IDENTITY PRESERVING IN DYNAMIC GROUPS FOR DATA SHARING AND AUDITING IN CLOUD

P.Raja Sekhar Reddy[1], A.Mallikarjuna Reddy[2], B.Ujwala[3]

[1]*Department of CSE, Anurag Group of Instiitutions, Hyderabad, India*
rajasekharreddycse@cvsr.ac.in
[2]*Department of CSE, Anurag Group of Instiitutions, Hyderabad, India*
mallikarjunreddycse@cvsr.ac.in
[3]*Department of CSE, Anurag Group of Instiitutions, Hyderabad, India*
ujwalacse@cvsr.ac.in

Abstract

Users of Cloud can store up their data and retrieve the required services or/and applications from a free pool of resources for computing. This method of compute provides an budgetable and suitable resolution for using resources among users of the cloud. It is one of the common places for data storage and shared across multiple users. Unfortunately, sharing of data in groups which are dynamic in nature while keeping data of the user and identity preservance from non trustable cloud server and general checking the correctness for such data while giving same priority to preserve the identity in dynamic group environment remains to be challenge. In the present proposed mechanism the signer identity is not disclosed to auditor to check the integrity of data that is distributed between different users without accessing the file in its entirety.

Keywords: *CloudServer, Auditor, Group Manager, Cloud users.*

----------***----------

I. INTRODUCTION

The companies offering the Cloud services deal with an excellent infrastructure that provides an elastic , protected and consistent atmosphere for users, at a nominal subsidiary cost due to environment which is distributed in nature. It has become custom for users in utilizing storage services of cloud to use data with others members in a group, as using and distributing of data and other resources has become a usual feature in the majority cloud storage contributions. The truthfulness of data in cloud, however, is question to doubt and check, as data in un-trusted cloud can effortlessly be ruined, due to human errors or/and hardware failures [1]. To guard the reliability of data, it is finest to carry out general auditing with the help of third party auditor (TPA), who render its services related to auditing with more potential computation and communication abilities than users can do on their own. The provable data possession (PDP) mechanism [2] to implement auditing is intended to test the accuracy of data resided in un-trusted server, by restricting to retrieve data in its entirety. Wang et al. [3] is considered to build an auditing method for data at cloud, thus during public auditing, the confidential data belonging to a individual is not opened to the auditor. We consider the distribution of data between several users is possibly the most attractive features that inspire the storage at cloud. An exceptional trouble comes into the picture during the process of auditing cloud data , how to safeguard identity from the Auditor, because the signer identities on data may specify that a user in the cluster or a particular building block in data is a privileged precious than others , in addition with this problem , how to administer the users dynamically in the group as the group may frequently changing the count as some users may be going out and some users may be joining the group, under this circumstances how to manage the groups dynamically without effecting the group signature in the resource trying to utilize by the group members. One of the most essential offerings from cloud service providers is storage of data. Lets take experimental data A corporation allows employees in the similar batch to hoard and distribute files or data in the cloud, the employees are not going to worry about storage and data preservation. However, it gives a threat to the privacy of stored files. Specially, the clouds monitored by providers are not trusted by users if the data stored is private and useful. To preserve privacy of data, a solution is transform data into cipher data and push the cipher text in the cloud [4].

These are some of the issues identified for sharing data in
1] Identity confidentiality is the most vital obstacles for the extensive use of cloud.
2] It is extremely suggested that , in a group a user should be able to completely feel free to store data and use sharing services provided. Each user in the group is to not only perform read operation on data , but also modify own data in the file using by all users of the group.
3] Groups usually dynamic in nature, change of membership makes secure and safe sharing of data is enormously hard. The contributions to solve the challenges, we suggest a protected sharing method for data in dynamic groups of the cloud followed by the public auditing process. A number of security solution for sharing on clouds of un trusted has focused in [5],[6],[7].In specified approaches ,owners of data will store the cipher text files in non-

trustable storage servers and share the decryption keys to certified group users. This makes un permitted users and servers cannot study the information of data files, since they don't know the information of decryption keys.

