International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2013): 6.14 | Impact Factor (2013): 4.438

An Efficient Privacy Preserving Scheme over Encrypted Data in Cloud

Kirti Panmand[1], **Sandeep Kadam**[2]

[1, 2]Department of Computer Engineering, Dr. D. Y. Patil College of Engineering, Pune, Maharashtra, India

Abstract: ***Cloud computing is developing and deliberated next generation architecture for computing. Cloud computing is a combination of computing resources accessible via internet. Historically the client or organizations store data in data centers with firewall and other security methods used to defend data against intrudes to access the data. Since the data was confined to data centers in limits of organization, the control over the data was more and well defined measures could be taken for retrieving its own data. On the other hand in cloud computing, the data is warehoused anywhere across the globe, the client organization has minimum control over the stored data. To form the conviction for the development of cloud computing the cloud providers must defend the user data from unauthorized access and disclosure. Encryption technique could be used on the data on client side before storing it in cloud storage, but this technique has too much affliction from client side in terms of key management, maintenance etc. Divide and rule can be other techniques; it means distributing the task among various cloud services providers can profit the client. A TPA (Third Party Auditor) is used to provide security services, while the other cloud provider would be data storage provider. TPA would not store any data at its end, and it's only concerned for providing security service. The application will provide data integrity verification by using hashing algorithm like SHA-1, encryption/decryption will be done by using MD5 algorithm, and defining group of people who can access the shared data can be reached by describing access list. The application is liable for encryption/decryption, computing the hash data and does not store any data in TPA system. The encrypted data and original data hash are stored in Separate Cloud. Therefore even if the cloud system administrator has access user data, the data is in encrypted form, hence it will be tough for the system administrator to recognize the encrypted data. When the user downloads the data from Storage Cloud, it is decrypted first and then new hash is calculated which is then equated with hash of original data stored in Security Cloud. This application provides the user with the ability to store the encrypted data in the cloud and hash and encryption/decryption keys in security cloud service, and no single cloud service provider has access to both. Further benefit of assigning responsibility to TPA is that it aids the client from any kind of key management or maintenance of any important information related to data, because of which it allows the client to use any browser enabled devices to access such service.***

Keywords: Cloud computing, encryption and decryption service, data protection and integrity.

1. Introduction

In recent years, cloud computing has become very popular in the global industry. The initiatives include Google's research project for building an infrastructure to support research needs of top-tier American universities. Weiss noted that cloud computing services include several existing computing technologies, such as service-oriented utility computing, grid computing with large amount of computing resources, and that using data centers for data storage services [1].

A Simple mechanism for system to defend user data is that user data is encrypted before it is stored. In a cloud computing environment, a user's data can also be stored ensuing additional encryption, but if the storage and encryption of a given user's data is performed by the same service provider, the service provider's internal staff (e.g., system administrators and authorized staff) can use their decryption keys and internal access constitutional rights to access user data. This could put his data at risk of illegal exposer, from the user's point of view [2]. The cloud computing system responsible for encrypting user data has ability over all encryption keys required for data encryption but the condition is; the encryption provider does not store the user's data, i.e. TPA, a unique feature of the model is that different services are provided by multiple operators. There are some cryptographic schemes like anonymous authentication schemes, group signatures, zero knowledge protocols that can hide both user identity and provide authentication. The cloud service providers need to control the authentication process to permit the access of only valid user's to their services and they must be able to invalidate malicious clients and disclose their identities. Thousands of users can access cloud services at the same time. Henceforth, the authentication process of user access must be as efficient as possible and the computational overhead must be minimal.

In the cloud system, a protocol should reach the following requirements.

Authentication: Only a valid/authorized user can access its own data fields, only authorized data can be recognized by the authorized user, and any bogus data fields cannot mislead the legal user. 2) **Data anonymity:** An unauthorized person cannot recognize the exchanged data and communication state even it intercepts the exchanged messages via an open channel. 3) **User privacy:** An unauthorized person cannot know or guess a user's access rights, which represent user's interest in another user's authorized data fields. If and only if the both users have common interests in each other's authorized data fields, the cloud server will inform the two users to get the access permission sharing. Researchers have been worked to support security and privacy conservation in cloud applications, and there are a variety of cryptographic algorithms to deal with security and privacy problems, including security architectures [4], [5], data possession protocols [6], [7], data public auditing protocols [9], [10], [10], data sharing protocols [12], [13], [14], access control mechanisms [2], privacy preserving protocols [1], [2], [11], and key management [4]. However, most of the researches focus on the authentication to understand that only a valid user can access authorized data. When a user challenges the cloud server to request other users for data sharing, the access request itself may reveal the user's privacy no matter whether or not it can obtain the data access permissions.

