Performance Evaluation of Homomorphic Encryption based Data Access Control

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

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

ABSTRACT

Cloud computing is the new innovation in the field of internet technologies. It supports heterogeneous computing and user has not any managerial loads. 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 platform, software counted and infrastructure as pay per use basis so as user have not burdened of supporting cost. 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. Several new approaches created from last few years along with traditional security but all of them 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 cipher text 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 Behavioral 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.

Keywords

Cloud Computing, Security Service, Homomorphic Encryption with Behavioral Attribute (HEBA), Authentication, User Attributes, Monitoring Service**;**

1. 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. This dissertation proposes a novel HEBA policy based work to provide higher security with less managerial concern. The work is using homomorphic encryption for security approach. The work will also use a security breach event , 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.

1.1 Why Security Breaches Occurred

Security violations and risk areas are primarily vulnerable to some of its environmental factors which somewhere lack towards lack in protection levels. Some of the factors are [4]:

- Operating System Vulnerabilities
- Poorly Configured Firewalls
- Weak Access Structures & Password
- Loose Authentication
- Weak anti-malware controls
- Lack of Monitoring and Session Management
- Poor patch management
- Lack of automated de-provisioning

But, there are a number of areas of security that are often ignored by various IT organizations, and these can drastically amplify the risk of a booming security violation. These circumstances need unambiguous attention and rectification, and include [15]:

(i) **Insufficient User Authentication**—the typical username/password validation methods are grossly poor to defend against security breach, particularly with the creative social engineering attacks that have been used recently. Strong authentication is required, and different methods should be used based on a number of background factors. Hardware tokens have been basically used, but the latest breach of Secure ID tokens, as well as the cost and effort mapped with hardware tokens, make this an unappealing option.

(ii) **Insufficient Regular User Access Validation**—As a user goes through role change, promotions, etc., their access privileges are frequently not attuned to reproduce only their present task. Although many companies apply validation on user access rights frequently, it should be done on a formal and regular basis, and it should desirably be automated so that it can be done fast and easily. Lack of daily verification can direct to raised risk due to SOD (segregation of duties) violations.

(iii) **Lack of System Controls on Privileged Users**—a main reason of security breaches, validate users, normally have more rights than compare to rights which need for their job. By definition, these users need wide approach rights to do their job, but “wide” shouldn’t imply "unlimited." heedless or hateful actions can have destructing changes on your critical assets.

(iv) Lack of Information Use Control—controlling access to information is not enough protection. You must control use of information also in order to help ensure that it isn't disclosed or stolen. This defects has been a contributing reason to some of the most apparent security breaches of the current past.

(v) Lack of Continuous User Monitoring—apparently in the Wikileaks breach, nobody noticed that someone was copying thousands of sensitive documents from military systems for a small duration of time. Lots of security breaches can be identified with a inclusive approach to observe user action for suspicious activities. As a result, the need of efficient and continuous observation of user action is also one of the most useful benefactors to a high risk of security breach.

1.2 Understanding Homomorphic Encryption (HE)

Encryption is used to secure data from tappers who would otherwise block private communication. Message encryption done by one side and this ciphertext sends to others side which receive this message and decrypt this ciphertext to make obtain message from ciphertext. To stop an un-trusted third party from eavesdropping, the problem of recovering any information about the message from the ciphertext should be sensible hard; in addition, the cipher text should itself no disclose any information for the message. Increasingly, Computation and storage of data is outsourced to these un-trusted parties, which increase to the need of a scheme of encryption in which we can compute ciphertexts. Homomorphic encryption approaches are flexible by design. We can use this in voting system to create secure voting system by using the homomorphic approach of various cryptosystems, private information retrieval and collision-resistant hash functions, and enables universal use of cloud computing by make sure the privacy of processed data. There are lots of partially, effective homomorphic cryptosystems, and lots of fully homomorphic, but less efficient cryptosystems. Although a cryptosystem which is casually homomorphic can be subject to attacks on this basis, if activated properly homomorphism can also be used to perform computations securely.

