USER REVOCATION WITH PUBLIC AUDITING FOR SECURE CLOUD DATA SHARING

M. Anandakumar
Department of Computer Science and Engineering,
Arasu Engineering College, Kumbakonam, India
*anandlogo@gmail.com

J. Israelin Insulata
Department of Computer Science and Engineering,
Arasu Engineering College, Kumbakonam, India
*info2intelligent@gmail.com

S. Dilipkumar
Department of Computer Science and Engineering,
Arasu Engineering College, Kumbakonam, India
*sdilipkumar85@gmail.com

ABSTRACT

Cloud storage auditing refers to the verification of exactness of shared data in cloud. Different users from different groups share different data in cloud. Privacy protection becomes the biggest question mark in cloud's data services. To ensure perfection of the cloud data and for efficient user revocation, Third Party Auditing is to be done with novel Collusion Resistant Proxy Re-signature Scheme. When a user gets revocated, the cloud is much able to resign the data chunks; these data chunks were already acknowledged by the repudiated user. But, now the cloud re-stamps with a re-signing key, and this process enhances the potency of the system. Moreover, Auditing has to be done over shared data which heightens the reliability of cloud data.

Keywords: Cloud computing, Data Sharing, Privacy Protection, Proxy Re-signature, User Revocation, Third Party Auditing.

INTRODUCTION

In the present-days, plenty of data are getting created and consumed. Cloud services are terribly useful for storing and maintaining such huge data. Users can store, share and modify any data in the cloud. The cloud gives assurance about the reliability regarding the data which are being stored in cloud, but, it may get hazarded either because of hardware errors or because of software errors and human flaws [1].

For safeguarding the data, Data Owner should verify the perfection criteria regarding the data. Different types of mechanisms were proposed for verifying perfections in cloud data [2]. All antecedently mentioned mechanisms are based on an attestation that has to be affixed to every data chunks; whereas, trustworthiness of all signatures denotes peak integrity level of data. Public auditing is proposed for ensuring the exactness of data under various system and the security models [2] and [3].

In shared data, once the user makes changes in blocks, the user should compute new stamp for it is to be attached to the changed block. Because of many alterations done by many users, various chunks are stamped by various users [4]. While coming to the concern of security purpose, the blocks ought to be acknowledged by proxy [5]. Bleumer and Strauss have proposed Proxy re-signatures. Here, the translator between users A and B is Half-trusted Proxy. Proxy will convert a stamp of 'A' into stamp of 'B' on same message for translation. The signing key which gets generated will not be learned by proxy and it cannot sign any whimsical data in support of either A or B [6].

NOVEL COLLUSION RESISTANT PROXY RESIGNATURE TECHNIQUE

In this proposed public verifying technique, by using novel proxy re-signatures technique, the cloud should re-acknowledge the blocks, by using a re-acknowledging key, when misbehaved or left users in group, gets revoked from group. These re-acknowledged blocks have already been acknowledged by the repudiated user. By utilizing this novel technique, the potency of repudiation of users' rights has been escalated, reckoning process is made easier and assets have been saved easily. If cloud that may not reside in very identic trusty realm with every user, subsequently that cloud has the ability to only convert a revoked user's signature into existing user's signature in the identic block. Nevertheless, it has no ability to acknowledge discretionary data chunks in support of neither one.

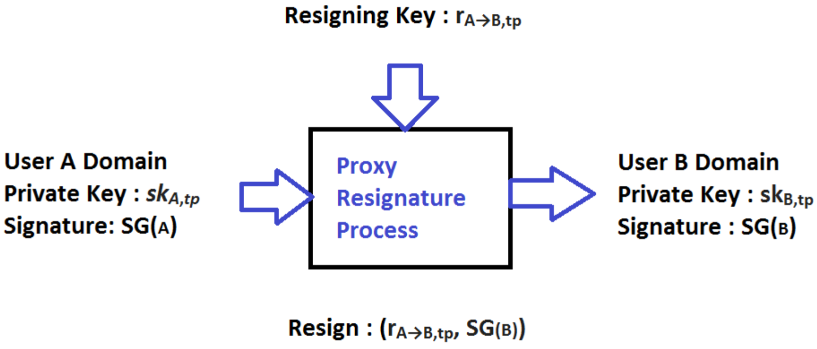

Resigning Key : $r_{A\to B,tp}$
User A Domain
Private Key : $sk_{A,tp}$
Signature: SG(A)
Proxy
Resignature
Process
User B Domain
Private Key : $sk_{B,tp}$
Signature : SG(B)
Resign : ($r_{A\to B,tp}$, SG(B))

Fig.1 Proxy Re-Signature

This table below depicts the differentiation among the duo techniques for-instance BBS scheme [12] and Ateniese scheme [8].Here, P_2 represents duplex proxy resignature scheme whereas P_1 represents single directional proxy re-acknowledgement technique.