Major contributions of current paper includes:
1. A secure method for sharing data when environment is dynamic in nature, which means that user in the group can strongly distribute data with other group members in un-trusted cloud.
2. This Method supports groups efficiently in dynamic environment. specially, approved users who are new to the group can straightaway decrypt data files which are uploaded before their joining into the group. Revocation of users can be obtained in a technique of extended catalog of revocation not included by making any updation in the secret keys of the outstanding members.
3. Offering safe or confidentiality access and managing members, which confirms any user in a group to use the cloud resources.
4. Current work Suggested a model of Third party public auditing protocol for privacy preserving.

II. RELATED WORK

Kallahalla et, al. [5] introduced solution based on cryptographic that gives secure file storing in storage in un-trusted servers. This can be done in splitting up files into group of files and enciphering each file group using a distinct block key for each file, the owner can distribute the same information with others, which encrypts the keys of file-block. Though, it pays the penalty of about weighty key allocation burden of sharing file when they are in large-scale , as well the key related to file-block required to reorganize and disseminate for revocation of user.

Stored data/files on the un-trusted cloud contain two parts [6]: Metadata and data, The metadata imply the access manage data which includes a sequence of key blocks which are encrypted, each block gets encrypted using public key of users who are approved. Therefore amount of metadata is relative to figure of users who are authorized. The revocation of users is an willful matter particularly for extensive sharing, as the metadata needs to update. In the construction of NNL[10] for efficiently revoking the keys for key revocation procedure. On the other hand, if a new user adds in the group, each users secret key in an NNL based solution has to be recomputed, which becomes a drawback for the application for groups which are dynamic in nature. An additional problem is that the computation complexity of encryption gradually increase with sharing.

In the research paper [7] proposed substitute encryptions carrying for multiple times to protected scattered storage space. Specially, using symmetric and content related keys .

In the paper proposed by Yu et al [8] a data access control method which is scalable in cloud computing and it works on KPABE method. The holder of data makes use a arbitrary key to encipher the file, the arbitrary key is additionally enciphered with attributes using KP-ABE. The manager assign an access arrangement and the matching Skey(Secret) to approved users, so that a member will decrypt a cipher text To attain revocation, the manager delegate's jobs of data file are multi encryption and user Skey update to cloud servers. Though, the single owner mode holds back the apps to execute with the situation, which associate in a group has permitted to store or share files with others. In the paper [9] Lu et al proposed a secured method, which is build on group signatures. Mainly, the system is set with a one attribute. Every member obtain two keys after successful completion of registration process:
1] Group _signature key
2] Attribute_key.

Using the attribute based encryption a member can encrypt his data and other members can decrypt using their attribute keys.But Revocation of user is not supported in this method.

III. PRELIMINARIES

A. Bi-linear Map Theory
From [11] Let S1 and S2 be an additive cyclic group and a multiplicative cyclic set or group of the same prime order q, respectively.

Let e: S1 X S1 -> S2 denote a bilinear map designed with the following properties:
1. Every Bilinear map is Bilinear: For all a, b $\in Z_q^*$ and P, Q $\in$ S1, $e(aP, bQ)=e(P,Q)^{ab}$.
2. Every Bilinear map is Non degenerate: $\exists$P such that $e(P,P)\neq 1$.
3. Every Bilinear map is Computable: There is an efficient algorithm to compute e(P,Q) for any P,Q $\in$S1.

B. complication Assumptions
Def 1: q-strong Diffie-Hellman Assumption [11]
Def 2: Decision linear Assumption [11]
Def 3: Weak Bilinear Diffie-Hellman Exponent Assumption [12].
Def 4: ((t,n)-general Diffie-Hellman Exponent Assumption [13]).

C. Group Signature
This technique was first proposed by Chaum and van Heyst[15]. usually, this scheme permits any user can sign on messages while making the characteristics(identity) undisclosed to verifiers, an alternative to short group signature method used to implement unidentified access control, which also supports well-organized revocation of members[12].

D. Dynamic Broadcast Encryption
The technique proposed in [16] enable to broadcast data units which is encrypted to all members so that a confidential users gets the opportunity to decrypt the data. In addition to specified distinctiveness, dynamic broadcast encryption permits the manager to dynamically append members without disturbing the past computed data, like member decryption keys , the volume of cipher texts are not

changed and the group E_key does not require any modification. The implementation of this technique is proposed under a technique called bilinear pairing in [14], which is used as the foundation for distribution of files in groups which are dynamic.