Informally , a homomorphic cryptosystem is the system by which we have use one more property by it 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 approach of different encryption schemes have been attraction of the cryptographic community for decades. With the increase the uses of cloud computing and decentralized processing, the requirement for security in such applications is increasing. Only recently, however, has the construction of a fully homomorphic encryption scheme been achieved.

2. MOTIVATION

Traditional security mechanism can be used to achieve the goals but somewhere it is not service the complete users and creators need. Detect the most best security technique is always questioned because of their black box detection and measurement phenomenon's. 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, need of security warranties in the SLAs between the cloud consumer and provider; distribution of assets with competitors or malevolent users.

Thus there must be some modification performed which increases the level of security for the users information with reduction in performance drops. These areas of security are getting complex day by day as the number of users gets exponentially increased. These large users are highly concerned with the data losses, unauthorized leakages, robust authentication, malicious users handling, services failures & hijacking of sessions while accessing their information. After studying the various issues related to information security and approaches used to solve them the work identifies some of the problems which remain unaddressed. Few of them are mentioned here as the basis of the proposed work.

3. OBJECTIVES

The main contributions of the proposed work are as follows:

1. A uniformly defined protocol is required for outsourcing data policies for normal user to cloud platform.
2. It provides user to computational operations allow continues on encrypted data.
3. Achieving the security of encryption and reading the data without decryption using Palliers homomorphic encryption.
4. Third party monitoring services for users behavior and response analysis.
5. The providers need not to be aware about the type of information or data stored on it. It can only be done by some of the authorized user with right permission.
6. An improvement is made over a traditional homomorphic encryption scheme, such that responsibility over key generation is given to the data owner and a trusted authority advises the action; the owner have not any burden for highest computational .
7. Add-on security mechanism is provided with computational working continues to be encrypted data.
8. Re-encryption is used as a method of changing the stored ciphertexts, allows effective recall of users; consequent key regeneration and attributes removal are not require for it, and may be administer by a trusted authority without involvement of the data vendor.

4. RELATED STUDY

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. For solving security problem in cloud computing on any single scheme is sufficient. 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 [18]. But still there are some issues and things which have to be opened and presented here as literature survey.

In the articles [16], 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 detects security, privacy, and other organizational demands for cloud services to meet, as a standard for a cloud provider selection. To supplement the server side of the equation, cloud-based applications need a client side to start and acquire services. While Web browsers usually serve as clients, other chances available. In addition, an passable and secure network communications infrastructure should be available. Lots of the simple interfaces and service abbreviation at client side, server, and network belie the inborn underlying complication that affects security and privacy. Therefore, So it is necessary things that to grasp the cloud provider uses approaches means technologies to provision services and the implications the technical controls involved have on security and privacy of the system throughout its lifecycle. 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.

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 [10] 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. Implementing bootstrapping could also lead to a number of nice applications of homomorphic encryption. The paper presents how to mitigate the problem of the large ciphertext size for the Ring-LWE based FHE solution. In all of the above applications, Client use FHE scheme for data encryption to upload for this client communicate with provider, and the cloud operates on these data and returns encrypted outputs to the client. The Scheme keeps data secret, but also allow 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 [3].

In the article [13] 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 gives the elasticity to permit customers to make a large range of applications with high privacy and integrity. Specifically, AWS physical and functional security thing are described for network and server structure under AWS's management, as well as service-specific security implementations.

