I.J. Modern Education and Computer Science, **2014, 10, 1-9**
Published Online October 2014 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijmecs.2014.10.01

Performance Evaluation of TPA-HE Based Fine Grained Data Access for Cloud Computing

Pawan Kumar Parmar
SIMS, Department of computer science, Indore, 452009, India
Email: pawan.parmar1@gmail.com

Megha Patidar and Mayank Kumar Sharma
SIMS, Department of computer science, Indore, 452009, India
Email: Jia.meg@gmail.com, mayank.sharma@sims-indore.com

Abstract—As the software technology evolves the focus of users are shifting form devices for data or information. This transformation requires reliable and scalable computing paradigms which satisfy the users processing and storage requirements. Service based, distributed, grid and web 2.0 are some of the most famous computing technologies. Conversions are occurring towards less managements and maintenance issues and despite of that the usage experience should be increased. But there are some security concerns like security, access control, privacy & isolation based trusted service delivery raises due to the data in an outsourced environment. Thus, several policies are created to define its boundaries. Also the type of user accessing the data and the service provided by the cloud needs to be verified. Thus the uses trust over the system can go down if the interoperability and security of services are satisfactory. To providing confidentiality to users data encryption is the traditional options which require decryption for reading or retrieving the data. But in outsourced environment the user is frequently accessing its data which may increase the overhead of performing such frequent encryption and then decryptions. Also for performing any operations the data need to be decrypted. It is something treating as a complex usage boundary. Thus, Homomorphic encryption is used to deal with such situations. This paper proposes a novel Third Party and Homomorphic Encryption (TPA-HE) based mechanism for secure computing. In this third party auditor and service provider is used for authentication and authorization of services & user profiles. It has three basic entities TPA, Cloud Service Provider, Encryption & Monitoring service to regularly analyze the security breaches in access & data transfer mechanism. To prove the effectiveness of suggested approach some of the results are taken which are better than the existing mechanism.

Index Terms—Cloud Computing, Security Service, TPA-HE (Third Party and Homomorphic Encryption), Authentication, User Attributes, Monitoring Service.

I. INTRODUCTION

Cloud computing is an overwhelming technology used to reduce the users operational and management load by introducing and comprehensive mechanism of distributed, grid, autonomic computing. This area is gaining popularity due to its wide applicability like client server and other browser dependent programming. It includes the delivery of various computing and storage aspects as a service to the end users. Measuring benefits among all security services is considered as one of the high priority open issues in adopting the cloud computing model. This service model faces a number of open issues that impact its credibility. Data confidentiality against cloud servers is hence repeatedly desired when users outsource data for storage in the cloud. Thus the security issues are generated because of these low trusted outside processing entities such as providers. Thus the trust factors at such services are very low. The consumer always likes to make its data & service in an isolated manner from external persons. In few practical situations of service application systems the data confidentiality will come under juristic boundaries by taking their security issues. For example, disclosure of Healthcare information from company to consumer is a legal act [1]. Thus, it needs to be taken as more secure data and when it is handled by providers, there is an always way to make is open. Thus to make the system more reliable client needs to make some security trusted deals with its data.

These operations are grouped into three main phases:

- *Security Requirements Definitions:* This phase explicitly elaborates the users and providers' security required for a related service or user category. It can be applied in assessing various security essential features provided by the creators.
- *Applying Security Requirements:* It includes the identification of policies related to the requirement so as to make the data transition more secure. It uses fine grained access control for satisfying the security policies.
- *Monitoring:* This phase includes the continuous monitoring of services and their usage pattern by some auditors. Usually it identifies the frequency of information change in the system. It also

analyzes the measured security status to identify existing security issues.

In some cases the cloud user will share their data among other consumers, but in a restricted access manner. There are several issues associated with access controls & data isolation for cloud consumers about cloud security, such as the loss of control over cloud hosted assets, lack of security guarantees in the SLAs between the cloud provider and cloud consumer; and haring of resources with competitors or malicious users.

