International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 06 Issue: 03 | Mar 2019 www.irjet.net p-ISSN: 2395-0072

REDSC: RELIABLITY OF DATA SHARING IN CLOUD

ADITYA SHITOLE[1], OMKAR DHAPTE[2], EMAAD PERZADE[3] ,SUSMITA GAIKWAD[4]

ASMITA PATIL[5]

[1] *Student, Dept of Computer Engineering, A.I.S.S.M.S. Polytechnic Pune, Maharashtra, India*
[2] *Student, Dept of Computer Engineering, A.I.S.S.M.S. Polytechnic Pune, Maharashtra, India*
[3] *Student, Dept of Computer Engineering, A.I.S.S.M.S. Polytechnic Pune, Maharashtra, India*
[4] *Student, Dept of Computer Engineering, A.I.S.S.M.S. Polytechnic Pune, Maharashtra, India*
[5]*Professor, Dept of Computer Engineering, A.I.S.S.M.S. Polytechnic Pune, Maharashtra, India*

Abstract: Cloud storage auditing schemes for shared data refer to checking the integrity of cloud data shared by a group of users. Remote data integrity checking enables a data storage server, says a cloud server, to prove to a verifier that it is actually storing a data owner's data honestly. To date, a number of Remote data integrity checking protocols have been proposed in the literature, but most of the constructions suffer from the issue of a complex key management, that is, they rely on the expensive public key infrastructure which might hinder the deployment of Remote data integrity checking in practice. In this paper, we propose a new construction of identity-based (ID-based) Remote data integrity checking protocol by making use of key-homomorphic cryptographic primitive to reduce the system complexity and the cost for establishing and managing the public key authentication framework in public key infrastructure based Remote data integrity checking schemes. We formalize ID-based Remote data integrity checking and its security model including security against a malicious cloud server and zero knowledge privacy against a third party verifier. The proposed ID-based Remote data integrity checking protocol leaks no information of the stored data to the verifier during the Remote data integrity checking process. The new construction is proven secure against the malicious server in the generic group model and achieves zero knowledge privacy against a verifier. Extensive security analysis and implementation results demonstrate that the proposed protocol is provably secure and practical in the real-world applications. We Extend this work with Group Management with Forward Secrecy & Backward Secrecy by Time Duration & Recovery of File when Data Integrity Checking Fault Occur

Key Words: cloud data, security, group data security.

1.INTRODUCTION:

In cloud storage auditing schemes, the data owner needs to use his/her private key to generate authenticators (signatures) for file blocks. These authenticators are used to prove that the cloud truly possesses these file blocks. When a user is revoked, the user's private key should also be revoked. For traditional cloud storage auditing schemes for share data, all of authenticators generated by the revoked user should be transformed into the authenticators of one designated non-revoked group user. Cloud computing, which has received considerable attention from research communities in academia as well as industry, is a distributed computation model over a large pool of shared-virtualized computing resources, such as storage, processing power, applications and services. Cloud users are provisioned and release recourses as they want in cloud computing environment. This kind of new computation model represents a new vision of providing computing services as public utilities like water and electricity. Cloud computing brings a number of benefits for cloud users. This non-revoked group user needs to download all of revoked user's blocks, re-sign these blocks, and upload new
to the cloud. Obviously, it costs huge amount of computation resource and communication resource due to the large size of shared data in the cloud. In order to solve this problem, recently, some auditing schemes for shared data with user revocation have been proposed.

2. EXISTING SYSTEM:

At present all the documents are stored using some credentials that users have to remember. So they are like any unique number or any password but these all hackable. As existing system is not much user friendly that every existing human being individual can use it. If any user needs any documents or any identity proof the user has to carry some copy if the same so sometimes it is not possible.

3. PROPOSED SYSTEM:

It is highly secured of the data as it uses a cloud storage with AES Alogrithm. Cloud is for the data owner to encrypt his data before storing into the Cloud, and hence the data remain information-theoretically secure against the Cloud provider and other malicious users. When the data owner wants to share his data to a group, he sends the key used for data encryption to each member of the group. Any member of the group can then get the encrypted data from the Cloud and decrypt the data using the key and hence does not require the intervention of the data owner.