S.NO	CHARACTERISTICS	BBS	P_2	P_1
1.	Single directional	x	x	✓
2.	Several use	✓	✓	x
3.	Non-public Proxy	x	✓	x
5.	Clear	✓	✓	✓
7.	Mutual	x	x	✓
8.	Non subjective	x	x	✓
9.	Short-term	x	x	✓

We have implemented a new method which is proxy re-stamp with multiple beneficial things, which ordinary proxy re-acknowledgements do not possess. Our system can efficiently examine the exactness of data which are shared. Here, it is unnecessary to retrieve whole data which are saved in cloud. To steer clear of the limitations in ordinary schemes, collusion-resistant proxy re-acknowledgement technique is proposed. This novel technique has the ability to bring about a re-stamping key along with repudiated user's common key and remaining user's personal key. User 'A' should do encryption over files using distinct public-keys, but should send 'B', only one (fixed-size) decryption key. Here, decryption key is essential, and it must be sent through the channel which has the characteristic of increased security and the decryption key must be hidden from view. Hence, small key size should be preferred. This novel Proxy re-signature mechanism comprises eight phases (Initialize, SecKey, UpKey, SignKey, ReKey, Sign, ReSign, Check):

- **Initialize**(Z) → (p*, mk): It takes security parameter Z, and generates public parameters p* and main secret key mk corresponding to the Private Key Generator.

- **SecKey**(p*, mk, ID) → s_{ID}: It takes p*, mk, user's identification ID and this process brings about a confidential (secret) key s_{ID} corresponding to the identification ID.

- **UpKey**(p*, mk, ID, ts) → $uk_{ID,tp}$: It takes p*, mk, an identification ID plus time span ts, this proposed process brings about an update key $uk_{ID,tp}$ with reference to identification ID plus time span ts.

- **SignKey**(p*, s_{ID}, $uk_{ID,ts}$) → $sk_{ID,tp}$: It takes p*, secret key s_{ID} and an update key $uk_{ID,ts}$. Suppose, when the identification ID has getting revoked in the course of time span ts, a flaw mark e is generated; otherwise, it generates a stamping key $sk_{ID,ts}$ on (ID, ts).

- **ReKey**(p*, $sk_{A,ts}$, $sk_{B,ts}$) → $r_{A\rightarrow B,t}$: It takes p* and two stamping keys ($sk_{A,ts}$, $sk_{B,ts}$) with reference to identifications (ID_A, ID_B) in the course time of time span ts, this algorithm brings about a re-acknowledging key $r_{A\rightarrow B,ts}$.

- **Sign**(p*, $sk_{ID,t}$, M) → SG: It takes p*, stamping key $sk_{ID,ts}$ and a message input M, this process brings about a stamp SG on M.

- **ReSign**(p*,$r_{A\rightarrow B}$, ID_A, ts, M, SG_A) → SG_B: Given p*, a re-acknowledging key $r_{A\rightarrow B}$ and stamp SG_A over message input M corresponding to identification ID_A plus time span ts, this algorithm generates e if **Check**(p*, ID_A, ts, M, SG_A) = NULL; Else, this process brings about a stamp SG_B over M corresponding to identity ID_B and time span ts.

- **Check**(p*, ID, t, M, SG) → {NULL, TRUE}: Given p*, an identification ID, time span ts, message input M and plus stamp SG, this process brings about TRUE if SG is a correct stamp of M on (ID, ts); else, it generates NULL.

SYSTEM PROTOTYPE

The system prototype shown in Fig.2 is explained in detail in this part. The four modules used in our proposed methodology are: Data sharing, User revocation, Third Party Verification and Proxy Re-signature.

