IDEA algorithm Secure Shared Data on Cloud

Ms.Priti Padole
M. Tech student, dept. of CSE
G.H Raisoni Institute of Eng. & Technology
For woman Nagpur, India
prititisha@gmail.com

Ms.Hemlata Dakhore
Ass. Prof, dept. of CSE
G.H Raisoni Institute of Eng. & Technology
For woman Nagpur, India
hemlatadakhore@raisoni.net

Abstract -- Cloud computing is most important part of fasted internet based computing world in which enables sharing of service as well as data in less time. This technology allows users to access services without installation any application to use their personal data and application at anywhere are placed computer with internet or intranet access. When many users store their data in the cloud, then security is a prime concern. Adoption of this technology is increasing fast. This system enabled to work that capable to change the working environment. It is interconnecting between resources of computer for integrated effectively. In this paper introduce a novel privacy preserving authenticated access control scheme for securing data in clouds. When user want to store their data in cloud first verifies the authentication of users. To securely access data from cloud access control feature is added in which only valid users are able to decrypt the stored information. This system prevents from attacks and creation, modification, and reading data stored all these function supported by cloud. The communication, computation, and storage overhead are comparable to centralized approaches. Here Work is focuses on IDEA Encryption Algorithm for stored data in cloud. This method is used to encrypt data and access securely.

Keywords -- *Cloud computing, Encryption, Data integrity, IDEA Algorithm, privacy-preserving, Batch auditing, Data Dynamics.*

*****

I. INTRODUCTION

Data storing to share with users with the use of cloud has become a fashion. The numbers of clients store their important data in remote servers in the cloud, without leaving a copy in their local computers. Many times the user important data stored in the cloud because of the clients must ensure it is not lost or corrupted. While it is easy to check data integrity after completely downloading the data to be checked, downloading large amounts of data just for checking data integrity is a waste of communication bandwidth. Hence, a lot of works have been done on designing remote data integrity checking protocols, which allow data integrity to be checked without completely downloading the data. Remote data integrity checking is first introduced in which independently propose methods for solving this problem. After that propose a remote storage Type Style and Fonts auditing method based on pre-computed challenge-response pairs.

To enhancement of security level of shared data in cloud we use dynamic hashing technique and ammonization technique in encryption algorithm to protect our shared data from attackers

Recently many works focus on providing three advanced features for remote data integrity checking protocols: data dynamic, public verifiability and privacy against verifiers. The system in support data dynamics at the block level, including block insertion, block modification and block deletion. It supports data append operation. In addition, can be easily adapted to support data dynamics. Can be adapted to support data dynamics by using the techniques. On the other hand, its support verifiability, by which anyone (not just the client) can perform the integrity checking operation.

II. OBJECTIVE

Our contribution in this paper is summarized as follows:

1) Secure shared data in cloud without using central authority

The main objective of this project is constructing a secure data storage system that supports multiple functions is challenging when the storage system is distributed and has no central authority.

2) Maintain integrity of data

A data verifier is able to verify the integrity of shared data without retrieving the entire data from cloud.

3) Maintain identity privacy

A verifier cannot distinguish the identity of the signer on each block in shared data during the process of auditing.

III. RELATED WORK

[1] Boyang Wang, Baochun Li, and Hui Li, in the paper "ORUTA: PRIVACY-PRESERVING PUBLIC AUDITING FOR SHARED DATA IN THE CLOUD" [1] Using HARS and its properties established Oruta, our privacy-preserving public auditing mechanism for shared data in the cloud. With Oruta, the TPA can verify the integrity of shared data for a group of users without retrieving the entire data. Meanwhile, the identity of the signer on each block in shared data is kept private from the TPA during the auditing.

[2] C. Wang, Q. Wang, K. Ren, and W. Lou, in the paper "PRIVACY-PRESERVING PUBLIC AUDITING FOR DATA STORAGESECURITY IN CLOUD COMPUTING" [6] Cloud computing is the long dreamed vision of computing as a utility, where users can remotely store their data into the

cloud so as to enjoy the on-demand high quality applications and services from a shared pool of configurable computing resources. By data outsourcing, users can be relieved from the burden of local data storage and maintenance. Thus, enabling public Auditability for cloud data storage security is of critical importance so that users can resort to an external audit party to check the integrity of outsourced data when needed.