4. APPLICATIONS:

Bank application

Social Web Application.

5. SYSTEM REQUIREMENTS:

5.1. HARDWARE REQUIREMENTS:

System Processors	:	Core2 Duo
Speed	:	2.4 GHz
Hard Disk	:	150 GB

5.2. SOFTWARE REQUIREMENTS:

Operating system and on words	:	64bit Windows 7
Coding Language	:	Java J2EE
IDE	:	Eclipse kepler
Database	:	MYSQL

6. Modules:

6.1. DATA GROUP SHARING:

Server can utilize this total trapdoor and some public information to perform keyword search and give back the outcome to Bob. In this way, in KASE, the assignment of keyword search right can be accomplished by sharing the single total key. We take note of that the assignment of decryption rights can be accomplished utilizing the key-total encryption approach as of late proposed in , however it remains an open issue to appoint the keyword search rights together with the decryption rights, which is the subject point of this paper. To outline, the issue of developing a KASE .

6.2. PUBLIC INTEGRITY AUDITING:

public integrity auditing for shared dynamic data to gathering client denial. Our contributions are three folds:1) We investigate on the protected and proficient shared data coordinate examining for multi-client operation for ciphertext database.2) By consolidating the primitives of victor responsibility, hilter kilter gathering key assention and gathering mark, we propose a proficient data examining plan while in the meantime giving some new elements, for example, traceability and countability. 3) We give the security and productivity examination of our plan, and the investigation results demonstrate that our plan is secure and effective.

6.3. CLOUD STORAGE MODEL:

Cloud storage is a model of data stockpiling where the computerized datais put away in consistent pools, the physical stockpiling compasses numerous servers (and regularly areas), and the physical environment is ordinarily possessed and oversaw by a facilitating organization. These cloud storage suppliers are in charge of keeping the data accessible and available, and the physical environment secured and running. Individuals and associations purchase or rent stockpiling limit from the suppliers to store client, association, or application data.Cloud stockpiling services may be gotten to through a co-found cloud PC benefit, a web application programming interface (API) or by applications that use the API, for example, cloud desktop stockpiling, a cloud storage gateway or Web-based substance administration frameworks. why should approved get to and alter the data by the data owner. The cloud storage server is semi-trusted, who gives data stockpiling services to the gathering clients. TPA could be any substance in the cloud, which will have the capacity to direct the data honesty of the mutual information put away in the cloud server. In our framework, the data owner could encrypt and transfer its data to the remote cloud storage server. Likewise, he/she shares the benefit, for example, get to and change (accumulate and execute if fundamental) to various group clients.

6.4. REVOKED GROUP USER:

The group signature will keep the conspiracy of cloud and denied bunch clients, where the data owner will partake in the client repudiation stage and the cloud couldn't renounce the data that last altered by the disavowed user.An assailant outside the gathering (incorporate the repudiated bunch client distributed storage server) may get some learning of the plaintext of the data. Really, this sort of aggressor needs to at lease break the security of the received gathering data encryption plan. The cloud storage server conspires with the disavowed bunch clients, and they need to give an illicit data

without being distinguished. Really, in cloud environment, we expect that the cloud storage server is semi-trusted. In this way, it is sensible that a disavowed client will conspire with the cloud server and share its secret group key to the cloud storage server. For this situation, in spite of the fact that the server intermediary bunch client repudiation way [24] brings much correspondence and calculation expense sparing, it will make the plan unstable against a pernicious cloud storage server who can get the secret key of renounced clients amid the client disavowal stage. Accordingly, a malignant cloud server will have the capacity to make data m, last altered by a client that should have been be disavowed, into a malevolent data m′. In the client renouncement handle, the cloud could make the malicious data m′ get to be legitimate.

6.5. GROUP SIGNAGTURE:

Group signature is presented by Chaum and Heyst It gives namelessness to signers, where every gathering part has a private key that empowers the client to sign messages. Be that as it may, the subsequent sign keeps the character of the signer secret. More often than not, there is an outsider that can lead the sign namelessness utilizing a unique trapdoor. A few frameworks bolster denial where bunch enrollment can be handicapped without influencing the signing capacity of unrevoked clients. Boneh and Shacham proposed a productive gathering signature with verifier-neighborhood denial. The plan gives the properties of gathering sign, for example, caring namelessness and traceability. Likewise, the plan is a short sign plan where client disavowal just requires sending repudiation information to signature verifiers. Libert et al. proposed another versatile denial technique for gathering sign taking into account the show encryption system. On the other hand, the plan presents vital capacity overhead at gathering client side. Later, Libert et al. outlined a plan to update the previous plan which could acquire private key of consistent size. In their plan, the unrevoked individuals still don't have to overhaul their keys at every repudiation.