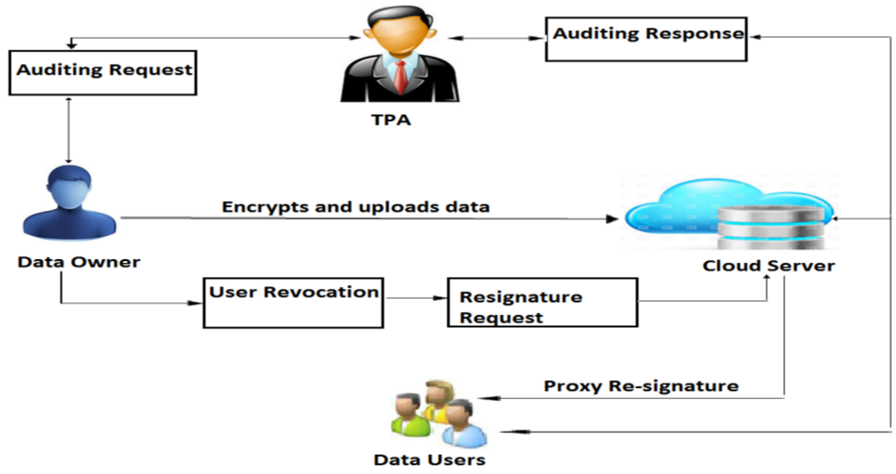


Fig.2 System Architecture

Data uploading and sharing in cloud

The cloud provides data repository and also data sharing facilities to its users. For protecting the perfection of data that are shared, every shared data chunks ought to be attached with a signature and these signatures has to be computed by one among the cloud users who resides in that group. The authentic user should compute each and every signature over shared data in cloud, as soon as the shared data firstly invented by the authentic user. Next to this process, while a particular user changes a chunk, the certain user must also stamp the changed data chunk using his/her own private key. Because of the data that the cloud possesses shared amid a batch of users, various data chunks might be stamped by various users as various alterations done by various users.

User Revocation

Suppose, a particular user goes away from group or when he gets noted for his misbehaviour inside group, then that group has to repudiate this user. Actually, as the Author for data which have been shared, who is the only authentic user and he functions as the group leader i.e. manager and can repudiate users instead of that batch. After repudiation of user’s rights, all signatures become invalid which were computed by that revoked user in that batch. The blocks that were before acknowledged by repudiated user ought to be re-stamped by the remaining user’s key which is private; therefore, the exactness of complete data is to be checked using common keys of remaining users.

Third Party Verification

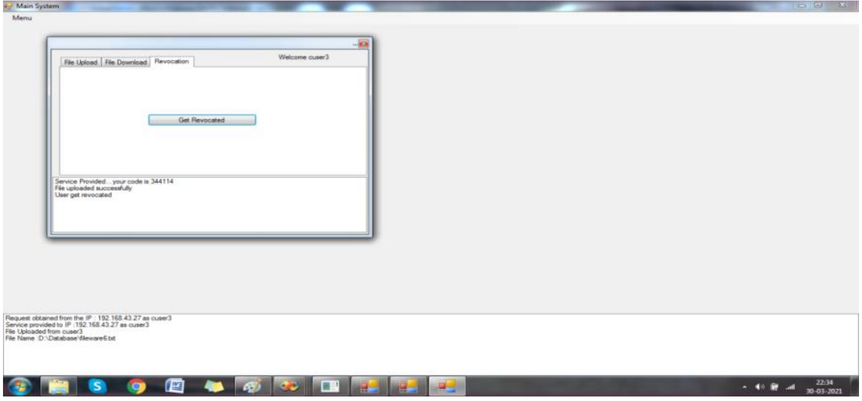
This Third Party Verification module verifies the perfection of data that are getting shared, along with the guarantee of efficacy in user repudiation process in cloud. Our newly proposed scheme, utilizes the new concept and that concept is based on collusion resistant proxy re-signatures, such that, while a particular user in a certain batch is repudiated, the cloud which functions as proxy, can re-acknowledge the chunks by means of a resigning key, which are already stamped by repudiated user. As an effect, efficacy of user repudiation has been heightened more; reckoning plus communication assets which belong to remaining users shall be saved easily. Moreover, cloud which not resides in identic trusty realm with every user, has the ability to convert a repudiated user’s signature into remaining user’s signature in the identic block.

Collusion Resistant Proxy Re-signature

Collusion Resistant Proxy re-signatures permit partially credible proxy to function like dragoman of stamps between the two users. For example, user ‘A’ and user ‘B’. More distinctly, the proxy actually is the cloud that has the ability to transform a stamp of A into a stamp of B over the identic block. In the mean time, the proxy cannot learn any secret keys which belong only to those two particular users; actually, it means that it has no ability to stamp any data chunk instead of either A or B. In this system, inorder to enhance efficacy of user repudiation, we have invented a new scheme which allows the functioning of cloud like proxy and it transforms signatures for users during the time span of user repudiation. Mainly, as the intrigue-renitent proxy re-stamp technique actually have two phases of signatures (That is, initial phase has been acknowledged by only the user; the upcoming phase has been re-acknowledged by proxy), and the two phases of acknowledgements are in diverse forms which ought to examined, thereby it achieves blockless verifiability on both levels of signatures and checking them together by third party verification.