E. MAC-based Solution

The two potential options to create and utilize MAC is to validate the data. An unimportant route is uploading the blocks of data along with MACs to the server, send the matching private key to Auditor. The Auditor can arbitrarily access blocks along with the MACs and perform test on accuracy with the private key. Besides the high computation and communication complexities, the auditor needs the data blocks information to do verification. The idea is as follows. The cloud user chooses s arbitrary message authentication code keys $\{sk\tau\}_{1\le\tau\le s,}$ and pre-computes MACs, $\{MAC_{sk\tau}(F)\}_{1\le\tau\le s}$ for the entire data file F before data outsourcing,, and publishes verification metadata to auditor. The auditor can disclose a private key sk_τ to the cloud server and demand for a fresh MAC for assessment in each audit. Privacy preserving is impossible to get back the file with given $MAC_{sk\tau}$ (F) and sk_τ.

Though, it has major drawbacks:

1) The Data can be audited for limited number of times because it depends on the number of private keys .If once all private keys are worn out, user has to access complete data for computing and publishing new MACs to auditor.

2) The auditor must preserve and inform conditions between audits.

3) It supports for data which is fixed in nature, and may not efficiently work with dynamic data .

HLA-based Solution:

To efficiently perform auditing without accessing the blocks of data , the HLA technique [14], [15], [16] can be used. HLAs meant for un-forgeable proof of metadata that validate the block reliability. Dissimilarity is that HLAs can be aggregated. It is always likely to calculate HLA aggregate which will authenticate a sequential combination of the data blocks. HLA system works as follow. The user authenticate each unit of $B = (p_1, \cdots, p_n)$ by a set of HLAs Φ. The cloud stores $\{B, \Phi\}$. The auditor verifies the storage by passing a arbitrary set of test $\{vi\}$. The cloud then returns $\mu = \sum_i v_i \cdot p_i$ and an aggregated authenticator σ that has to authenticate μ. The use these HLA techniques is unfit for need. It is due to the sequential combination of blocks, $\mu = \sum_i v_i \cdot p_i$, may disclose user data to auditor, and may violate the privacy preserving. specially, if a sufficient number of of the blocks are composed, the TPA can estimate the content of users by giving solutions to linear equations .

IV. SYSTEM MODEL

As illustrated in the figure the current scenario consists of four parties: the cloud server, Auditor (TPA), Group Manger and cloud users.

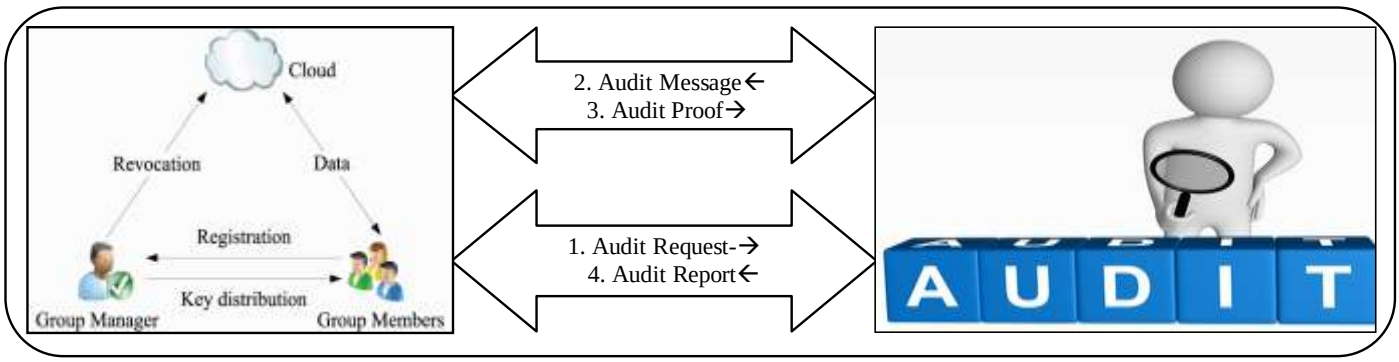

Fig.1. Audit process under dynamic groups for file sharing in cloud.

