ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 7, July 2016

Maintaining Integrity and Security for the Data Shared in the Cloud

Gayathri Dili, Anu V.R

Final Year M.Tech Student, Dept. of Computer Science and Engineering, Sree Narayana Gurukulam College of Engineering, Mahatma Gandhi University, Kerala, India

Associate Professor, Dept. of Computer Science and Engineering, Sree Narayana Gurukulam College of Engineering, Mahatma Gandhi University, Kerala, India

ABSTRACT: Cloud computing platform provides global sharing and accessing of resources. Cloud offers data storage and sharing facilities that provides better scalability. Apart from the advantages offered by Cloud, it also finds difficulties in maintaining the integrity and security for the shared data. Public auditing is a mechanism by which the integrity of data could be maintained so that the correctness of data could be verified thereafter. Even if the system could assure data correctness, there may be chances of some security threats. Security must be established for the data shared and one who shares it. So, for that a new form of signing method is to be developed for sharing the data to Cloud, which could verify whether the data is shared by an authenticated user or not. In this paper, we discuss about a system that helps to verify the integrity of data and also to make sure that, signatures are made by an authenticated signer.

KEYWORDS: Controllable Linkability, Group Signatures, Public auditing, Re-signing, Shared data, User revocation.

I. Introduction

Cloud computing is an internet-based computing, which provide shared resources, software and information to users, computers and other devices on demand. Cloud storage allows the users to store data, share it with others users within that Cloud group, which will be further taken for modification and shared back to Cloud. Cloud user computes signature for each block of data he shares to it. When the data is taken and modified by any other user in the group, signature of the data is getting re-computed. Therefore, each block of data can have different signatures made by different users. Computing signature for the data helps in checking the integrity of data [1]. When a user in the group gets revoked, the blocks previously signed by the revoked user are now needed to be re-signed by one of the existing users of that group [4].Once the blocks of data get re-signed the integrity of data must be verified. The re-signing process started with a straight forward method of downloading data, verifying them and re-signing them and finally uploading the re-signed data. This means that the entire tasks are performed by a single user. To reduce the complexities in terms of workload and time, there came the concept of public auditing called Provable Data Possession (PDP) [2] which made use of homomorphic authenticators. This method was first used to check the correctness of data, especially for data stored at an untrusted server. Here the auditing could be done without downloading entire data blocks from Cloud. Later on many improvements were made and concluded with the actual concept of public auditing or public verifiability performed by Third Party Auditor (TPA) [3]. In public auditing the first phase tells about the concept of checking whether the server is trusted or an untrsuted one and whether it contain the original data, that is to ensure the correctness of data and this could be done without retrieving entire data from the data stored server [5]. This concept of appointing a third party member for auditing task improved the efficiency of integrity check as compared to the previous one. However, the correctness of data cannot define its security. When a user's signature is used for data sharing, it results in revealing the user's anonymity to the intruders. Cloud could support the protection of user's identity while he is sharing some files to Cloud. A new form of signing concept could help to tackle this issue and that is by signing with the group's signature [6]. Again the group signature's validity is needed to be checked. That is to confirm that the group signature is made by an authenticated user. This could be done by comparing the two signatures made by a person while he is uploading two messages to the system. This principle of checking whether the two group signatures are generated by same signer is called Controllable Linkability [7].

II. Related Work

In [2] authors proposed a method called Provable Data Possession (PDP) which allowed a public verifier to check the correctness of data which was being stored by the user or a client on an untrusted server. Even though, it offered high privacy for data of the user, it was good for only the static data. An extension to the PDP was introduced in [5]. In this extension model, authors implemented PDP using some symmetric keys which could provide support for the dynamic data. But it couldn't do much for verifying the integrity of data as verifier could only provide limited number of verification request. Later, introduced the Merkel Hash Tree for supporting the public auditing mechanism by providing a complete support for fully dynamic operations.Users or clients who share the data on a storage space were so much worried about how to maintain the integrity of data, as the data became larger and larger the idea of checking the integrity of data by users itself need to get changed and authors suggested the idea to bring the Third Party Auditor (TPA) in [3] to overcome the workload or complexity felt by the users or clients to a greater extent. But protecting the private or confidential data of users from TPA came forward as an issue, but Wang solved it in a better way by random masking. In [8] authors proposed a model "Oruta" which could help in identifying the each of the signers who have signed on the data blocks being shared in that storage space and keep the signer's identity private form the public verifiers and thus provide integrity of shared data without retrieving the entire file. Apart from the other previously discussed mechanisms this could perform multiple auditing tasks. And in [9] authors proposed another model called "Knox". Even if there is large number of users, it is not affecting the auditing of large amounts of data shared by a client and the time taken to audit those data. This could be considered as privacy –preserving mechanism too. But the user revocation and public auditing couldn't be implemented so successfully here. In [6] authors proposed the concept of Group signatures, which was a technique of authorizing the documents, messages or relevant information anonymously on behalf of group by any member belonging to it. Such scheme also involves a group manager, who is able to open any group signature by showing which group member issued it. According to the author the only person capable of addition of new members and revoking of the existing members from the group is the group manager [10]. User's identity maintenance is promoted by this idea of group signature. Later, in [17] the authors proposed the pairing based group signature scheme which is far more efficient regarding bandwidth and computational efficiency. This method was suitable for higher security levels. This paper could also discuss about the basics of Controllable Linkability. A more secure and coalition resistant Group signature scheme was introduced in [11]. This brought the idea of revocable anonymity. This scheme used a registration protocol called JOIN that hides the group member's message. In [12] the authors proposed Short group signatures, which reduced the need for complex calculations for generating lengthy group signatures as in the earlier cases. This satisfies the correctness, full anonymity and full traceability. Later, a group signature scheme for groups with dynamic membership got developed in [13].This paper also discuss about strong formal definitions of security and construction, proven secure under general assumptions. This paper also discuss about different trust levels and the identity privacy of signers and also about traceability. In [14 authors proposed a mechanism that supported concurrent joins and dynamic memberships and revocation situations. The eXtremely short signatures (XS) are used here that provides anonymity with every short signature. The security level offered by this methodology also provides a very efficient signing and verification procedures. And in [15] authors discussed a group signature mechanism with an added on idea of Controllable Linkability with this Group signatures. This system could support dynamic groups and also provided a fine grained control over the release of user information. This found importance in privacy preserving data mining applications and customized anonymous authentication and thus promote controllable linkability. The signature scheme used short signatures and reduced the need for complex calculations.