EXPERIMENTAL RESULTS

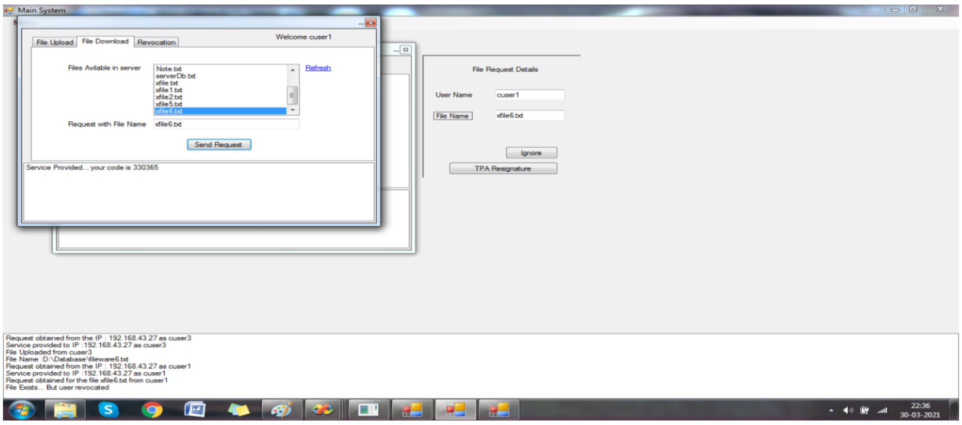
Our newly recommended technique is executed in a successful manner and running of this novel scheme, done using C#.NET in front end and SQL server in back end. The reliability pertaining to the data which are shared in cloud has been highly guaranteed. Also, the reckoning process overhead during the course of user repudiation has been considerably decreased. Fig.3 shows the revocation of user. Fig.4 depicts the appeal to get access for the revocated user’s file. The data correctness verifiability done by TPA has been screen shotted and shown in Fig.5. Finally, revocated user’s file is accessed successfully, after the re-signature process has been screen shotted and shown in Fig.6.



```
Service Provided... your code is 344114
File uploaded successfully
User get revocated
```

```
Request obtained from the IP : 192.168.43.27 as cuser3
Service provided to IP :192.168.43.27 as cuser3
File Uploaded from cuser3
File Name :D:\Database\fileware6.txt
```

Fig.3 User Revocation



Welcome cuser1

Files Avilable in server: Note.txt, serverDb.txt, xfile.txt, xfile1.txt, xfile2.txt, xfile5.txt, xfile6.txt

Request with File Name: xfile6.txt

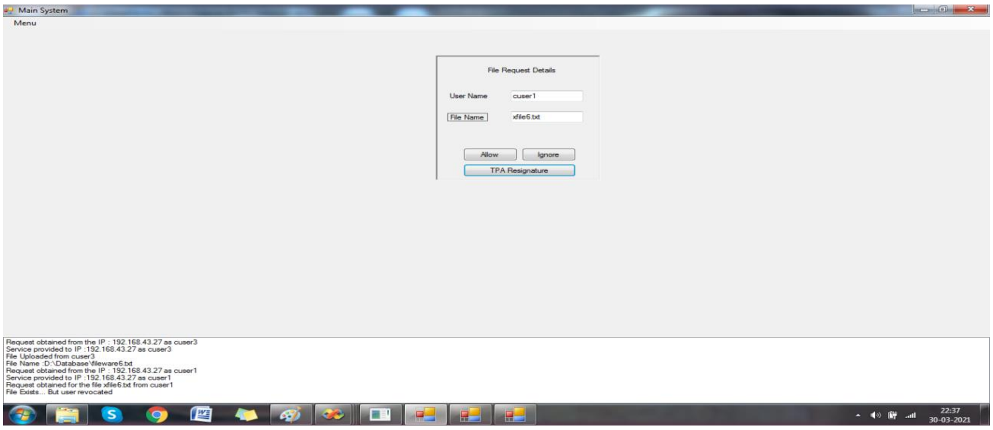
Service Provided... your code is 330365

User Name: cuser1

File Name: xfile6.txt

```
Request obtained from the IP : 192.168.43.27 as cuser3
Service provided to IP :192.168.43.27 as cuser3
File Uploaded from cuser3
File Name :D:\Database\fileware6.txt
Request obtained from the IP : 192.168.43.27 as cuser1
Service provided to IP :192.168.43.27 as cuser1
Request obtained for the file xfile6.txt from cuser1
File Exists... But user revocated
```