Cloud service providers operate the cloud and make available of huge storage services, even though the cloud platform is not trusted by users because the data is not being within the user environment. The group manager is responsible for user registration, user revocation.

The group of registered users will store their sensitive data into the cloud server which is shared by all the members of the group.

The auditor (TPA) is liable for auditing the data and checking its correctness, in this process if user wants to check the correctness of data that is being shared, he will send a request to the auditor, after receiving, auditor will generate the auditing message to the server and retrieves a proof of shared data from the cloud server, then the auditor verifies the proof, finally the auditor sends an audit statement to the user after verification.

V. THE PROPOSED METHOD

A. Overview

To execute the sharing of data in secured fashion under groups of dynamic in cloud ,we merged the broadcast encryption and group signature techniques which makes data users to safely share the data files with others. In the current scheme manager of group computes the parameters for revocation and made the result accessible by moving it to the cloud [19]

B. Method Description

- *User Registration*

For the user *i* registration with ID_i, identity , manager arbitrarily use a number $x_i \in Z_q^*$ then calculates C_i and D_i as the following equation.

$$C_i = 1/\gamma+x_i \,.P \in S_1$$
$$D_i = x_i /\gamma+x_i \,.\, G \in S_1 \quad (1)$$

next, the manager initiates his actions by adding (C_i,X_i,ID_i) in the user list of the group.

- *User Revocation*

Revocation of user is carried out by the manager with the help of a public revocation set , on which any user can encipher the data files and guarantees the privacy from revoked users.

Let ID_{group} indicate the identity of group. The record (C_i, x_i, t_i) the user *i* with the partial private key (C_i,x_i) is revoked at time t_i. $H_1,H_,\ldots, Hr$ and Z_r are estimated by manager with the private secret γ as follows:

$$H_1 = 1/\gamma+x_1 \,.P \in S_1$$
$$H_2 = 1/(\gamma+x_1) \,.\, (\gamma+x_1).P \in S_1$$
$$H_r = 1/(\gamma+x_1) \,.\, (\gamma+x_1)\ldots. (\gamma+x_r).P \in S_1$$
$$Z_r = 1/Z(\gamma+x_1) \,.\, (\gamma+x_1)\ldots. (\gamma+x_r). \in S_2 \quad (2)$$

Which is forced in provable reply method in [17], to ensure that to obtain the recent list of revocation from revocation set. The manager updates the revocation set regularly even though none was revoked from group .The revocation set is enclosed by a signature *sig(Revocation Set)* to state its validity. The generated signature is derived from BLS signature algorithm [18] by the manager i.e. ,

sig(Revocation set) = γf_1(Revocation set).

The manager sends the revocation set to the cloud for usage by the public.

- *File Generation*

The operations performed for file generation:

1. Get the revocation set , then the member sends the request ID_{group} to cloud. It reverts back with the revocation set to the requestor.
2. Check the legality of the revocation set which is received and ensure that the current timestamp is set or not and verify the signature *sig*(Revocation set) with $e(W,f_1(\text{Revocation set}))= e(P,sig(\text{Revocation set}))$. Result of this equation produces the invalid revocation set then owner of the data needs to stop this scheme.
3. File needs to be encrypted. This process is explained in following cases:

Case 1. None of users available in the revocation set [19].

Case 2. *R* number in the revocation set [19].

- *File Deletion*

Data owner or manager can delete the file. For file ID_{data} deletion the manager calculates a sig $\gamma f_1 (ID_{data})$ and transmits the sig with ID_{data} . The formulae $e(\gamma f_1 (ID_{data}),P)=e(W,f_1 (ID_{data}))$ satisfies the deletion of file is done by the cloud.

Procedure 1: Sig-Gen

Input: Private key (A,*x*), system parameter (P,U,V ,H,W) and data M.

Output: create a group sign on M data file.