III. Existing System

Cloud computing is the architecture for providing computing service via the internet on demand and pay per user access to a pool of shared resources namely networks, storage, servers, services and applications, without physically acquiring them. So it saves managing cost and time for organizations. Cloud storage is a service where the data is remotely maintained, managed, backed up. It allows users to store their files online, so that they could access it via internet from anywhere at any time. In single storage cloud system each cloud customer's data is stored on single higher configuration server. Even if that server has huge amount of resources such as RAM, Hard disk, processing power, it has certain limit. If it crosses that limit then particular resource performance slows down. It may loss the data or does not provide the services. Cloud computing is efficient and scalable, but maintaining the stability of processing so many jobs in the cloud computing environment is a very complex problem with load balancing, which receives much attention for researchers. The Cloud concept helped the users to outsource the data to Cloud rather than just storing them in the local devices. Once the data get shared in the Cloud, it is accessible for all the users in that group. They can read the data, modify it and whatever the modifications done will be reflected to all those who use the data. For each data shared in the Cloud any one of the user go for generating signatures for the data blocks. If users go for modification of the data, re-signing of the data also takes place. If the Cloud possess user's private key, then it can finish the re-signing task without asking the users to go for downloading the data blocks. But security issues are more if the Cloud keeps something that is private to the users. Another important thing to be noted is the need for verifying the correctness of data after it is shared to Cloud after re-signing. A public auditing mechanism was developed, that helped to check the correctness of data and could provide efficient user revocation in the Cloud. Panda is also known as Homomorphic Authenticable Proxy Re-signature (HAPR) scheme that support the blockless verifiability and non-malleability [16].

- Blockless verifiability allows the verification process or checking the correctness of data by considering linear block of data, without downloading/retrieving entire data.
- Non malleability shows that users, who do not possess private keys, cannot generate valid signatures.

Panda makes use of the idea called proxy re-signature to find the key of next user to which the data is needed to be re-signed, when a user got revoked from the group.

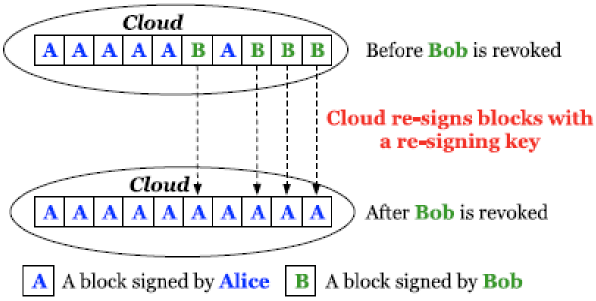


Fig.1. When Bob is revoked, the Cloud re-signs the blocks that were previously signed by Bob with a re-signing key.

The Cloud could act as the Third Party Auditor who could re-sign the data without downloading the entire blocks, to the key specified by the proxy (fig.1). The user is now completely free from the burden of re-signing the data and verifying it. Thus the time taken and cost of communication and computation resources by downloading data, verifying it, resigning and finally uploading could be easily saved.

A. *Properties:*

- Correctness: The public verifier is able to correctly check the integrity of shared data.
- Efficient and Secure User Revocation: On one hand, once a user is revoked from the group, the blocks signed by the revoked user can be efficiently re-signed. On the other hand, only existing users in the group can generate valid signatures on shared data, and the revoked user can no longer compute valid signatures on shared data.
- Public Auditing: The public verifier can audit the integrity of shared data without retrieving the entire data from the cloud, even if some blocks in shared data have been re-signed by the cloud.
- Scalability: Cloud data can be efficiently shared among a large number of users, and the public verifier is able to handle a large number of auditing tasks simultaneously and efficiently.

B. *System Model:*

The system model [16] includes three entities: the cloud, the public verifier and users who share data (fig.2). The cloud offers data storage and sharing services to the group. The public verifier, such as a client who would like to utilize cloud data for particular purposes (e.g., search, computation, data mining, etc.) or a third-party auditor who can provide verification services on data integrity, aims to check the integrity of shared data via a challenge-and-response protocol with the cloud. There is an original user who is the original owner of the data who creates and shares data with others users in the group, so that the data could be accessed by them as well as other users and will further go for its modification. Data stored in the Cloud is divided in to blocks and with each block of data a signature is computed by one of the users in that group. When the shared data is initially created by the original user, he computes signature for the data. Later when data is modified by any other user, he goes for re-signing the data with his/her private key. This makes the data blocks to have different signatures. When a user gets revoked from the group, data signed by him remains within the Cloud and signature computed by them for those data blocks become invalid. It is now needed to be re-signed by any of the existing users in the Cloud using their private key. Still the integrity can be verified by utilizing the public keys.

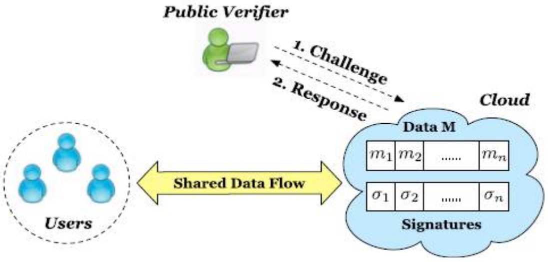


Fig.2. Verifier performs challenge-response with Cloud (database server) to perform the verification of data.