Thus, in this paper a novel TPA-HE policy based work is proposed to provide higher security with less concern management. It focuses on features of cloud computing models for all kinds of applications and data on the cloud platform which have no fixed infrastructure and security boundaries. The work will also consider the event of a security breach, which overcomes from data isolation issues. The work also analyzed the existing service delivery models of cloud computing and identifies that the resources of cloud services based on may be owned by multiple providers. Thus the work also proposes a novel security model with enhanced mechanism.

II. Background

Your Effective storage and retrieval requires flexible computation to be performed on each transform to satisfy the users increasing demands of information systems. To reduce the users end computation load the information and processing capability needs to be transformed to some other trusted locations. These locations or servers are using shared medium and reduces risk in areas such as data integrity and privacy and need to recognize issues in areas such as e-discovery, compliance and audit reporting [3]. Various security solutions have been proposed to handle such privacy & confidentiality concerns among them encryption is widely accepted. So middleware functions data obfuscation is added on virtual machine to provide such security service. A Data De-Obfuscator de-obfuscates obfuscated data so that a user can see the plain data. A Data De-Obfuscator remains in the user's personal computer all the time [4]. Compensating controls by the provider is taken periodically by using multiple SLAs and regular assessment of capabilities, or new audit or monitoring functions is required. Moreover, the lack of security constraints in the Service Level Agreements between the cloud providers and consumers results in a loss of trust as well. Obtaining a security certificate such as ISO 27000 or NIST-FISMA would help providers to improve consumers trust in their cloud platforms security [5]. It is bounded by many security issues like data privacy and service availability which can be measured by calculating the utilization of processing and storage devices are discussed in [6, 7] and their comparison is made in [8].

Wentao [9] explain that 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. The users do not know what position the data and do not know which servers are processing the data. It also does not have any information about network used for transmitting the data due to its scalable & flexible nature. The different locations will also sustain various security laws about the data privacy in a confidential way. To ensure that the security guideline is being given by NIST [10]. 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.

A key management system performs vital functions for the secure operation of cryptographic systems, in particular for governing the life cycle of cryptographic credentials. Although usually proprietary today and tied to particular hardware incarnations, such key-management systems are expected to become network-enabled, open, and standard-based in the future. They will address the needs of cryptographically protected cloud computing services; more importantly, the key-management systems of the future will operate from the cloud. Open key-management systems need a common language. Cachin et al. [11] recently developed OASIS Key Management Interoperability Protocol (KMIP) standard establishes a single, comprehensive protocol for the communication between enterprise key-management systems and cryptographic services.

Taking user privacy for confidential data needs the user access definition in the first phase. It can only be achieved by authentication mechanism. But using traditional mechanism for dynamically changing scenarios is not effective. The traditional mechanism is scheme is still susceptible to impersonation attack and server spoofing attack. So a new counter measure has been proposed by Dianli *et al.* [12] by dynamic identity based authentication. It suggests the changes in all the phases include registration, log-in, authenticating session key agreement and password change. Thus, the patch is fortunately simple to remedy the security flaws of the scheme.

After the user gets authenticated, data needs to be provided in secured manner. This can be achieved by applying encryption. The privacy concerns can be satisfactorily addressed if users encrypt the data they send to the cloud. If the encryption scheme is homomorphic, the cloud can still perform meaningful computations on the data, even though it is encrypted. This encryption scheme allows arbitrary computation on encrypted data. It has a basic extension of the scheme ring learning with encryption (RLWE) and uses five major functions to do effective computation [13]. The scheme implementation in MAGMA gives an efficient and reasonably short ciphertexts. The further usage & modification of fully homomorphic encryption (FHE) is explained by Gentry [14]. Here the fully homomorphic encryption scheme

means that it keeps data private, but that allows 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.