Begin

choose arbitrary numbers $\alpha,\beta,r_\alpha,k_\beta,k_x,k_{\delta1}k_{\delta2} \in Z_q^*$

Set $\delta_1=x\,\alpha$,$\delta_2= x\,\beta$

Calculate the values

$J_1 = \alpha.U$

$J_2= \beta.V$

$J_{3=} A_i + (\alpha+\beta).H$

$K_1 = k_\alpha \,.\, U$

$K_2 = k_\beta \,.\, V$

$K_3 =e(T_3, P)^{kx} e(H,W)^{-} k_\alpha - k_\beta \; e(H,P)^{-} k_{\delta1} . k_{\delta2}$

$K_4 = k_x \,.\, J_1 - k_{\delta1.} U$

$K_{5=} r_x \,.\, J_2 - k_{\delta2.} V$

put c =f(M,J_1,J_2,J_3,K_1,K_2,K_3,K_4,K_5)

Build the numbers

$L_\alpha = k_\alpha + E\alpha$

$L_\beta = k_\beta + E\beta$

$L_{x=} k_x + Ex$

$L_{\delta1}= k_{\delta1} + E\delta_1$

$L_{\delta2=} k_{\delta2} + E\delta_2$

Return $\sigma=(J_1,J_2,J_3,c, L_\alpha ,L_\beta ,L_x,L_{\delta1},L_{\delta2})$

End

Procedure (2). Sig-Veri

Input: constraints (P,U,V,H,W),M and a

Sig $\sigma =(J_1,J_2,J_3,c, L_\alpha ,L_\beta ,L_x,L_{\delta1},L_{\delta2})$

Begin

Calculate the following values

$K_{11}= L_\alpha \,.U - c.\, J1$

$K_{12} = L_\beta \,.V - c \,.J2$

$K_{13} = e(J3,W)/e(P,P)^c \, e(J_3,P)^{Lx} \, e(H,W)^{-L\alpha - L\beta} \,.e(H,P)^{-L\delta1 - L\delta2}$

$K_{14} = L_x . J1 - L_{\delta1} \,.U$

$K_{15} = L_x - J2 - L_{\delta2} \,.\, V$

if c = f(M,J1,J2,J3,K11,K_{12},K_{13},K_{14},K_{15})

Return 1

else

Return 0

End

Output: 1 or 0

Procedure (3). Revoc - Veri

Input: constraints (Q_0,Q_1,Q_2), signature of group σ, and group of keys(revocation keys) $w_1,\ldots w_r$

Output: Valid or Invalid

Begin

Set temp=$e(J_1,Q_1)e(J_2,Q_2)$

while I =1 .. n

If $e(J_3-W_i, Q_0)$=con

Return valid revo-set

End if

End while

Return Invalid revo-set

End

The owner of the data do the following actions for deleting a file :
Get the record (ID_{data} ,*T*), and use algorithm 1 to compute a signature on [group(ID_{data}, *T*)] and transmits the same as request for deleting of file in the cloud.

On receiving request, server calls procedures 2 and 3 to check the signature of group .Only after group signature verification, the cloud deletes the file.

File Access
The user do the following actions for accessing the file :
1. The user gets the private key(A,X) for calculating a signature σ_u on the message(ID_{group},ID_{data},t) by means of procedure1, where t denote the system time, and the ID_{data} can be retained thru the manager maintained in file of local machine .The member transmits a request containing (ID_{group},ID_{data},t,σ_u) to the server. On receipt of request the cloud server applies procedure 2 to verify the signature and do verification of revocation with procedure 3.
After verification, the server reverts back with the data and the revocation set to the member.
2. Inspect the revocation set to know its validity. This is same as to the step 2 in file generation phase.
3. Verify the file validity and decrypt it.