C. *Working of Existing System:*

The proxy re-signature scheme includes six algorithms [17]:

1. **KeyGen:** Soon after the user u_i joins the Cloud group, private key pk_i and private key sk_i $_=$ π_i are different for each users, or we can say it is unique. Here, the ids of all the users in the group are added to a user list (**UL**) or **REG** which keep the details of active users.
2. **ReKey:** This step comes after the user revocation, where the re-signing of data is needed to be done and data will be re-signed by new keys or new signer, who is the other existing user of the same group. Thus the Cloud performs the job of creating a new key (re-signing keys r_k) for the data ownership. And here the Cloud choses a random number say r and send to the user side, the user will be making a calculation that is r/π_i and send that value to next user $u_{j,}$ and u_j makes a calculation $r\pi_j/\pi_i$ and send it to Cloud, the Cloud then recovers the resigning key by π_j/π_i□□
3. **Sign:** The original user could share data in the cloud by computing a signature and using that signature to sign on each data they share. Here total number of blocks in shared data is n and the shared data is represented as $Ms = (m_1,m_2,\ldots m_n)$. The user u_i share data to Cloud with a block id of id_k and the signature for the shared data m_k is given as $\alpha_k = (H\,(id_k)\,w^m{}_k)^{\pi}{}_i\ \varepsilon G_1$,

where α_k is the signature of shared data m_k,, H is a hash function, id_k is the id of the data block, w is the generator of multiplicative cyclic group of prime order $G_{1.}$

4. **ReSign:** A user is revoked from the group, and the cloud re-signs the blocks of revoked user to any of the existing users in the Cloud group which was calculated after **ReKey**. For calculating the new signature with which the next user is going to sign the data (represented by α'_k)we used the re-signing key r_k, public key $pk_{i,}$ signature α_k ,the shared data id m_k and the block identifier id_k. The Cloud verifies whether the initial signature calculation is done in the right way or not and then use steps in **ReKey** to get the key to which the re-signing is needed to be done and then use the steps in **Sign** to calculate the new signature with which the existing user will sign. Thus the new signature will form the signature of the data after being re-signed by the new user and is calculated as follows,

$\alpha'_k = (H\,(id_k)\,w^m{}_k)^{\pi}{}_i{}^{.\pi}{}_j/{}^{\pi}{}_i = (H\,(id_k)\,w^m{}_k)^{\pi}{}_j$

Now the revoked user u_i is removed out of the **(UL) REG.**

5. **ProofGen:** The verification on data integrity is performed via a challenge-and-response protocol, between the cloud and a public verifier. More specifically, the cloud is able to generate a proof of possession of shared data under the challenge of a verifier. And the Cloud performed the auditing by taking the request from verifier, which contain the resource details for which the auditing is needed to be done and thus to match/ verify that the Verifier specified data is maintained as such by the Cloud.
6. **ProofVerify:** A public verifier is able to check the correctness of a proof responded by the Cloud when they are asked to perform the auditing so as to verify integrity. The auditing proof generated will be having the resource

details that should match with the details specified in the Verifier's request. If it matches the integrity is maintained. Thus completed the user revocation and re-signing procedure. And this is how the existing system panda could prove the integrity maintenance. Since it is a proxy based model, the presence of proxy can be also inlcuded in the system model in fig.2. Thus the system model for our Panda will look like fig.3.

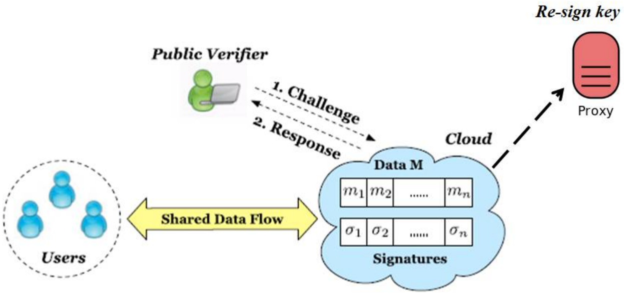


Fig.3. Diagrammatic representation of working of Panda

The panda system constitutes six modules. They are cloud group, user, admin, auditor, proxy, and verifier. Cloud group is the group created by an Admin who is the Group manager and one who owns the group. The user is the member of the Cloud group who joins the group. The user's request to join the Cloud group will be handled by the Admin of the Cloud group. The admin could handle the user request, generate private, public keys for the users. Admin can also revoke them out of the group. Auditor forms the Third Party Auditor, who performs the data re-signing and monitor the revocation procedures and generate new signatures for re-signing. Proxy will assist the Auditor in choosing random keys to which the re-signing takes place. Finally the Verifier raise the need for auditing, after the completion of re-signing to make sure the correctness of data is maintained after re-signing by the Auditor. Here in fig.4 the user first join the Cloud group. Once he became a group member of the Cloud group, he could share data to Cloud. Each data block will be having different signatures, as it is signed by different users after modifications are done, if any. Later, when a user is revoked from the group, the admin will inform Cloud about the revoked user and now Cloud think about re-signing the data to an existing user in the group. Cloud will send a request to proxy asking it to choose a new