Compared to the common audit, the audit services for cloud storages should provide clients with a more effective proof for authenticating the reliability of stored data. Unfortunately, the traditional cryptographic systems, based on hash functions and signature schemes, cannot support for data integrity verification without a local copy of data. In addition, it is unrealistic for audit services to download the whole data for inspecting data validation due to the communication cost, especially for large-size files. Therefore, following security and performance objectives should be considered to accomplish an efficient audit for outsourced storage in clouds: **Public auditability:** To allow TPA or clients with the help of TPA to verify the correctness of cloud data on demand without retrieving a copy of the whole data. **Dynamic operations:** To ensure there is no attack to compromise the security of verification protocol by using dynamic data operations.

2. Related Work

Boyang Wang, BaochunLi[1] has proposeda novel privacy-preserving mechanism that supports public auditing on shared data stored in the cloud. In this scheme, ring signature is used to compute verification metadata needed to audit the correctness of shared data. With this scheme, the identity of the signer on each block in shared data is kept secret from public verifiers, who are able to proficiently verify shared data integrity without retrieving the entire file. It also supports batch auditing i.e. auditor is able to perform multiple auditing tasks simultaneously instead of verifying them one by one. Mohamed Nabeel and Elisa Bertino [2] in their subsequent work proposed Access Control Policies (ACPs) with the help of group key management. A challenging issue in the TLE(Two Layer Encryption) approach is how to decompose the ACPs so that fine-grained ABAC (attribute-based access control) enforcement can be delegated to the cloud while at the same time the privacy of the identity attributes of the users and confidentiality of the data are assured. Each ACP should be decayed to two sub ACPs such that the combination of the two sub ACPs result in the original ACP. The two layer encryption should be performed such that the data owner first encrypts the data based on one set of sub ACPs and the cloud re-encrypts the encrypted data using the other set of ACPs. In [4] cheng, Tzeng consider a public-key cryptosystem mechanism that produces constant-size ciphertexts such that efficient allocations of decryption rights for any set of ciphertexts are possible. The innovation is that one can combined any set of secret keys and make them as compact as a single key, but surrounding the power of all the keys being aggregated.
Yong, Jianbing, Ho [5]introduceMerkle Hash Tree (MHT), to validate the signatures such that the signatures protect the reliability of file blocks while the authenticated data structure ensures the integrity and security of signatures. Barsoum [12] proposed a scheme which performs block-level dynamic operations on the outsourced data. It provides authorized access to the data and grant or revokes access to the outsourced data. Lin [13] introduces a threshold proxy re encryption scheme which integrate with a decentralized removal code such that a secure distributed storage system is formulated. This system not only supports secure data storage and retrieval, but also lets a user forward his data in the storage servers to another user without retrieving the data back.

3. System Model

A cloud data storage system consist of three different parts.: **User (U):** who has large amount of data files to be stored in the cloud; **cloud server (CS):** It is managed by the cloud service provider(CSP) to provide data storage facility and has large storage space and computation resources; **(TPA):** It has some capabilities that cloud users do not have and is trusted to assess the cloud storage service reliability on behalf of the user upon request. Users rely on the CS for cloud data storage and maintenance. Users can also interact with the CS to access and update their stored data for various application purposes. To save the computation resource as well as the online burden, cloud users may resort to TPA for ensuring the storage integrity of their outsourced data.

Sometimes CS might behave unfaithfully for his own benefit. CS might neglect to keep or delete rarely accessed data files which belong to regular cloud users. Likewise, the CS may decide to hide the data corruptions caused by server hacks or complicated failures to maintain reputation. It is assumed that TPA, who is in the business of auditing, is consistent and independent, and thus has no motivation to conspire with either the CS or the users during the auditing process. However, it troubles the user if the TPA could learn the outsourced data after the audit.

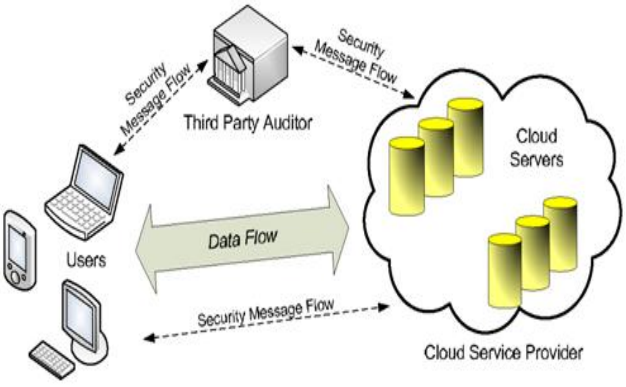


Figure 1: System Model

3.1 Preparatory

3.1.1 Batch Auditing

The goal of Cloud Audit is to provide cloud service providers with a way to make their performance and security data readily available for potential customers. The specification provides a standard way to present and share detailed, automated statistics about performance and security.