Bharath *et al*. [15] the usage & behavior of homomorphic encryption are thoroughly defined. They also propose an efficient and Secure Data Sharing (SDS) framework using homomorphic encryption and proxy re-encryption schemes that prevent the leakage of unauthorized data when a revoked user rejoins the system. This framework is secure under the security definition of Secure Multi Party Computation (SMC) and also is a generic approach - any additive homomorphic encryption and proxy re-encryption schemes can be used as the underlying sub-routines.

Griffin *et al.* [16] propose a novel protocol Key Management Interoperability Protocol (KMIP) for improved security is formulated. It is a single, comprehensive protocol for communication between clients that request any of a wide range of encryption keys and servers that store and manage those keys. By replacing redundant, incompatible key management protocols, KMIP provides better data security while at the same time reducing expenditures on multiple products.

Many organizations are recently offering various solutions for such security breaches in a specific manner of security as a service. Among them Amazon Web Services (AWS) notifies their keen presence in the market. It delivers a scalable cloud computing platform with high availability and dependability, offering the flexibility to enable customers to build a wide range of applications [17]. Helping to protect the confidentiality, integrity, and availability of our customers' systems and data is of the utmost importance to AWS, as is maintaining customer trust and confidence. 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.

III. Problem Identification

Primarily there are two areas required for working on the security of the greatest concern of organizations migrating to cloud services. Can security really be guaranteed and will the service always be available? The problem is if a cryptography – entirely based on trust – could be implemented as a cloud service, in which – in many opinions – security, trust and availability cannot be fully guaranteed. From the perspective of data security, which has always been an important aspect of quality of service, Cloud Computing inevitably poses new challenging security threats for a number of reasons.

1. *Firstly,* traditional cryptographic primitives for the purpose of data security protection cannot be directly adopted due to the users' loss control of data under Cloud Computing. Therefore, verification of correct data storage in the cloud must be conducted without explicit knowledge of the whole data. Considering various kinds of data for each user stored in the cloud and the demand of long term continuous assurance of their data safety, the problem of verifying the correctness of data storage in the cloud becomes even more challenging.

2. *Secondly,* Cloud Computing is not just a third party data warehouse. The data stored in the cloud may be frequently updated by the users, including insertion, deletion, modification, appending, reordering, etc. To ensure storage extent up to which it needs to be secure. This can be categorized according to its granularities given as:

Level 1. Transmission of the file using encryption protocols
Level 2. Access control to the file itself, but without encryption of the content
Level 3. Access control (including encryption of the content of a data object)
Level 4. 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.) correctness under dynamic data update is hence of paramount importance.

These techniques, while can be useful to ensure the storage correctness without having users possessing data, cannot address all the security threats in cloud data storage, since they are all focusing on the single server scenario and most of them do not consider dynamic data operations. As a complementary approach, researchers have also proposed distributed protocols for ensuring storage correctness across multiple servers or peers. Again, none of these distributed schemes are aware of dynamic data operations. As a result, their applicability in cloud data storage can be drastically limited.

IV. Proposed TPA-HE Approach

The work evaluates a novel distributed security scheme for effective encryption of dynamic data of users. Here a novel Third Party Auditor and Homomorphic Encryption (TPA-HE) [18] based mechanism for secure computing is evaluated and proves the efficiency of using auditing and monitoring mechanisms with security. In this third party auditor is used for authentication and authorization user profiles. It has three basic entities TPA, Cloud Service Provider, Encryption & Monitoring service to regularly analyze the security breaches in access & data transfer mechanism. To prove the effectiveness of suggested approach some of the results are taken which are better than the existing mechanism. It ensures the correctness of data by thoroughly verifying the type of information passed by the user & is provided by the server. It uses an explicit third party auditor to perform the desired task. This third party auditor will verify the user & auditing data on demand. This comes under the phase of authorization. It uses multi-tenancy identifications of service & user both.

It will also reduce the storage load in an effective & secure manner. The work uses a specific type of

encryption known as homomorphic encryption by which security can be verified on the encrypted text without decoding the text. So far this work has covered user authentication and authorization. This category concentrates on securing the communication between the user and a service provider. These methods do not apply to database providers, which execute within the provider's database. If the communication is not secured, it is possible for someone to reproduce a portal and fool the web provider into returning sensitive information.