[3] C. Erway, A. Kupcu, C. Papamanthou, and R. Tamassia, in this "DYNAMIC PROVABLE DATA POSSESSION"[5] As storage-outsourcing services and resource-sharing networks have become popular, the problem of efficiently proving the integrity of data stored at untrusted servers has received increased attention. In the provable data possession (PDP) model, the client preprocesses the data and then sends it to an untrusted server for storage, while keeping a small amount of meta-data. The client later asks the server to prove that the stored data has not been tampered with or deleted (without downloading the actual data). However, the original PDP scheme applies only to static (or append-only) files. Present a definitional framework and efficient constructions for dynamic provable data possession (DPDP), which extends the PDP model to support provable updates to stored data. Use a new version of authenticated dictionaries based on rank information.

[4] G. Ateniese, R.D. Pietro, L.V. Mancini, and G. Tsudik, **"SCALABLE AND EFFICIENT REMOTE DATA POSSESSION CHECKING IN RITICALINFORMATION INFRASTRUCTURES"[7]**Checking data possession in networked information systems such as those related to critical infrastructures (power facilities, airports, data vaults, defense systems, and so forth) is a matter of crucial importance. Remote data possession checking protocols permit checking that a remote server can access an uncorrupted file in such a way that the verifier does not need to know beforehand the entire file that is being verified.

IV. EXISTING SYSTEM

Cloud improves due to centralization of data, increased security-focused resources, etc., but concerns can persist about loss of control over certain sensitive data, and the lack of security for stored kernels. Security is often as good as or better than other traditional systems, in part because providers are able to devote resources to solving security issues that many customers cannot afford. To securely introduce an effective third party auditor (TPA), the following two fundamental requirements have to be met: 1) TPA should be able to efficiently audit the cloud data storage without demanding the local copy of data, and introduce no additional on-line burden to the cloud user; 2) The third party auditing process should bring in no new vulnerabilities towards user data privacy.

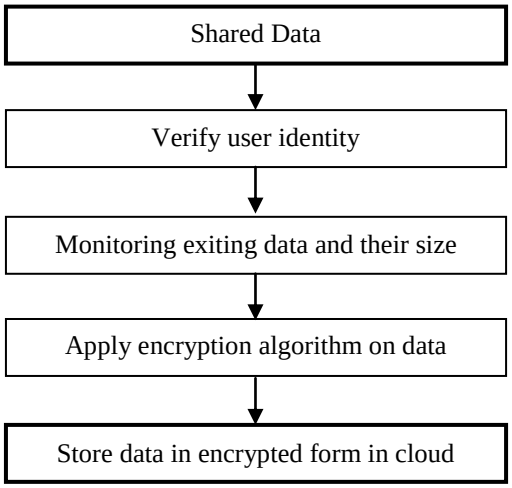
A. Drawbacks of existing system

Cloud Storage system provides the user for safe and consistent place to save valuable data and documents. However, user's files are not encrypted on some open source cloud storage systems. I.e. TPA demands retrieval of user data, here privacy is not preserved.

The storage service provider can easily access the user's files. This brings a big concern about user's privacy. The user has no supreme control over the software applications including secret data. User has to depend on the provider action, maintenance and admin it.

V. PROPOSED SYSTEM

Our scheme for policy-based content sharing in the cloud involves four main entities: the data owner (Owner), the users (Users), the identity providers (IdPs), and the cloud storage service (Cloud). The interactions are numbered in the figure. Our approach is based on three main phases: identity token issuance, identity token registration, and document management.



The work is distributed in following steps

Step 1: Verify User identity

To share data on cloud first verify user that trusted or not by using their id tokens.

Step 2: Monitoring existing data and their size

After user verification monitor their data i.e. new data or any modificated old data that want to replace it. And check their size to store data.

Step 3: Encryption Algorithm

Uses encryption algorithm to secure shared data in cloud by applying secret key on data. In this algorithm we used data hiding, dynamic hashing and ammonization technique to enhancement security of shared data in cloud.