- *public auditing system for preserving privacy*

If the group user needs to check accuracy of stored and valuable data in server, the public audit mechanism can be effectively implemented to achieve this task. Various public auditing protocols were implemented to do these tasks. To achieve privacy preserving we suggest the model which uses homo-morphic linear authenticator. This scheme follow the implementation process with the
Init- Phase: The consumer of cloud runs KeyGeneration to create the secret and public arguments. Specially, the consumer selects an arbitrary signing pair of keys (*cpk, csk*), a arbitrary $y \leftarrow Z_p$, an arbitrary element $v \leftarrow G_1$, and calculates $u \leftarrow g^y$. The secret parameter is *sk* = y, *csk*) and the public parameters are *pk* = (*cpk, u, g, v, e(v, u)*).
A file $F = (b_1, \ldots, b_n)$, the user run Signature generation to calculate authenticator Ω_I for every block b_i : $\Omega_i \leftarrow (H(N_i) \cdot v\, b_i)^y \in G1$. Here N_i = *name of the file* is selected by the user arbitrary from Z_p. represent the group of authenticators by $\theta = \{\Omega i\}_{1 \le i \le n}$. The last part of the signature generation is to ensure the correctness of *name*-identifier .
To ensure correctness of storage, Specially, the cloud selects an arbitrary element $q \leftarrow Z_p$, and computes $R = e(v, u)^r \in G_T$ Let δ represents the sequential blocks specified in challenge $\delta = \sum_{i \in I} v_i\, b_i$. To blind δ with r, the server calculates: $\delta = r + \gamma\delta \bmod p$, where $\gamma = h(R) \in Z_p$. In meantime, it computes an authenticator $= \pi_{i \in I}\, \sigma_i^{vi} \in G_1$. It then transmits { θ, Ω, R} as the result for correctness of storage to the auditor.

Table 1: Public Auditing process

TPA		Cloud Server
1. Access file tag *e*, check Sign, and terminates if fail;		
2. Create an arbitrary test $v_i\, b_i$, and also $\Omega = \pi_{i \in I}\, \Omega_i^{vi}$	$\{(i,v_i)\}i \in I$ → Test request	
		3. Compute $\theta = \sum_{i \in I}$
		4. Arbitrarily pick x ← Zp, and hash(R);
		5. Calculate $\delta = x + \gamma\, \delta \bmod p$;
	{ δ, Ω ,R} Storage ← correctness	
6. Calculate γ = hash(R), and then test { δ, Ω ,R}		

Audit Phase: The auditor access the tag *e of* file. *As* we described in the Init- phase, the auditor checks the sig $SSig_{csk}$ *with cpk*, and terminates the process by obtaining result as false. Otherwise auditor recovers *name*. To create the test message for audit , auditor picks an arbitrary subset item $I = \{c_1, \ldots, c_c\}$ of set [1, *n*]. Every item $i \in I$, the auditor selects a arbitrary value u_i of length that is smaller than |p| [16]). The test message identifies the block positions that are needed to check. The auditor sends test = $\{(i, u_i)\}_{i \in I}$ to server. On receipt of test, the cloud runs Generation Proof to produce a result and fresh users can decrypt files directly which are stored before their actual participation. By using homo-morphic linear authenticator , auditor will study no information about the valuable data available in the cloud while audit process is in progress.

VI. TEST ANALYSIS

To evaluate the response rate of the cloud, testing its cost of computation to respond different operations issued by client requests which includes generation of files, access to files , and deletion of files . As it is assumption based approach and the burden on the cloud server in performing these operations are negligible

VII. CONCLUSION

In the current work, an extended secure and suitable data sharing method, for non static groups is introduced. The members can use the data and share to rest of the members of the group without disclosing his characteristics(identity) to the cloud. This works for existing user revocation and fresh user joining. The revocation of existing users can be done through a revocation set without making any changes in the keys information of the existing users. All the consumers can decipher the stored files in the cloud. If group user's wishes to check correctness of the data the audit, can be effectively implemented to achieve this task. The proposed method is suitable for dynamic groups in file sharing without revealing his identity and can able to implement public auditing.

ACKNOWLEDGEMENTS

This work was carried out with motivation of Dr.V.Vijaya kumar, Professor and Dean, CSE, AGI and with the help of reference papers , I thank our professor and authors of reference papers without which this work would not be carried out.

REFERENCES

[1] 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," *Communications of the ACM*, vol. 53, no. 4, pp. 50–58, Apirl 2010.

[2] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson,and D. Song, "Provable Data Possession at Untrusted Stores," in *Proc. ACM Conference on Computer and Communications Security (CCS)*, 2007, pp. 598–610.