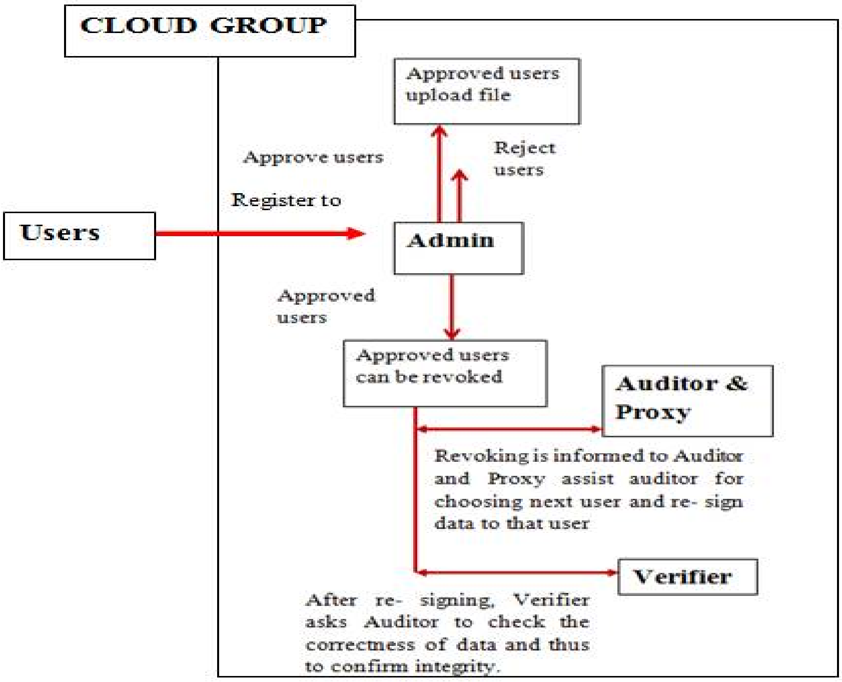

CLOUD GROUP
Approved users upload file
Approve users
Reject users
Register to
Users
Admin
Approved users
Approved users can be revoked
Auditor & Proxy
Revoking is informed to Auditor and Proxy assist auditor for choosing next user and re- sign data to that user
Verifier
After re- signing, Verifier asks Auditor to check the correctness of data and thus to confirm integrity.

Fig.4. Block diagram showing working of Panda

key to which data is needed to be re-signed. The re-signing keys are chosen randomly and further it is sent to Cloud and then Cloud re-sign the data to the key specified by proxy. Further, verifier come in action and will perform a challenge and response protocol session with Cloud, asking it whether the data re-signed to the other user is exactly the same data of the revoked user, i.e. whether the correctness of data is maintained or not. Cloud performs auditing mechanisms to prove that the data correctness is maintained and send audit proofs back to verifier. Verifier will now go for verification of the response or the audit proofs from Cloud. Thus completes the re-signing job. The working of Panda can be understood from the diagram (fig.4).

IV. Proposed System

Panda could provide an efficient mechanism by which the scalability, reliability and integrity of data could be maintained. However data shared in Cloud is still having a deficiency of security. Here the security is implemented in terms of privacy. Privacy can be characterized in to two forms anonymity and unlinkability. Anonymity means the property of concealing the identifiable information or user's identity in authentication messages and unlinkability means that given two authentication messages, an unauthorized entity cannot tell whether they are generated by the same user or not. Generally speaking, for accessing a service, users prefer to preserve their privacy, but the service provider may want to relax their privacy to gain sufficient user information. For an application environment, privacy needs to be adjusted according to the desired policy or reasonable expectation of profit [7]. The privacy could be preserved by two solutions. One is pseudonym system and the other is group signature. The pseudonym system could support anonymity, but a signer cannot avoid being linked by anyone who obtains their signatures. Whereas, the group signature (GS) scheme is considered as one of the most versatile primitives for anonymity. However, following the concept of a traditional GS (or referred to as a normal GS), the linkability is given only to an opener, who is not usually a service provider but a special group manager. A verifier or a service provider who needs individual statistics on transactions, such as a consumer's buying pattern, based on the linkability of the transactions cannot obtain these statistics without the help of an opener. Involving an opener for linkability introduces a bottleneck that is not only require an extra on-line process but also require a strong trusted relationship between the user and the opener, since the opener breaks the anonymity. Our proposed methodology introduced a technique of authorizing the documents, messages or relevant information anonymously on behalf of group by any member belonging to it, and is termed as a Group Signature scheme, where the group consists of a manager and valid members [10]. The system modified the traditional Group Signature idea by adding a concept called Controllable Linkability [7]. Controllable Linkability focuses on authorization checking. Here a signer's signature looks random as it is generated by the group in which they belong to. So his or her identity is always kept hidden, and the signatures can be anonymously linked under a linking key, secretly managed by a Linker. Since the linkage information cannot reveal user's identity with the linking key, we can establish a fine-grained control on anonymity by adding this CL to the controllable anonymity of a GS [15].

The system mainly focuses on hiding the details of users within that Cloud group and the data shared by them. A user can start his session only when he input a token/ linking key, which is further taken for verification. Once the user's identity is confirmed user is now free to share his data. However, the current session of the user ends after ten minutes of time interval and he need to renew that session by entering the linking key again when it is asked. On expiring the session, the user is considered to be temporarily revoked from that Cloud group. So the intruders cannot see that user's details and the files he has shared. Intruders cannot even figure out the existence of such a user in the Cloud group. Users who are temporarily revoked from the Cloud group can restart their data sharing session by choosing a renew option. In order to renew their session they need to provide a token or a key which should match with the token which was provided to them at the time when they got approved to that Cloud group. That means this token is acting like a link which connects the user and his sessions. So that token or key controls the connection between him and his profile operations. Thus the proposed system can implement the idea of checking the group signatures, i.e. to check whether the user who logs in and given the access token to share or download file is exactly the same person who is a member of the Cloud group. The proposed system use a PS-OL scheme i.e. Privacy-protecting Signature scheme with both Opening and Linking capabilities in a controllable manner [7]. PS-OL has adapted the model of CL-GS scheme (Controllable Linkability Group Signature) and made a security model by modifying that. Here the anonymity and unlinkability can be controlled by keys. Earlier versions of PS-OL separated the linkability proving property from Opener to Linker, thus the bottleneck of additional online processing and strong trusted relationship could be eliminated. But this PS-OL had some deficiencies, it couldn't support the dynamic membership cases and the signature length was long. It even needed complex computations. To reduce these overheads, there came the proposed system,

DOI: 10.15680/IJIRCCE.2016. 0407002

which is capable of supporting dynamic groups, where adding new users and revoking conditions can be satisfied. One thing to be noted is that, since our linkability for dynamic membership enables an opener to obtain a static linking tag uniquely associated with a signer, computation overhead for opening can be significantly reduced.