4.1 General Issues in HE

As homomorphic encryption is used for reducing the decryption complexity of online mechanism and thus its real time behavior is required. When a file is gets encrypted by FHE then its update will handled directly by the encryption algorithms. Considering the above behaviors some of issues which need to be overcome by the HE techniques are given here as:

- Real time complete encryption for data transfers and for retrieval decryption is not required
- Handling of frequent updates directly on file
- Key managements with exchanges through some trusted mechanism
- On demand access and control operations
- Reduced access and response time
- High privacy and confidentiality
- Scalable and elastic environment
- Trusted channels for encryption

5. WORK EMBODIED

In today's world the information is having the highly solicited by its value which is considered to be most important asset for the user. It holds the user's personal, social, health and other kind of relevant data which needs to be offered with high security. If somewhere due to any circumstantial condition the data is loosed causes heavy failure to the authenticity policies or commercial profits of the user. Computation involves massive processing along with heterogeneous data support which is quite expensive for the user to purchase permanently, thus along with several other web services now the security is also getting this tradition followed. Some of the basic security concerns of the system are authenticity, fine grained data access, confidentiality, privacy, integrity etc. To ensure the control of client over such system more robust security mechanism is now available with the recent technologies which make data unreadable. The problem associated with the traditional security systems is that they are having lesser control and known policies which gives a vacant space to attacker for planning his malicious activity. This work aims towards improving the traditional security approaches through some more user oriented controls and deeper protection levels with optimized performance.

6. PROBLEM DEFINITIO N

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 guaranteed with available 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.

7. PROPOSED WORK

This work proposes a novel HEBA (Homomorphic Encryption with Behavioral Attribute) based security scheme for cloud computing. It gives optimal computational load for dynamic data encryption using some of the user's behavior 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 behavioral 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

7.1 Algorithmic Steps

(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 behavior 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

7.2 Description

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 behavior 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 blinders a partly homomorphic cryptosystem, let we consider the public key is the modulus *m* and the base *g*, then the encryption of a message *x* is ε (x) = $g^x r^m$mod $m^{2,}$ for some random. The homomorphic property is then

We can evaluate encrypted function which guarantees its privacy by Using homomorphic cryptosystems . Homomorphic approach also work for encrypted data to calculate publicly while supporting the privacy of the secret data. This can be done encrypting the data in further and then attempting the homomorphic property to calculate 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 behavior for fault localization. It uses adversary monitoring through a weak or strong bond. Weak opponent is involved in degrading the user’s data files stored on singular servers. Once a server is composed, an opponent can degrade the original data files by including its our bogus data to stop the real data from being fetched by the user. While strong opponent is the best case scenario, in which we presume that the opponent can settlement all the storage servers so that he can on purpose change the data files as long as they are internally consistent. In fact, this is quall to the case in which all servers are banding jointly to cover a data loss or degrade 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.