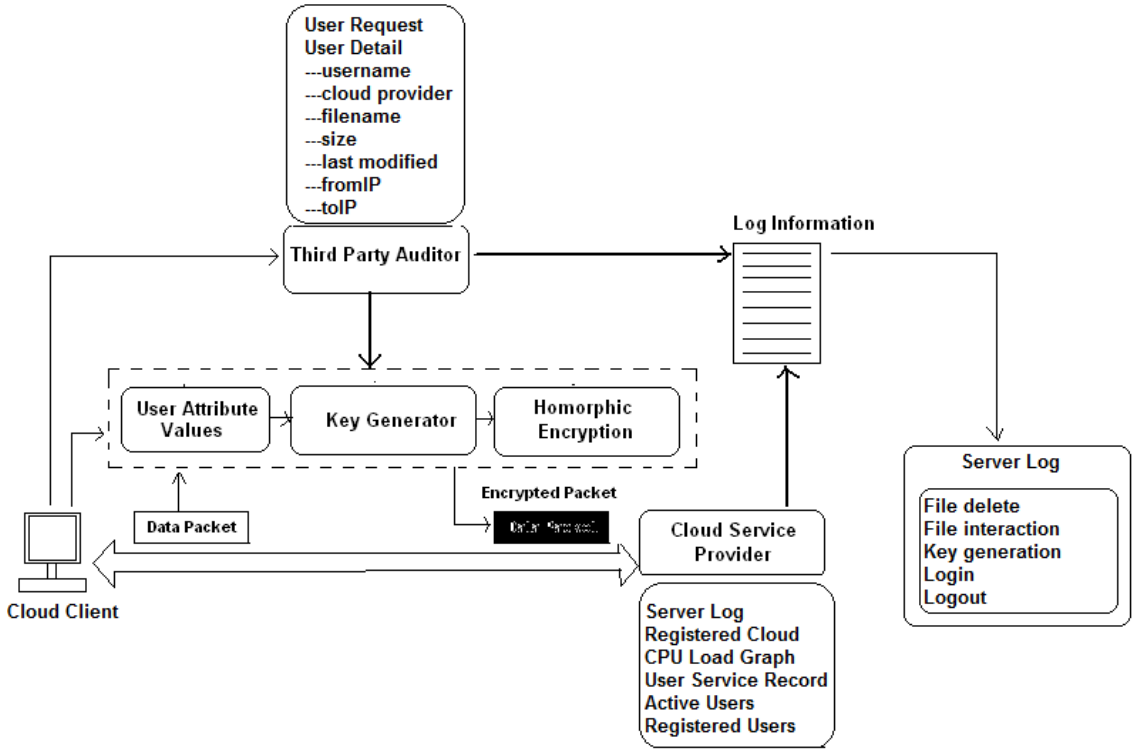

Fig 1:- Modified Cloud Security Service Architecture for TPA-HE

The work basically governs the security through three component communication security:

1. Third Party Auditor
   a. User Authentication
   b. Auditing data

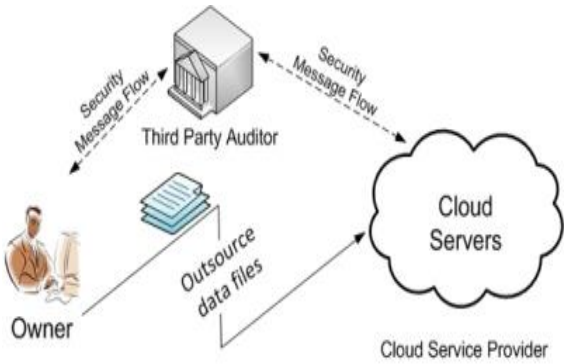

Fig 2:-Cloud Data Storage Architecture

2. Cloud Service Provider
3. Data Encryption & Monitoring Service

Algorithm Proposed TPA-HE

Algorithm
Initiate TPA-HE ()
{
User Login ();
//Creation of Credentials is Required For New User Create
//Information is sent to TPA
//Third Party Auditor Authenticate user
{Fetch Current Users Attribute Elements (Att1, Att2 ….Att N)
KeyGen (Att1, Att2 ….AttN);
//Key Based on Attributes
Generate Digest (MD5 (KeyGen ());//Digest View
TimeMeasure (KeyGen ());
//Key Generation Time Calculations
SizeMeasure (KeyGen ());
//Key Size Calculations
UpdateServiceDetails (ControlsInfoUser ()→ServiceProvider());
//Transferring Information to Auditor
FHEncryption-Palliers (KeyGen (), Data (M));

//Applying encryption based on Generated Key on Data of message M
FHPacket→ServiceProviders ();
//Sending Data Packets to Server
FHRetrieve (PassKeyGen (), Cipher (*)) →Message (M);
//Retrieving Original Message by Passing the Attribute Based Key
}}
ServiceProvider ();
{
CheckStatus (User1, User2….. User N);
//Checking Users Status Connected to Providers or Auditors
CheckServiceUsed (User 1(Ser1, Ser2…SerN)….UserN (Ser1, Ser2…SerN));
//Checking service Used by Different Users
CheckLoad (Memory, CPU Cycles);
//Before and After Service
Print ((Before Service Value, After Service Value)→FileName.TXT);
//Generating Log}

Description: Initially, a user makes its credential to login to the system. Now here this information is stored on third party and is being used for some further users' behavior based element generation. This element is taken as user attributes which in combine effect generates a key based on such values. The attribute elements are, username, password, timestamp, login failed attempts, uploaded files count and last session time of a user. Form this composite key is generated using MD5 algorithm which gives a digests value of 128 bits. The generation time of such digest is also measured for proving the effectiveness of the suggested approach. Later on this key can be updated by the user's selection of services and this value is being verified by some third party auditors. Now to make the user data secure and reduces the load of decryption this work is using fully homomorphic encryption using paillier[20] algorithms. Here the generated composite key of the combined values of attributes is passed for encryption algorithms to make the data secure.

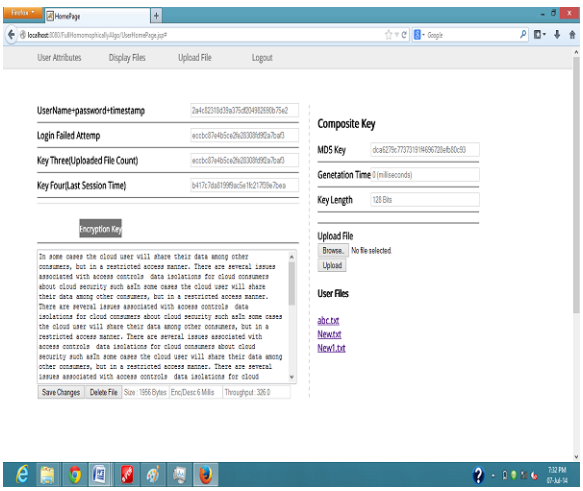

Fig 3:-Actual Implementation Window

The reverse process is applied for getting the data in the secure channel with reduced load on servers. Now the above process takes into consideration and monitored from some external entity such as a third party auditor which continuously watches the user's behavior. In this, the auditor and CSP trace the user's activity, its status, service details of users, load affected by taking the value from memory and CPU. This value is taken before activity and after activity to evaluate the changes made by the approach.

The approach is proving its efficiency of its components and their integration verified by some of the modules and policies. When the load of decryption is reduced form servers the effectiveness and performance will also be raised to a certain value in terms of their processing cycles and memory. The actual values of working model are given in the next section for its evaluation.