A. *Design Objectives:*

- *Correctness:* Signatures generated by an honest user should be verified correctly. With inputs of a message and a signature, Open should correctly identify the signer. Link should link the signatures from a signer.
- *Tracebility:* The origin of the signature should not be identified, i.e., a (public) proof to check the origination should not be forged.
- *Non-Frameability:* Colluding users with even trusted authorities should not forge a signature of an honest user.
- *Linkability:* Three notions for linkability are *E-linkability, JP-Unforgeability* and *LO-linkability*

E-linkability also called as Enforced linkability captures that colluding users should not be able to generate two pairs of a message and a signature satisfying any of the following conditions, even with help of authorities such as the Linker or Opener. *JP-Unforgeability* also called as Judge-Proof Unforgeability Captures that a linking key cannot be used for generating a Judge proof, which completes the representation of Linker's restricted capability. *LO-linkability* also called as Link–Only linkability captures that a linking key should be used only for linking signatures, not for gaining useful information for opening.

B. *System Model and its Working:*

The proposed system comprises of eight modules, they are cloud group, issuer, user, opener, linker, auditor, proxy, and verifier. The cloud group is the group which is created by issuer. Issuer is the coordinator or admin of the cloud and he is also the group manager, who issue private key, public key and linking key, verify file uploads and shared data, approve the user requests, renew the sessions on receiving linking key and can also verify the linking key, verify file uploads and shared data, approve the user requests, renew the sessions on receiving linking key and can also verify the linking key. The users are the members of the cloud group who register to the group, who can upload files or share data to cloud and they need to input the linking key to start a new session and to renew it. Opener is an entity that is responsible for opening the messages after being verified by the issuer. Linker can deal with every user's information with a linking key and enable the entity which has a linking key to find whether two group signatures were generated by same signer or not. The auditor performs the same job as in the proxy re-signature mechanism that is, they could handle the user revocation, re-signing of data to another existing user. Proxy is the entity that assist auditor to choose re-signing keys and finally, verifier does the job of requesting the auditor to get a proof showing the maintenance of data integrity. The working of our proposed system is explained in fig.5. Here the proposed system has two cases. The first case deals with proving controllable linkability (security) and second case shows how to prove integrity.
CASE A: Proving the Controllable Linkability -The user starts his session by giving linking key along with the data he wants to share. The Issuer take the user input and verifies it, to check whether it is a valid linking key or not and check the group signature. After verifying the linking key, the Issuer forwards the data to Opener for opening. The user can continue in his session for about ten minutes and after that his session expires. Thus he will be in temporary revocation list. To renew the session, the user needs to request for updating the group public key and user signature key. After opening the messages by the Opener, the session now proceeds to Linker side, where the linkability can be proved by checking whether the two group signatures with which the messages has been uploaded is generated by same signer. Unlike the existing methodology, here the user is not signing with his signature, but with the Group signature. So, the user's identity is no longer revealed to an external entity.
CASE B: Proving the Integrity -The Issuer can also revokes a user permanently. At that time the data shared by him will be re-signed by any other user in the Cloud group. This re-signing is handled by Auditor. Auditor works with Proxy to choose the re-signing keys and re-sign data to the new user. Now the data will be signed in the name of that new user. Once the re-signing is over, Verifier asks for correctness check. The Verifier requests Auditor to audit the data and make sure that the data re-signed by the new user is exactly the same one that was shared by the revoked user. Thus auditor performs auditing and sends that auditing proof to Verifier.