ALGORITHMS:

VC.KeyGen(, q).
Given the security parameter k and the size q of the committed vector (with q = poly(k)), the key generation outputs some public parameters pp.

VC. (m1, ...,mq).
On input a sequence of q messages m1, ...,mq∈ M (M is the message space) and the public parameters pp, the committing algorithm outputs a commitment string C and an auxiliary information aux.

VC. (m, i, aux).
This algorithm is run by the committer to produce a proof i that m is the i-th
committed message. In particular, notice that in the case when some updates have occurred the auxiliary information aux can include the update information produced by these updates.

VC. (C,m, i,_A).
The verification algorithm accepts (i.e., it outputs 1) only if Ai is a valid proof that C was created to a sequence m1, ...,mq such that m = mi.

VC. (C,m,m′, i).
This algorithm is run by the committer who produces C and wants to update it by changing the i-th message to m′. The algorithm takes as input the old message m, the new message m′ and the position i. It outputs a new commitment C' together with an update information U.

VC.Proof (C,_j,m′, i,U).
This algorithm can be run by any user who holds a proof Aj for some message at position j w.r.t. C, and it allows the user to compute an updated proof A′j (and the updated commitment C′) such that A′j will be valid with regard to C′ which contains m′ as the new message at position i. Basically, the value U contains the update information which is needed to compute such values.

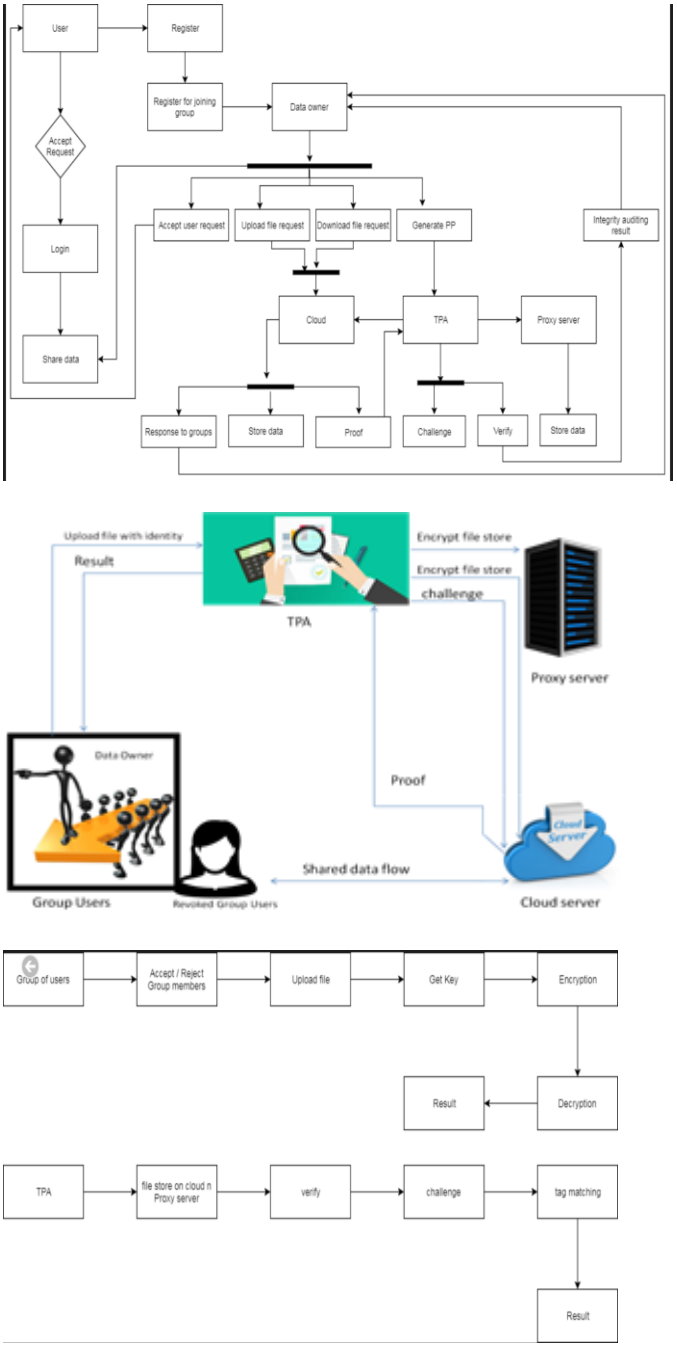
7. DATA DESCRIPTION:

Describing and documenting data is essential in ensuring that the researcher,and others who may need to use the data, can make sense of the data and understand the processes that have been followed in the collection, processing, and analysis of the data.
Research data are any physical and/or digital materials that are collected, observed, or created in research activity for purposes of analysis to produce original research results or creative works.

8. DATA OBJECTS AND RELATIONSHIPS

A data object is a part of the repository whose content can be addressed and interpreted by the program. All data objects must be declared in the ABAP program and are not persistent, meaning that they only exist while the program is being executed. Before you can process persistent data (such as data from a database table or from a sequential file), you must read it into data objects first. Conversely, if you want to retain the contents of a data object beyond the end of the program, you must save it in a persistent form.

9. ARCHITECTURAL DESIGN:



10. FUTURE SCOPE:

A novel privacy-preserving mechanism that supports public auditing on shared data stored in the cloud. In particular, we exploit ring signatures to compute verification metadata needed to audit the correctness of shared data. With our mechanism, the identity of the signer on each block in shared data is kept private from public verifiers, who are able to efficiently verify shared data integrity without retrieving the entire file.

11. MAJAOR CONSTRAIN:

We are building Web base application so user requires IDE for accessing functionalities of developed services. So you can use any device which has web browser.
System will be built on java platform, so it needs around 500 Mega-Byte of space.
The database which is being used for this application is H-Base. As it is a re-mote application, a number of users can access it simultaneously.
To send/receive packets, Internet connection either through GPRS(Service provider) or WLAN.

12. CONCLUSION:

In this, we investigated a new primitive called identity-based remote data integrity checking for secure cloud storage. We formalized the security model of two important properties of this primitive, namely, soundness and perfect data privacy. We provided a new construction of this primitive and showed that it achieves soundness and perfect data privacy. Both the numerical analysis and the implementation demonstrated that the proposed protocol is efficient and practical. Extend this work with Group Management with Forward Secrecy & Backward Secrecy by Time Duration & Recovery of File when Data Integrity Checking Fault Occur.

REFERENCES

[1 P. Mell, T. Grance, Draft NIST working definition of cloud computing, Reference on June. 3rd, 2009. http://csrc.nist.gov/groups/SNC/cloudcomputing/index.htm.

[2] Cloud Security Alliance. Top threats to cloud computing. http://www.cloudsecurityalliance.org, 2010.

[3] M. Blum, W. Evans, P.Gemmell, S. Kannan, M. Naor, Checking the correctness of memories. Proc. of the 32nd Anual Symposium on Foundations fo Vomputers, SFCS 1991, pp. 90–99, 1991.

[4] G. Ateniese, R. C. Burns, R. Curtmola, J. Herring, L. Kissner, Z. N.J. Peterson, D. X. Song, Provable data possession at untrusted stores. ACM Conference on Computer and communications Security, 598-609,2007.

BIOGRAPHIES

ADITYA SHITOLE
Student, Dept of Computer Engineering, A.I.S.S.M.S. Polytechnic Pune, Maharashtra, India

OMKAR DHAPTE
Student, Dept of Computer Engineering, A.I.S.S.M.S. Polytechnic Pune, Maharashtra, India

EMAAD PERZADE
Student, Dept of Computer Engineering, A.I.S.S.M.S. Polytechnic Pune, Maharashtra, India

SUSMITA GAIKWAD
Student, Dept of Computer Engineering, A.I.S.S.M.S. Polytechnic Pune, Maharashtra, India