V. Result Evaluations

As compared to many of its existing approaches, which only provide binary results about the storage for the distributed servers, the proposed work further provides the localization of data error. 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. *Table and Graph Summary:* According to above values in the table the approach is proving its effectiveness than other existing security standards. It can be used in various application areas as required accordingly and can be taken as partial or complete features. In the above results

Table-1 to Table-3 gives the key formation using the keygen function and its characteristics for a generation like credential, key attributes, generation time, digest, etc. Table-4 gives the key generation and encryption time required with respect to different users. Table-5 give the values of different services or function based activation conditions which is measured before the initiation and after the completion and give the load value changes in memory and CPU cycles as a percentage.

Table-1: Key Type with Attribute Elements of Users

Serial No	Key Type	Key Length	Key Attributes
1	Hexadecimal	128 Bit	UserName+password+timestamp
2	Hexadecimal	128 Bit	Login Failed Attempts
3	Hexadecimal	128 Bit	Key Three(Uploaded File Count)
4	Hexadecimal	128 Bit	Key Four(Last Session Time)

Table-2: Active User Details And Service Usage

Serial No	User Name	Service 1-Generating Key	Service 2- Encryption
1	pawan123	1	0
2	Cloud computing	0	0
3	pp	2	1

Table-3: Generated key with Digest Based On Users Types

S. No	UID and Password	Key 1	Key 2	Key 3	Key 4
1	pawan123	ddb602de72508b7c6c9de d8fd1db9b2f	cfcd208495d565ef66e7dff 9f98764da	cfcd208495d565ef66e7dff 9f98764da	cfcd208495d565ef66 e7dff9f98764da
	pawan123				
2	Cloudcomputing	81d34eebe0e73947b858d 087aa0de7de	cfcd208495d565ef66e7dff 9f9hgkfd77	cfcd208495d565ef66e7dfa kdfkjhds77	cfcd208495d565ef66 e7dff9uiytwer89
	cloud123				
3	Pp	a17aef9d6c267455e1ab55 145d6f8cdf	cfcd208495d565ef66e7dff 9f98764da	eccbc87e4b5ce2fe28308fd 9f2a7baf3	9edd04c32a175d1c8 7f8dbfebba55327
	Pp				

Table-4: Digest Comparison Based On Users Types

S.No	Generated Composite Key (MD5 Digest)	Generation Time (ms)	Key Length (Bits)
User 1	a346d8687bc36674000062eb5fad338a	0	128
User 2	kd8687bc360e08ca125f5400062hjg77	1	128
User 3	f681cead0e08ca1c25f54a27d7619027	1	128

Table-5: Server Logs for Different Users and Services

Server Log Entry	File Delete		File Interaction		Key Generation		Login		Logout	
	Before	After	Before	After	Before	After	Before	After	Before	After
1	4	4	3	3	5	5	5	5	3	3
2	4	4	6	5	4	4	3	3	4	4
3	4	5	5	5	4	4	5	5	3	3
4	3	7	5	5	4	4	4	4	5	5
5	5	9	5	5	6	6	4	4	4	4
6	0	0	5	5	6	6	4	4	5	5
7	0	0	5	5	5	5	5	5	5	5
8	0	0	5	5	5	5	5	5	4	4
9	0	0	5	4	0	0	4	4	4	4
10	0	0	4	4	0	0	5	5	4	4
11	0	0	4	4	0	0	5	5	3	3
12	0	0	4	4	0	0	5	5	3	3

Table-6: Feature and Performance Based Comparison of TPA-HE from Existing Standards

Approach	Homomorphism Support	Operations	Monitoring and Auditing Service	Throughput=Size/ Time
DES [19]	Nil	Encrypt/ Decrypt	NA	3.30
AES [17]	Nil	Encrypt/ Decrypt	NA	2.42
RSA [19]	Nil	Encrypt/ Decrypt/Key Generation	NA	2.66
TPA-HE (Proposed)	Complete	Encrypt/Key Generation/Retrieval Without Decryption	Available	3.16