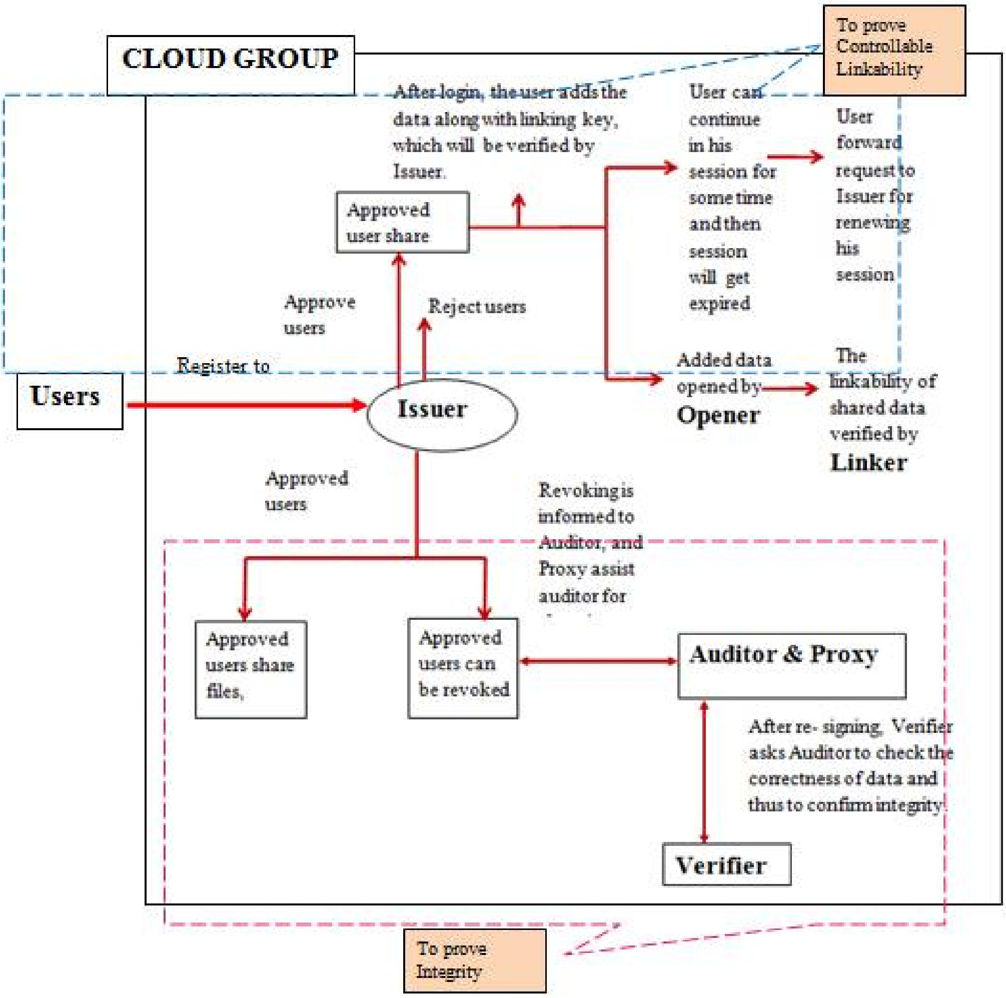

CLOUD GROUP
To prove Controllable Linkability
After login, the user adds the data along with linking key, which will be verified by Issuer.
User can continue in his session for some time and then session will get expired
User forward request to Issuer for renewing his session
Approved user share
Approve users
Reject users
Register to
Users
Issuer
Added data opened by
Opener
The linkability of shared data verified by
Linker
Approved users
Revoking is informed to Auditor, and Proxy assist auditor for
Approved users share files,
Approved users can be revoked
Auditor & Proxy
After re- signing, Verifier asks Auditor to check the correctness of data and thus to confirm integrity
Verifier
To prove Integrity

Fig.5. Working Model

Thus the combination of Group signature concept and Panda mechanism could guarantee the integrity and security (via controllable linkability) of the data shared in the Cloud. However, in the proposed system the information to identify a user or link signatures is not revealed publicly from the signature. This information can only be appropriately recovered with an opening key or a linking key. In our proposed system, we are keeping a temporary revocation list called **RL**, which keeps the details of the users who are temporarily revoked from the system i.e., the users' whose session got expired. The session expiring is done as a part of security. The session can be renewed only with the help of linking key. The linking key is a key that is unique for each of the users and is a collection of some random numbers. Once the user logged in he start his session by giving this linking key so as to initiate message sharing. The user can continue in the session for some time and after that his session expires. The user can continue within his session and share data for about ten minutes and if the session expires he can go for further sharing only when the session is renewed by the issuer (admin). The user may request the issuer for renewing the session and, after the renewing, the user can restart his session by giving the linking key. The **RL** used here contain an index and private information for a user who has been revoked. This revocation list is managed by the Issuer. The **RL** contain the details of the users who are temporarily revoked and this list helps to update or renew the user's private key and group public key. This renewal will remove the user who has been put in to **RL** back to list of active users (**REG** OR **UL**). Here the system also keeps another list called **REG** which stores the details of users who joins the Cloud group. The proposed system has two sections one sections is to prove integrity and the other one to prove controllable linkability (security). Here, we will discuss some algorithms for integrity proving and controllable linkability proving. The algorithms to prove integrity is same that was used in section III

The algorithms used to prove controllable linkability are:

1. **SetUp:** This will set up the Cloud group to which the members are needed to be added.
2. **(UserJoin,Issue):** The user join the Cloud group and soon after the approval of his registration the user will be supplied with private key sk_i and public key pk_i. and *Mlk* which is the message linking key or linking key which is unique. The issuer also develops a group public key gpk_0 which is the group public key and s user signature key usk_i. Once the user's registration got approved he will be added on to the **REG** (**UL**).
3. **GSig:** The user can now share data to Cloud, for that he need to sign the data/message *M* with group signature *s* other than his signature *α*.The user also need to specify the linking key *Mlk* which start the user's session. The user can continue in the session for ten minutes and after that his session expires and he will be moved to **RL.** The user can restart his session by sending a request to renew the session to issuer. That request will contain the user details with which the issuer can identify which is that user. The user can restart the session by giving the linking key *Mlk* after the session renewal.
4. **GVfy:** The data/message shared by the user will be verified by the issuer by checking the signature (*s*) on that data and make sure that the data is sent by an authorized user and whose identity is not revealed. Soon after the verification the data is forwarded to opener.
5. **Open:** The opener will receive the data (*M*) forwarded from issuer after verification. The details available to opener about the message *M* are, the message itself and the signature *s*. If there is two messages (from same signer) to be opened then details are (M_1,s_1) and (M_2,s_2).
6. **Judge:** Once the message (data) is received, the opener makes a judgment proof *β* which shows that the message has been signed with an authorized group signature. This judgment proof contains the message *M*, the signature *s* and the user public key upk_i.
7. **Link:** The data from the opener part is fed to linker for linkability proving. The linker proves linkability by taking details *Mlk*, (M_1,s_1), (M_2,s_2) i.e $(Mlk, (M_1,s_1),(M_2,s_2))$. If it outputs 1 then the signatures are generated by same signer.
8. **Revocation:** The temporarily revoked user can be handled by this step, where the user after expiring his session will be moved to **RL** and when he request to update the session or renew the session, the issuer update the group public key gpk_0 and user signature key usk_i ,which was generated at time of user's joining to Cloud group. After the renewal the user is moved from **RL** to **REG,** where **REG** shows the list of active members.