7.3 Architecture

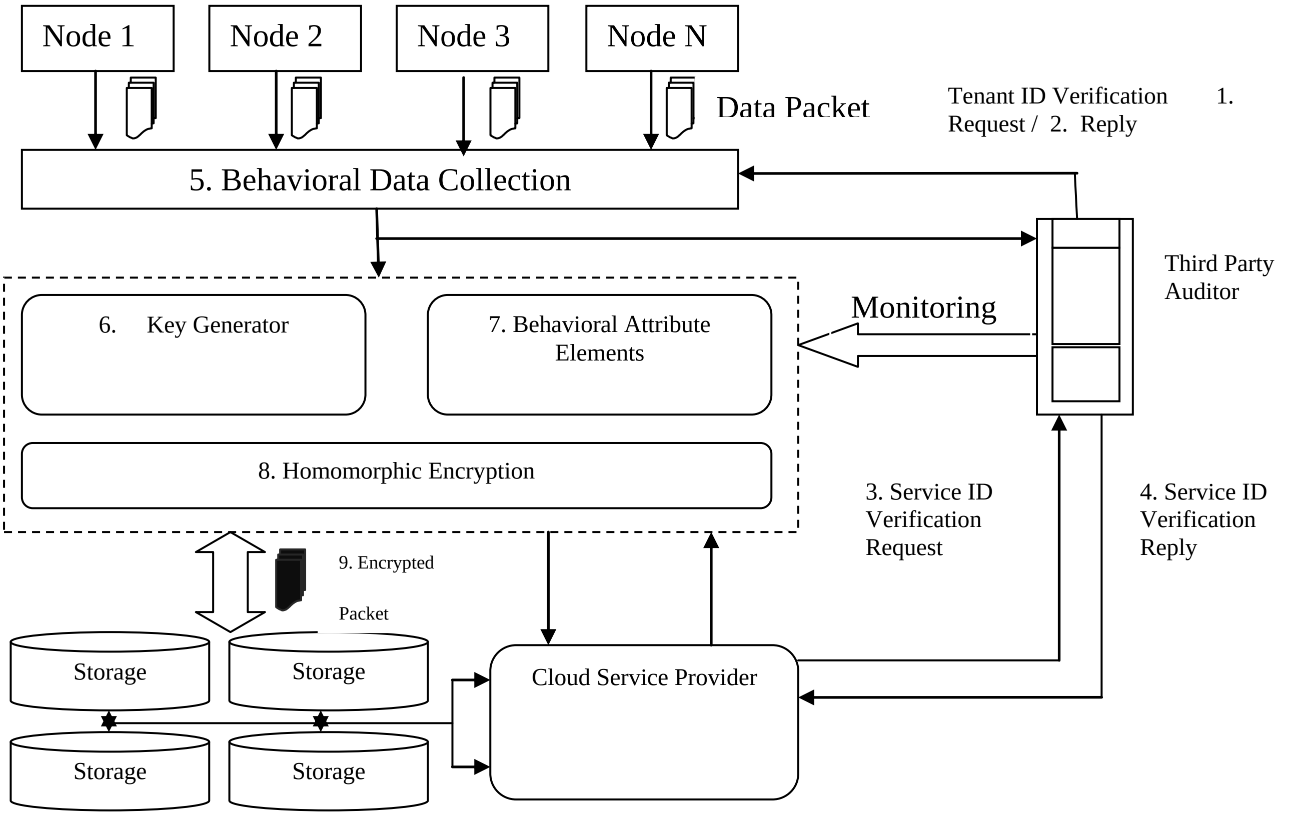
Node 1
Node 2
Node 3
Node N
Data Packet
Tenant ID Verification 1. Request / 2. Reply
5. Behavioral Data Collection
Third Party Auditor
Monitoring
6. Key Generator
7. Behavioral Attribute Elements
8. Homomorphic Encryption
3. Service ID Verification Request
4. Service ID Verification Reply
9. Encrypted Packet
Storage
Storage
Storage
Storage
Cloud Service Provider

Fig 1: Implemented HEBA Cloud Security Service Architecture

8. RESULT ANALYSIS

8.1 Constrained Analysis of Tool

Table 1: Services Configured Using Cloud Implementation of HEBA

S. No	User Name	Cloud Provider	RAM	Processor	Cloud OS	Services Configured
1	Amit	Ants (50 INR/10min)	4 GB	i3 Dual Core	Cent OS	Calculator /Tic Tac Toe (game)
2	John	Eucalyptus (100 INR/10min)	8 GB	i3 Quad Core	Red Hat	Calculator /Tic Tac Toe (game)
3	Tom	Hadoop (200 INR/10min)	16 GB	i5 Dual Core	Fedora	Calculator /Tic Tac Toe (game)
4	Marry	Nimbus (200 INR/10min)	32 GB	i5 Quad Core	Windows	Calculator /Tic Tac Toe (game)

Table 2: Security as a Service Analysis (Encryption)

S. No	User Name	File Type	Size (Kb)	Key Generation Time	Encryption Time	Decryption Time
1	User A	txt	159	398(Microseconds)	5 Millis	5 Millis
2		docx	40	256(Microseconds)	4 Millis	4 Millis
3		rtf	276	227 (Microseconds)	4 Millis	4 Millis
4	User B	txt	98	146(Microseconds)	5 Millis	5 Millis
5		docx	35	210 (Microseconds)	4 Millis	4 Millis
6		rtf	276	282 (Microseconds)	3 Millis	3 Millis