[3] C. Wang, Q. Wang, K. Ren, and W. Lou, "Privacy-Preserving Public Auditing for Data Storage Security in Cloud Computing," in *Proc. IEEE International Conference on Computer Communications (INFOCOM)*, 2010, pp. 525–533.

[4] S. Kamara and K. Lauter, "Cryptographic Cloud Storage," Proc. Int'l Conf. Financial Cryptography and Data Security (FC), pp. 136-149, Jan. 2010.

[5] M. Kallahalla, E. Riedel, R. Swaminathan, Q. Wang, and K. Fu,"Plutus: Scalable Secure File Sharing on Untrusted Storage," Proc. USENIX Conf. File and Storage Technologies, pp. 29-42, 2003.

[6] E. Goh, H. Shacham, N. Modadugu, and D. Boneh, "Sirius: Securing Remote Untrusted Storage," Proc. Network and Distributed Systems Security Symp. (NDSS), pp. 131-145, 2003.

[7] G. Ateniese, K. Fu, M. Green, and S. Hohenberger, "Improved Proxy Re-Encryption Schemes with Applications to Secure Distributed Storage," Proc. Network and Distributed Systems Security Symp. (NDSS), pp. 29-43, 2005.

[8] S. Yu, C. Wang, K. Ren, and W. Lou, "Achieving Secure, Scalable, and Fine-Grained Data Access Control in Cloud Computing,"Proc. IEEE INFOCOM, pp. 534-542, 2010.

[9] R. Lu, X. Lin, X. Liang, and X. Shen, "Secure Provenance: The Essential of Bread and Butter of Data Forensics in Cloud Computing," Proc. ACM Symp. Information, Computer and Comm. Security, pp. 282-292, 2010.

[10] D. Naor, M. Naor, and J.B. Lotspiech, "Revocation and Tracing Schemes for Stateless Receivers," Proc. Ann. Int'l Cryptology Conf.
Advances in Cryptology (CRYPTO), pp. 41-62, 2001.

[11] D. Boneh, X. Boyen, and H. Shacham, "Short Group Signature," Proc. Int'l Cryptology Conf. Advances in Cryptology (CRYPTO),pp. 41-55, 2004.

[12] D. Boneh, X. Boyen, and E. Goh, "Hierarchical Identity BasedEncryption with Constant Size Ciphertext," Proc. Ann. Int'l Conf.Theory and Applications of Cryptographic Techniques (EUROCRYPT),pp. 440-456, 2005.

[13] C. Delerablee, P. Paillier, and D. Pointcheval, "Fully Collusion Secure Dynamic Broadcast Encryption with Constant-Size Ciphertexts or Decryption Keys," Proc. First Int'l Conf. Pairing-BasedCryptography, pp. 39-59, 2007.

[14] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song, "Provable data possession at untrustedstores," in *Proc. of CCS'07*, Alexandria, VA, October
2007, pp. 598–609.

[15] Q. Wang, C. Wang, J. Li, K. Ren, and W. Lou, "Enabling public Verifiability and data dynamics for storage security in cloud computing," in *Proc. of ESORICS'09, volume 5789 of LNCS*.Springer-Verlag, Sep. 2009, pp. 355–370.

[16] H. Shacham and B. Waters, "Compact proofs of retrievability," in *Proc. of Asiacrypt 2008*, vol. 5350, Dec 2008, pp. 90–107.

[17] B. Sheng and Q. Li, "Verifiable Privacy-Preserving Range Query in Two-Tiered Sensor Networks," Proc. IEEE INFOCOM, pp. 46-50, 2008.

[18] D. Boneh, B. Lynn, and H. Shacham, "Short Signature from the Weil Pairing," Proc. Int'l Conf. Theory and Application of Cryptology and Information Security: Advances in Cryptology, pp. 514-532, 2001.

[19] Xuefeng Liu, Yuqing Zhang, , Boyang Wang, and Jingbo Yan "Mona: Secure Multi-Owner Data Sharingfor Dynamic Groups in the Cloud" IEEE Transactions On Parallel And Distributed Systems, Vol. 24, No. 6, June 2013,Pp 1182-1191.