficient dynamic operations on data blocks, including: update, delete and append. Extensive security and performance analysis shows that the proposed scheme is highly efficient and resilient against malicious data modification attack, and even server attacks.

ACKNOWLEDGEMENT

The authors wish to acknowledge LKCT administration for their support & motivation during this research. The authors would also like to thank anonymous referees for their many helpful comments, which have strengthened the paper. They also like to give thanks to Dr. Sanjay Thakur, Mr. _____ and _____ for discussions in specific domain.

REFERENCES

[1] Shucheng Yu, Cong Wang, Kui Ren and Wenjing Lou, "Achieving Secure, Scalable, and Fine-grained Data Access Control in Cloud Computing", in Proceedings of IEEE Infocomm, ISSN: 978-1-4244-5837-0/10, 2010.

[2] Deyan Chen and Hong Zhao, "Data Security and Privacy Protection Issues in Cloud Computing", in International Conference on Computer Science and Electronics Engineering, IEEE 2012, ISSN: 978-0-7695-4647-6/12, DOI 10.1109/ICCSEE.2012.193, 2012.

[3] Daryl C. Plummer, Thomas J. Bittman, Tom Austin, David W. Cearley and David Mitchell Smith, "Cloud Computing: Defining and Describing an Emerging Phenomenon", Research Article in Gartner Research, ID Number: G00156220, June 2008.

[4] Stephen S. Yau and Ho G. An, "Confidentiality Protection in Cloud Computing Systems", in International Journal of Software and Informatics, ISSN 1673-7288, Vol.4, No.4, December 2010, pp. 351-365.

[5] Mohemed Almorsy, John Grundy and Amani S. Ibrahim, "Collaboration-Based Cloud Computing Security Management Framework", in IEEE 4th International Conference on Cloud Computing, ISSN: 978-0-7695-4460-1/11, DOI 10.1109/CLOUD.2011.9, 2011.

[6] Pardeep Kumar, Vivek Kumar Sehgal , Durg Singh Chauhan, P. K. Gupta and Manoj Diwakar, "Effective Ways of Secure, Private and Trusted Cloud Computing", in International Journal of Computer Science Issues, ISSN: 1694-0814, Vol. 8, Issue 3, No. 2, May 2011, pp 412-521

[7] Farzad Sabahi, "Cloud Computing Security Threats and Responses", in IEEE Transaction, ISSN: 978-1-61284-486-2/11, 2011.

[8] Farhan Bashir Shaikh and Sajjad Haider, "Security Threats in Cloud Computing", in IEEE 6th International Conference on Internet Technologies & Transactional databases, ISSN: 978-1-908320-00-1/11, UAE, 2011.

[9] Wentao Liu, "Research on Cloud Computing Security Problem and Strategy", in IEEE Transaction, ISSN: 978-1-4577-1415-3/12/, 2011.

[10] Wayne Jansen and Timothy Grance, "Guidelines on Security and Privacy in Public Cloud Computing", in NIST Special Publication 800-144, Dec 2011.

[11] Christian Cachin, Divay Bansal, Gllunter Karjothm, "Key Management with Policy-based Access Control", in IBM Research, April 2012.

[12] Dianli GUO and Fengtong WEN, "A More Secure Dynamic ID Based Remote User Authentication Scheme for Multi-server Environment", in Journal of Computational Information Systems, ISSN; 1553–9105, Vol. 9:No. 2, 2013, 407-413

[13] Kristin Lauter, Michael Naehrig and Vinod Vaikuntanathan, "Can Homomorphic Encryption be Practical", in ACM, 2008.

[14] Craig Gentry, "Computing Arbitrary Functions of Encrypted Data", in ACM by IBM T.J. Watson Research Center, 2008.

[15] Bharath K. Samanthula, Gerry Howser, Yousef Elmehdwi, and Sanjay Madria, "An Efficient and Secure Data Sharing Framework using Homomorphic Encryption in the Cloud", in Cloud 1st conference of the ACM, ISSN: 978-1-4503, DOI: 1596-8/12/08, 2012.

[16] Robert Griffin and Subhash Sankuratripati, "Key Management Interoperability Protocol Profile Version 1.1", in OASIS Standards Organizations at http://docs.oasis-open.org/kmip/profiles/v1.1/os/kmip-profiles-v1.1-os.doc, 2013

[17] Web Article, "Amazon Web Services: Overview of Security Processes" by Amazon Services at http://aws.amazon.com/security,June 2013.