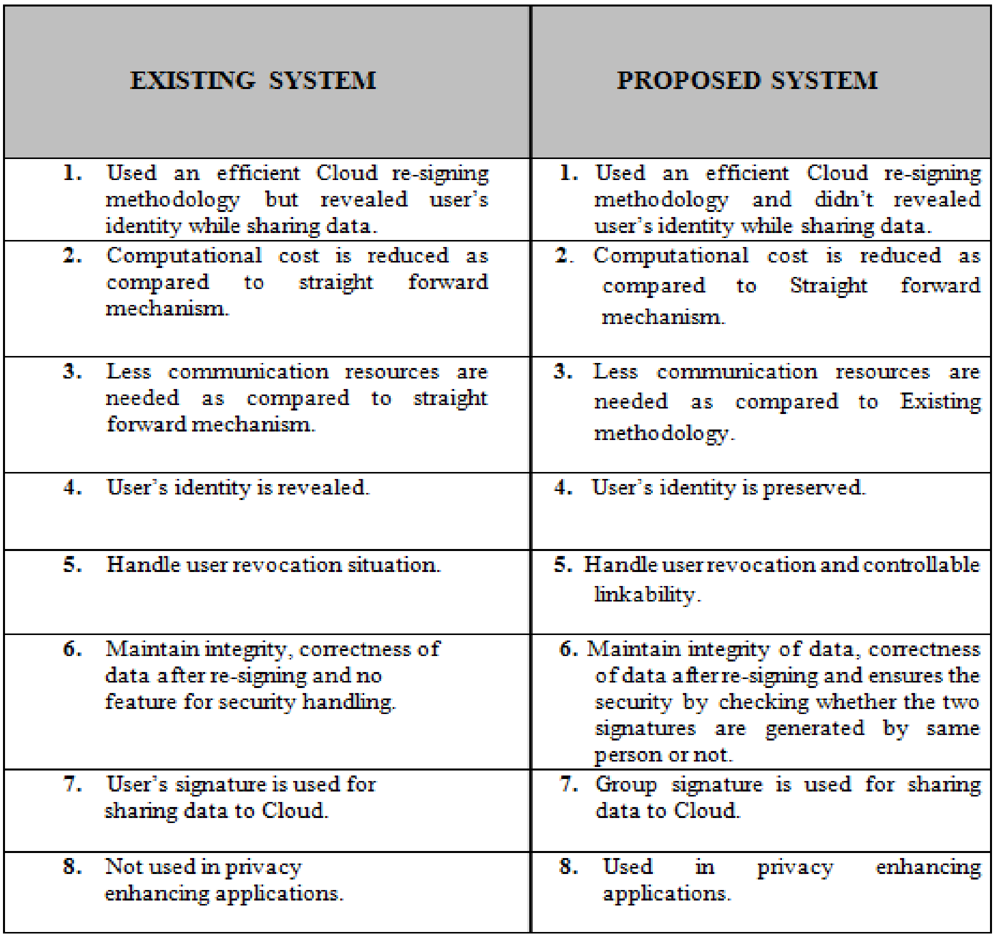
V. Implementation Details

Our proposed system is designed and implemented with the help of a Hospital application which runs on a Cloud platform (i.e. Cloudera –Quickstart-VM-5.4.2.0-vmware. Here the extension part (Controllable Linkability) is added as a feature to the existing system Panda. So the name of the system can be also termed as Panda + Controllable Linkability. Eight entities are included in the application. They are hospital which is a cloud group that was created by hospital admin. Hospital Admin (issuer) is one who created the hospital (cloud group), issues the keys like private, public and linking keys for the doctor (user) after they join the Cloud group, they can also renew the sessions when it is requested by the doctor. Opener will take the messages given by the admin after verification and open it as they found that the messages are signed with a valid group signature and is signed by a user who belongs to a valid group. The linker is another entity that accepts the data shared from opener side and check whether the signatures are made by same user or not. The doctors are the users of the Cloud group (hospital) who joins to share data (message) and uploading files to Cloud. Their registration is approved by the Hospital Admin. The auditor is responsible for handling user revocation situations and further calculates re-signing key by assisting proxy and therefore, re-sign data to new user. The proxy assist auditor for choosing the re-signing keys at the time of user revocation. Finally, the verifier is responsible for requesting the auditor to prove that the data re-signed by new user is exactly the same data once shared by the revoked user. The application takes some Cloud space. A Hospital Group could be accommodated in the Cloud. A Hospital Admin can accommodate many Doctors (users). The application gives options for registering Doctors. Doctor's registration is approved/ disapproved by the Hospital Admin (Issuer). Once the Hospital Admin approved the Doctor's request, he generates private, public and linking keys for that Doctor. As mentioned in the system model (in section IV), our proposed system has two sections. One is proving the controllable linkability and second is to prove integrity. In controllable linkability part we are considering messages as the data getting shared to Cloud. And in the integrity proving part we are taking file uploads as the data getting shared to the cloud. The application proves integrity and controllable linkability in the same way as explained in section IV. Thus the modules of the system are hospital which is the cloud group, doctor which is same as user, hospital admin same as issuer, linker, opener, auditor, proxy, verifier.

VI. RESULTS

The Working of Panda could be implemented very efficiently using a Hospital Application. The various modules in the application could represent the different entities of system model. After re-signing data to the new user, the integrity check could be done efficiently without affecting the other data stored in the Cloud. The idea of including the Controllable Linkability concept helped to prove the linkability of group signatures, i.e., to prove that the two signatures are being generated by same person or same signer. And implementation of the system by splitting it in two parts helped to prove the Integrity and Controllable Linkability concepts. Implementing Panda in the form of application helped to make the concept simpler and understandable. Cloudera helped to develop a cloud environment for our system. The idea of splitting the Opener's task to Issuer and Linker as compared to previous version of proposed system reduced the complexity of Opener in verification and linking. However, adding the concept of Controllable Linkability as a feature to the existing methodology, Panda could implement the concept of security and integrity maintenance together in a better way. The comparison between existing system and proposed system is shown in Table.1. This comparison shows the drawbacks of existing system and how it can be corrected using the idea introduced in the proposed system.

Table.1 Existing System vs. Proposed System



EXISTING SYSTEM	PROPOSED SYSTEM
1. Used an efficient Cloud re-signing methodology but revealed user's identity while sharing data.	1. Used an efficient Cloud re-signing methodology and didn't revealed user's identity while sharing data.
2. Computational cost is reduced as compared to straight forward mechanism.	2. Computational cost is reduced as compared to Straight forward mechanism.
3. Less communication resources are needed as compared to straight forward mechanism.	3. Less communication resources are needed as compared to Existing methodology.
4. User's identity is revealed.	4. User's identity is preserved.
5. Handle user revocation situation.	5. Handle user revocation and controllable linkability.
6. Maintain integrity, correctness of data after re-signing and no feature for security handling.	6. Maintain integrity of data, correctness of data after re-signing and ensures the security by checking whether the two signatures are generated by same person or not.
7. User's signature is used for sharing data to Cloud.	7. Group signature is used for sharing data to Cloud.
8. Not used in privacy enhancing applications.	8. Used in privacy enhancing applications.