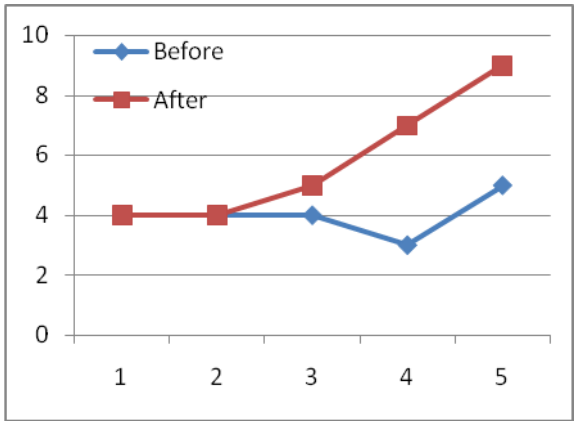

Fig 4: File Delete Graph Before and After Process

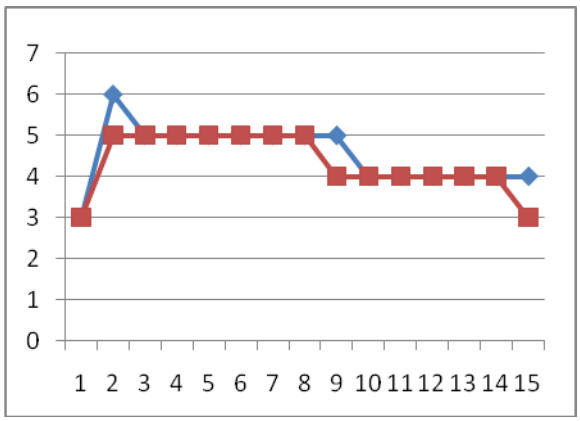

Fig 5: File Interaction Graph Before and After Process

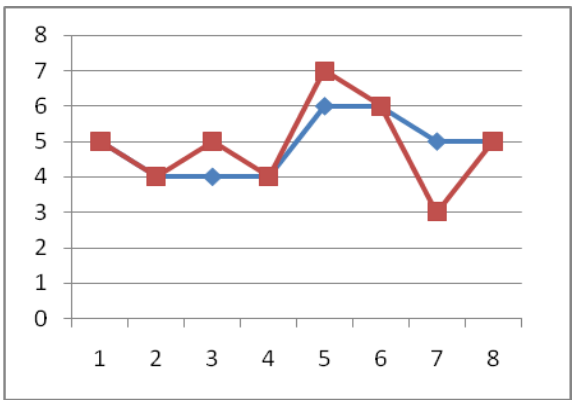

Fig 6: Key Generation Graph Before and After Process

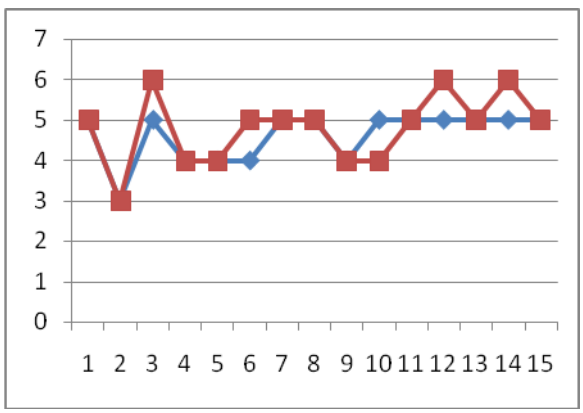

Fig 7: Login Graph Before and After Process

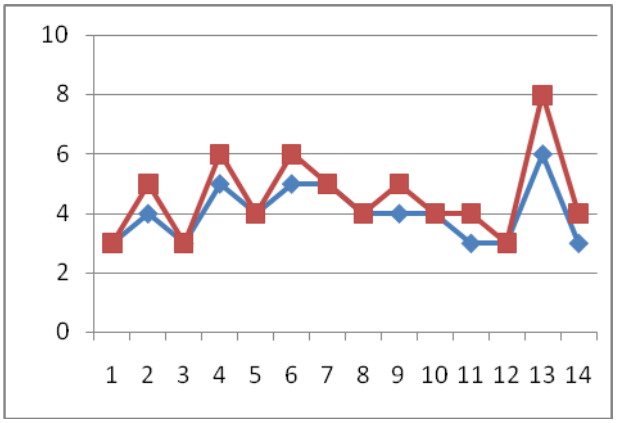

Fig 8: Logout Graph Before and After Process

Table-6 give the resultant outcome comparison of suggested TPA-HE with some existing mechanism by taking the function based parameter. Here the approach is verifying its results by the throughput values. The above changes measured in Table-5 based on the services or functions suggested in TPA-HE and their memory and CPU load. It is later on plotted and analyzed using graphs by which the approach is getting the results.

Thus the various parameters and results scenarios values are proven that the approach will serve as a great improvement over the existing security model. Its later implementations are more filtered and formally drafted so as to make the approach more applicable to different areas of fine grained access controls and homomorphic encryptions.

VI. Conclusion

The purpose of this proposed TPA-HE work is to identify and develop new security policies for improved data access in the cloud. It gives its implementation scenarios and modules behalf of which the suggested phenomenon can be applied. It will totally depend upon the trust of user & cloud that is handled by a third party auditor. The continuous checking mechanism is applied for regular monitoring of this data exchange. The work also uses homomorphic encryption for reduces overhead related to data modification and access by the authenticated user. The primary concern with the proposed approach is a better access mechanism with security policies for data and service delivery by cloud service providers. At the initial level of results evaluations on different parameters given by graphs and tables are proving better than existing ones. The later versions of research will going to be more formal and appropriate.

Future Work

Taken security as a major concern in this work has generated so many integration issues. While applying the above proposed architecture component must be placed in correcting order for better results. The security breaches identification can be done as a real time entity.

Homomorphic encryption & key handling issues can also be improved effectively by using KMIP protocol standard. Hence some problems and concepts that remain unaddressed can be performed in the future.