Fig.4 Request for revocated user's file

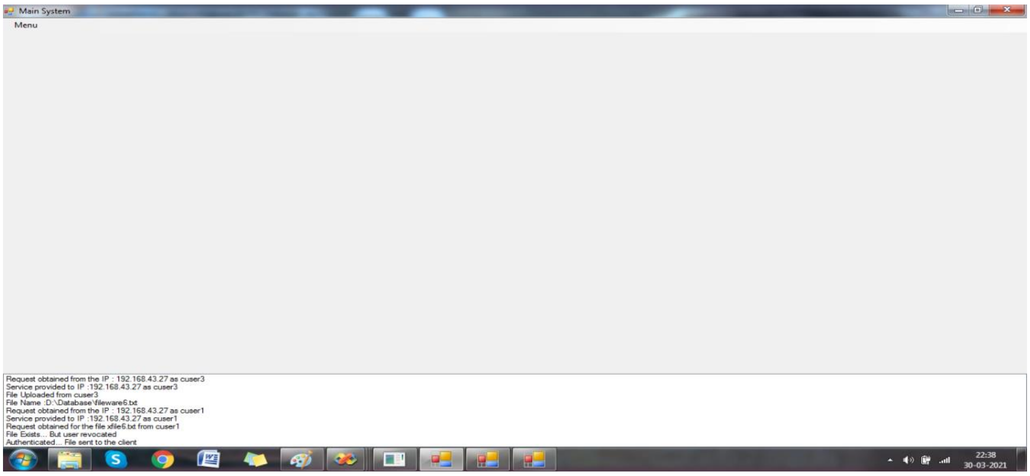


User Name: cuser1

File Name: xfile6.txt

```
Request obtained from the IP : 192.168.43.27 as cuser3
Service provided to IP :192.168.43.27 as cuser3
File Uploaded from cuser3
File Name :D:\Database\fileware6.txt
Request obtained from the IP : 192.168.43.27 as cuser1
Service provided to IP :192.168.43.27 as cuser1
Request obtained for the file xfile6.txt from cuser1
File Exists... But user revocated
```

Fig.5 TPA Re-signature



```
Request obtained from the IP : 192.168.43.27 as cuser3
Service provided to IP :192.168.43.27 as cuser3
File Uploaded from cuser3
File Name :D:\Database\fileware6.txt
Request obtained from the IP : 192.168.43.27 as cuser1
Service provided to IP :192.168.43.27 as cuser1
Request obtained for the file xfile6.txt from cuser1
File Exists... But user revocated
Authenticated... File sent to the client
```

Fig.6 Revocated user's File access

CONCLUSION

Thus the tedious operation of user repudiation can be made easier, also the cloud attains highest possible protection through newly introduced novel scheme of Collusion Resistant Proxy Re-acknowledgement technique. Our implementation results obviously prove that the cloud is now finely able to enhance the efficacy of user repudiation, and remaining users those who abides in the group has the excellent ability to save a noteworthy amount of reckoning plus communication assets during user repudiation. In future, it is planned to do classification or decision making through the training data which are kept in the secure distributed cloud environment.

REFERENCES

1) G. Ateniese et al. “Provable Data Possession at Untrusted Stores,” Proc. 14th ACM Conf. Computer and Comm. Security (CCS’07),pp. 598-610, 2007.
2) Cloud Security Alliance, “Security guidance for critical areas of focus in cloud computing,” 2009, http://www. cloudsecurityalliance.org.
3) Giuseppe Ateniese and Susan Hohenberger, “Proxy resignatures :New definations alogorithms and applications” November 28, 2005
4) S. R. Tate, R. Vishwanathan, and L. Everhart, “Multi-User Dynamic Proofs of Data Ownership Using Trusted Hardware,”Proc. Third ACM Conf. Data and Application Security and Privacy(CODASPY’13), pp. 353-364, 2013
5) A. Boldyreva, A. Palacio, and B. Warinschi. Secure proxy signature schemes for delegation of signing rights. Cryptology ePrint Archive, Report 2003/096. 2003.
6) M. Blaze et al. “Divertible Rules and Atomic Proxy Cryptography,” Proc. Int’l Conf. the Theory and Usage of cryptographic Techniques (EUROCRYPT’98), pp. 127-144, 1998.