With the convention of auditing in the cloud, the TPA may get large no of auditing requests from different users in a short period of time. Unfortunately, permitting the TPA to validate the integrity of shared data for these users in some distinct auditing tasks would be very ineffective. Therefore, with the help of bilinear mapping, batch auditing mechanism has been introduced, which can increase the effectiveness of authentication on multiple auditing tasks. Batch auditing will divide the task among different TPA who works independently, it will save a lot of time and performance will improve.

Assume that there are B auditing tasks need to be worked. To efficiently audit these public data for different users in a single auditing task, the TPA sends an auditing message to the cloud server. After receiving the auditing message, the cloud server generates an auditing proof for each shared data. After the calculation, CS (Cloud Server) sends all the auditing proofs to the TPA. Then, the TPA verifies the accuracy of these proofs. After successful verification, TPA believes that the integrity of all the data is correct. Otherwise, there is some shared data, which is corrupted.

3.1.2 Dynamic Operations

This application also supports Dynamic Operation on the shared data, which allow each user in the group to easily modify cloud data and share the up-to-date version of data with the remaining group. A dynamic operation includes an insert, delete or update operation on a single block of data. Though, the calculation of a ring signature comprises an identifier of a block, which uses the index of a block as its identifier, is not appropriate for supporting dynamic operations on shared data. The reason behind this is, when a user modifies a single block in shared data by performing delete or update operation, the indices of blocks that after the modified block are transformed. Drawback of this change is, users require re-computing the signatures of these blocks, even if the content of these blocks are not modified.

3.1.3 Ring Signature

In cryptography, a ring signature is a type of digital signature that can be performed by any member of a group of users that each have keys. Therefore, a message signed with a ring signature is authorized by someone in a specific group of people. One of the security properties of a ring signature is that it should be computationally infeasible to determine which of the group members' keys was used to produce the signature. Ring signatures are similar to group sign mechanism but differ in two key ways: first, there is no way to revoke the anonymity of an individual signature, and second, any group of users can be used as a group without additional setup.

A ring signature where the group is the sender and the recipient of a message will only seem to be a signature of the sender to the recipient: anyone else will be unsure whether the recipient or the sender was the actual signer. Thus, such a signature is convincing, but cannot be transferred beyond its intended recipient.

A ring signature scheme is set-up free: The signer does not require the knowledge, permission, or help of the other ring members to place them in the ring; he only requires knowledge of their public keys. Different members can use different public key signature schemes, with different key and signature sizes. Authentication must fulfill the reliability, but along with that we want the signatures to be signer-ambiguous, so that that a signature cannot leak information about the identity of the signer.

4. System Implementation

4.1 Data Upload Scenario

1. The end user login to the system with his/her username & password.
2. Once the user is authenticated, the Deffie Hellman key is exchanged for the session.
3. Now a user can select the files which he/she wants to upload it to storage cloud
4. The user can also select is he/she wants to share the file with specific users.
5. The hash of the data in _le is calculated, using SHA-1 (original hash).
6. The data in file is now encrypted using DH keys.
7. The complete encrypted file and original hash of file data are now transferred to Security Cloud.

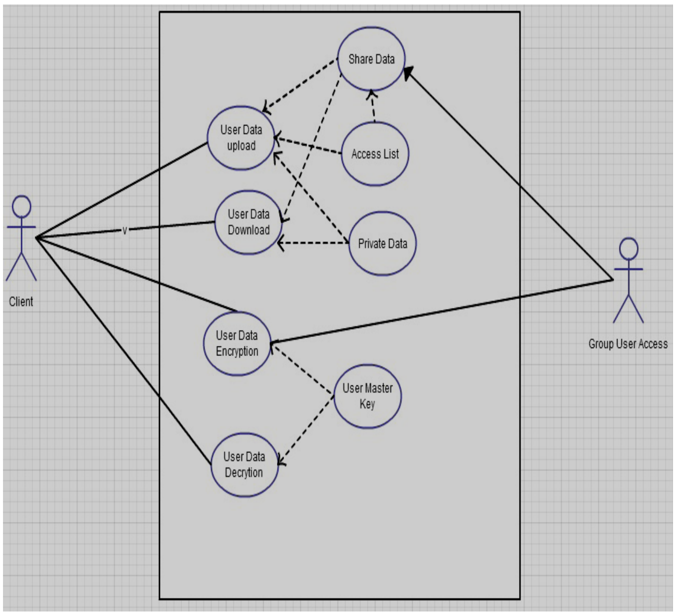


Figure 4.1: Use Case

8. At Security Cloud, encrypted file is decrypted back using DH key, while the hash is sorted in security cloud database.
9. The decrypted file is now encrypted with Symmetric Algorithm namely AES, using the Master Key generated for each user during user creation.
10. File ID, original hash (file/data hash), master key for each user is stored in Security Cloud database.
11. The Security Cloud now discards any contents of the files from its system, and does not store any file contents in its system.
12. The Encrypted file is sent back to user, to be uploaded to Storage Cloud.