ACKNOWLEDGMENT

This research paper is made possible through the help and support from everyone; First and foremost, I would like to thank Mr. Mayank Kumar Sharma for his support and encouragement. He kindly read my paper and offered invaluable detailed advices on grammar, organization, and the theme of the paper.

He discussed regarding the cloud security policies & producing the approach adopted for this paper. I want to thank Mr. Shailendra Singh Bhalla, Mr. Ritesh K Shah to reproof the paper, Mr. Ramchandra Hablani, Ms. Megha Patidar for their support during this research.

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 by 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.

[18] Pawan Kumar Parmar and Megha patidar, "A Novel TPA-HE Security Service Architecture for Fine Grained Data Access in Cloud Computing", in International Conference on Cloud Big Data and Trust (ICCBDT), at RGPV Bhopal, India, Nov 2013.

[19] Sombir Singh, Sunil K Maakar and Dr. Sudesh Kumar, "A Performance Analysis of DES and RSA Cryptography ", in International Journal of Emerging Trends & Technology in Computer Science (IJETTCS), ISSN 2278-6856, Volume 2, Issue 3, May – June 2013.

[20] Pascal Paillier, "Public-Key Cryptosystems Based on Composite Degree Residuosity Classes", Published in J. Stern,Ed., Advances in Cryptology – EUROCRYPT'99,vol. 1592 of Lecture Notes in Computer Sc. Springer-Verlag, 1999.

AUTHORS

Pawan kumar parmar received BE degree in computer science and engineering from RGTU University in 2007.He is currently M.Tech student in the computer science and engineering department at sanghvi institute of management and science, Indore. His research interest includes cloud computing, database and data mining; he is a student member of computer society of India

Megha Patidar received her M. tech degree in information technology with a silver medal. She has many years of research experience in the field of distributed system, computer architecture, network and cloud computing. She has published many international and national level papers.

Mayank Kumar Sharma received his BE degree in computer science and M. Tech in information Technology (specialize in information Security-Gold medalist).He is currently pursuing Ph.D in Computer Engineering. He has published many international and national level papers. His research area includes Cloud computing, mobile adhoc network, Wireless sensor network, network security.