VII. CONCLUSION

Cloud emergence transforms the way in which IT infrastructure is constituted and managed through consumable services for infrastructure, platform, and applications. Panda implementation helped a lot in conducting public auditing of data, so as to check the correctness of data at the time of user revocation. Panda helped to improve the efficiency of user revocation and for the faster re-signing of data blocks with less time, less computation cost and communication resources. But our existing system lacked in security. In order to provide better authorization capability a new methodology was introduced, which kept a linking key concept to enable secure data sharing to Cloud. The expired users are moved to a temporary revocation list. Thus the attackers think that, there are no users in that particular Cloud group and further, intruders are failed to access the shared data. Every time, when the user tries to share data to Cloud, he is authenticated by confirming the link made between him and his profile operations. This link or connection is controlled by means of a concept called Controllable Linkability which performs authorization checks by using some token/key. The linkability proving can make sure that the authenticated users have made the signature on the shared data. This methodology could preserve the anonymity and restrict the intruders to a greater extent. Thus the proposed system could ensure data privacy, user's identity privacy, and data mining versatility, maintain correctness of data, reduce communication cost and computational overhead.

REFERENCES

1. B.Wang, B.Li, and H.Li, "Public Auditing for Shared Data with Efficient User Revocation in the Cloud," Proc. IEEE INFOCOM, pp. 2904-2912, 2013.
2. G.Ateniese, R.Burns, R.Curtmola, J.Herring, L.Kissner, Z.Peterson, and D.Song, "Provable Data Possession at Untrusted Stores," Proc. 14th ACM Conf. Computer and Comm. Security (CCS'07),pp. 598-610, 2007.
3. C.Wang, Q.Wang, K.Ren, and W.Lou, "Privacy-Preserving Public Auditing for Data Storage Security in Cloud Computing," Proc .IEEE INFOCOM, pp. 525-533, 2010.
4. Gayathri Dili, Assoc.Prof.Anu V.R, "Public Auditing Mechanisms to Protect the Integrity of Data Shared in the Cloud" ,IOSR Journal of Computer Engineering (IOSR-JCE) e-ISSN: 2278-0661,p-ISSN: 2278-8727 PP 27-34
5. G.Ateniese, R.D.Pietro, L.V. Mancini, and G.Tsudik, "Scalable and Efficient Provable Data Possession, "Proc. Fourth Int'l Conf. Security and Privacy in Comm. Networks (ICST SecureComm'08),2008.
6. D.Chaum and E. van Heyst, "Group Signatures," in Advances in Cryptology (Lecture Notes in Computer Science), vol. 547. Berlin, Germany: Springer-Verlag, 1991, pp. 257–265.
7. Jung Yeon Hwang, Liqun Chen, Hyun Sook Cho, and DaeHun Nyang ,"Short Dynamic Group Signature Scheme Supporting Controllable Linkability", IEEE Transactions on Information Forensics and Security, Vol. 10, No. 6, June 2015.
8. B.Wang, B.Li, and H.Li, "Oruta: Privacy-Preserving Public Auditing for Shared Data in the Cloud," Proc. IEEE CLOUD, pp. 295-302, 2012.
9. B.Wang, BaochinLi, Hui Li,"Panda:Public Auditing for Shared Data with Efficient User Revocation in the Cloud", IEEE Transactions on Services computing, Vol.8,2015.
10. Subhra Mishra Tilak Rajan Sahoo "A Survey on Group Signature Schemes ", Department of Computer Science and Engineering National Institute of Technology Rourkela, 2014.
11. G. Ateniese, J. Camenisch, M. Joye, and G. Tsudik, "A practical and provably secure coalition-resistant group signature scheme," in *Advances in Cryptology* (Lecture Notes in Computer Science), vol. 1880. Berlin, Germany: Springer-Verlag, 2000, pp. 255–270.
12. D. Boneh, X. Boyen, and H. Shacham, "Short group signatures," in *Advances in Cryptology* (Lecture Notes in Computer Science), vol. 3152. Berlin, Germany: Springer-Verlag, 2004, pp. 41–55.
13. M.Bellare, H. Shi, and C. Zang, "Foundations of group signatures: The case of dynamic groups", in *Topics in Cryptology* (Lecture Notes in Computer Science), vol. 3376. Berlin, Germany: Springer-Verlag, 2004, pp. 136–153.
14. C. Delerablée and D. Pointcheval, "Dynamic fully anonymous short group signatures", in *Progress in Cryptology* (Lecture Notes in Computer Science), vol. 4341. Berlin, Germany: Springer-Verlag, 2006, pp. 193–210.
15. J. Y. Hwang, S. Lee, B.-H. Chung, H. S. Cho and D. Nyang, "Group signatures with Controllable Linkability for dynamic membership", *Inf. Sci.* vol. 222, pp. 761–778, Feb. 2013.
16. B.Wang, BaochinLi, Hui Li, "Panda: Public Auditing for Shared Data with Efficient User Revocation in the Cloud", IEEE Transactions on Services computing, Vol.8,2015.
17. Daniel Slamanig, Raphael Spreitzer and Thomas Unterluggauer , "Adding controllable linkability to Pairing-Based Group Signatures For Free", 17th International Conference on Information Security (ISC 2014).

BIOGRAPHY

Gayathri Dili is doing her Master of Technology (M.Tech) degree in Computer Science and Engineering at Sree Narayana Gurukulam College of Engineering, Kadayiruppu, Kerala, India. Her research interest is Cloud Computing.

Anu V. R is working as Assoc.Professor at Sree Narayana Gurukulam College of Engineering. She received her B.Tech in Computer Science from Cooperative Engineering College Vadakara under CUSAT and M.Tech in Computer Science from DR.MGR University, Maduravoyal, Chennai. Her research interests are Virtual Machines, Virtualization, Cloud Computing, and Computer Architecture.