Step 4: Store data on cloud

After applying encryption on data that encrypted data store on cloud

A. International Data Encryption Algorithm (IDEA)

Idea was to develop a strong encryption algorithm, which would replace the des procedure developed in the USA. In the seventies. It is also interesting in that it entirely avoids the use of any lookup tables or s-boxes. When the famous pgp email and file encryption product was designed by Phil Zimmermann, the developers were looking for maximum

security. Idea was their first choice for data encryption based on its proven design and its great reputation.

The block cipher IDEA operates with 64-bit plaintext and cipher text blocks and is controlled by a 128-bit key. The fundamental innovation in the design of this algorithm is the use of operations from three different algebraic groups. The substitution boxes and the associated table lookups used in the block ciphers available to-date have been completely avoided. The algorithm structure has been chosen such that, with the exception that different key sub-blocks are used, the encryption process is identical to the decryption process.

B. Bit serial architecture

Bit serial architectures are characterized by the property that operators perform their computations in a bitwise fashion and communications between operators are multiplexed in time over a single wire. Dataflow begins with either the least significant bit or the most significant bit, but the former is more commonly used due to its compatibility with two's complement arithmetic.

C. Batch auditing

Organizations tend to relax their security posture and so there is always a need to perform regular security check for in audit stage, auditor can effectively verify the proof of data possession by the service provider through a challenge-response protocol. During extraction phase, the auditor verifies data integrity of data returned to the customer through the auditor. The encrypted data and a "blinded" version of encryption key are forwarded to the auditor. The auditor checks its completeness and passes it to the customer who then recovers the actual data. Proposed protocols in this we provide completeness & soundness of data with zero knowledge of data contents to auditor. The protocols divide the data in two parts, an encryption key and encrypted data. The encrypted data rely on a cryptographic hash function and symmetric key encryption. The protocols rely on external authentication methods for communication the cloud service providers must adhere to security measures policies to mitigate security risks. A strict check on policy compliance must be implemented through it security auditing.

D. Data Dynamics

To enable each user in the group to easily modify data in the cloud. A dynamic operation includes an insert, delete or update operation on a single block. However, since the computation of a ring signature includes an identifier of a block, traditional methods, which only use the index of a block as its identifier (i.e., the index of block miss j), are not suitable for supporting dynamic operations on shared data efficiently. The reason is that, when a user modifies a single block in shared data by performing an insert or delete operation, the indices of blocks that after the modified block are all changed and the changes of these indices require users, who are sharing the data, to re-compute the signatures of these blocks, even though the content of these blocks are not modified.

VI. CONCLUSION

In this paper, we propose to enhancement of security level of shared data in cloud we use dynamic hashing technique and ammonization technique in encryption algorithm to protect our shared data from attackers and maintain their integrity and privacy.

REFERENCES

[1] Boyang Wang, Student Member, IEEE, Baochun Li, Senior Member, IEEE, and Hui Li, Member, IEEE" Oruta: Privacy-Preserving Public Auditing for Shared Data in the Cloud" IEEE TRANSACTIONS ON CLOUD COMPUTING, VOL. 2, NO. 1, JANUARY-MARCH 2014.

[2] M. Armbrust, A. Fox, R. Griffith, A.D. Joseph, R.H. Katz, A.Konwinski, G. Lee, D.A. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, "A View of Cloud Computing," Comm. ACM,vol. 53, no. 4, pp. 50-58, Apr. 2010.

[3] D. Song, E. Shi, I. Fischer, and U. Shankar, "Cloud Data Protectionfor the Masses," Computer, vol. 45, no. 1, pp. 39-45, 2012

[4] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner,Z. Peterson, and D. Song, "Provable Data Possession atUntrusted Stores," Proc. 14th ACM Conf. Computer and Comm.Security (CCS '07), pp. 598-610, 2007.

[5] C. Wang, Q. Wang, K. Ren, and W. Lou, "Ensuring Data StorageSecurity in Cloud Computing," Proc. 17th Int'l Workshop Quality ofService (IWQoS'09), pp. 1-9, 2009.

[6] B. Wang, B. Li, and H. Li, "Certificateless Public Auditing forData Integrity in the Cloud," Proc. IEEE Conf. Comm. and NetworkSecurity (CNS'13), pp. 276-284, 2013

[7] C. Erway, A. Kupcu, C. Papamanthou, and R. Tamassia,"Dynamic Provable Data Possession," Proc. 16th ACM Conf. Computerand Comm. Security (CCS'09), pp. 213-22