Table 3: Resource Optimization Analysis Using JConsole for HEBA

S. No	No. of CPU	Physical Memory(Kb)	Total Threads (Count)		Heap Size (Kb)		CPU Usage (%)	
			Before	After	Before	After	Before	After
1	Single	3073296	22	32	15,114	19,283	0.2	0.6
2	Dual	6130215	26	37	30245	36566	0.3	0.8
3	Quadruple	94168881	31	40	43678	58849	0.5	1.2

8.2 Parametric Analysis of Services

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. Thus to measure the results some of the identified parameters of work by which comparison can be made with existing work.

Table 4.6: Key Type with Attribute Elements of Users

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

Table 4.7: 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 4.8: Feature and Performance Based Comparison of HEBA from Existing Standards

Approach	Homomorphism Support	Operations	Monitoring and Auditing Service	Throughput=Size/ Time
DES	Nil	Encrypt/ Decrypt	NA	3.08
AES	Nil	Encrypt/ Decrypt	NA	2.42
RSA	Nil	Encrypt/ Decrypt/Key Generation	NA	2.28
HEBA (Proposed)	Complete	Encrypt/Key Generation/Retrieval Without Decryption	Available	3.16

8.3 Table Summary

According to above values in table the approach is proving its effectiveness than other existing security standards. It can be used for various application areas as required accordingly and can be taken as partial or complete features. The above results gives the key formation using the keygen function and its characteristics for generation like credential, key attributes, generation time, digest etc. Next table gives the key generation and encryption time required with respect to different users. It also give the values of different services or function based activation conditions which are measured before the initiation and after the completion and give the load value changes in memory and CPU cycles as a percentage. The result also gives the comparison of suggested HEBA 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 are based on the services or functions suggested in HEBA and their memory and CPU load. It is later on plotted and analyzed using graphs by which the approach is getting the results.

8.4 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.

9. 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 previous works for make sure integrity of remote data, the new approach supports efficient and secure dynamic operations on data blocks, like delete, append and update. Extensive performance and security analysis shows that the new implemented scheme is highly resilient and efficient against malicious data modification attack, and even server attacks. It manages each interaction scenario of user, cloud and storage through a token system. The work also uses a user attribute based encryption method for security of data. This attribute will work as parameters of key generation and will improve the data isolation issues. The work increase the trusted data sharing technology into the cloud computing environment to ensure security, computing requirements with security and efficiency then fulfill consumer requirements and gains the higher trust values.

10. FUTURE WORKS

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. Behavior based 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. The implementation of the above proposed mechanism is configured in Aneka 3.0 cloud platform tool in near future.

11. REFERENCES

[1] Dr. Sanjay Thakur , Amit Kanungo and Prateek Nahar , "A Novel Homomorphic Encryption with Behavioural Attribute Based Fine Grained Data Access in Cloud Computing", in International Journal of Advanced Research in Computer Science and Software Engineering, Volume 4, Issue 12, December 2014

[2] Farhan Bashir Shaikh and Sajjad Haider, "Security Threats in Cloud Computing", in IEEE 6th International Conference on Internet Technologies & Transactional databases, UAE, 2011.

[3] Farzad Sabahi, "Cloud Computing Security Threats and Responses", in IEEE Transaction, 2011.

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

[5] Mohemed Almorsy, John Grundy and Amani S. Ibrahim, "Collaboration-Based Cloud Computing Security Management Framework", in IEEE 4th International Conference on Cloud Computing, 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, Vol. 8, Issue 3, No. 2, May 2011.

[7] 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

[8] 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, 2010.

[9] Stephen S. Yau and Ho G. An, "Confidentiality Protection in Cloud Computing Systems", in International Journal of Software and Informatics, Vol.4, No.4, December 2010.

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