13. The user now can upload the encrypted file to Storage Cloud.

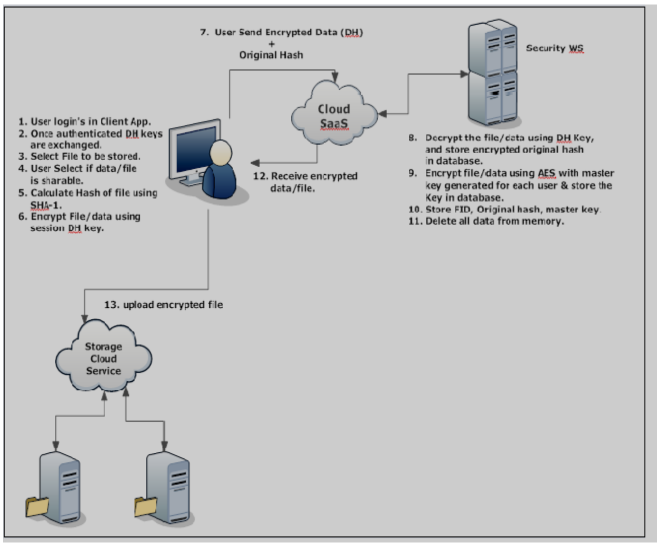

7. User Send Encrypted Data (DH)
+
Original Hash
Security WS
Cloud SaaS
1. User login's in Client App.
2. Once authenticated DH keys are exchanged.
3. Select File to be stored.
4. User Select if data/file is sharable.
5. Calculate Hash of file using SHA-1.
6. Encrypt File/data using session DH key.
12. Receive encrypted data/file.
8. Decrypt the file/data using DH Key, and store encrypted original hash in database.
9. Encrypt file/data using AES with master key generated for each user & store the Key in database.
10. Store FID, Original hash, master key.
11. Delete all data from memory.
13. upload encrypted file
Storage Cloud Service

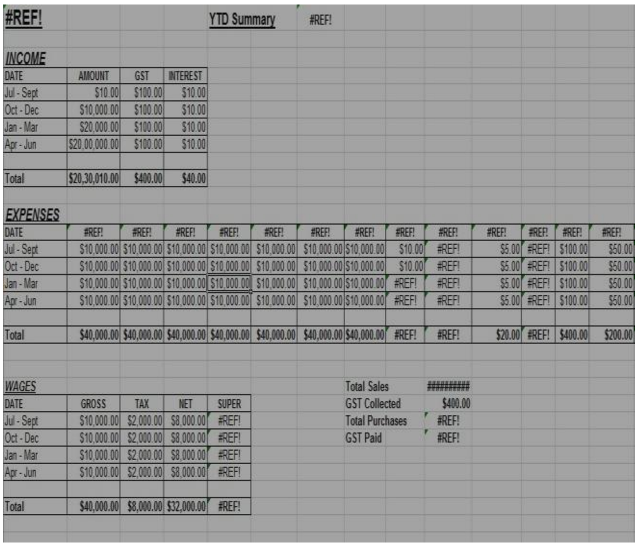
Figure 4.2: User Data Download Scenario

4.2 User Data Download Scenario

1) The end user login to the system with his/her username & password.
2) Once the user is authenticated, the Deffie Hellman key is exchanged for the session.
3) Now a user can select the files which he/she wants to download it from storage cloud.
4) The encrypted file is now downloaded from storage cloud to users match in.
5) The complete encrypted file is now transferred to Security Cloud.
6) The data in file is now encrypted using DH keys.
7) The complete encrypted file and original hash of file data are now transferred to Security Cloud.
8) At Security Cloud, decrypted files with Symmetric Algorithm namely AES using Master Key stored in security cloud database for each user.
9) The decrypted file is now encrypted with DH key.
10) The DH encrypted file and hash of the corresponding file is now passed to the users.
11) At user end, on receiving the encrypted file, it is decrypted with DH keys.
12) The hash of decrypted file is calculated using SHA-1 and original hash are now compared to see if they match, and accordingly appropriate message like, File tampered or File is intact are flashed on user screen. Thus the integrity of the data is verified.

5. Result Set

Let's take the example of "Yearly Tax Deduction" Excel File which we provide as an input to the system.



#REF! YTD Summary #REF!

INCOME

DATE	AMOUNT	GST	INTEREST
Jul - Sept	$10.00	$100.00	$10.00
Oct - Dec	$10,000.00	$100.00	$10.00
Jan - Mar	$20,000.00	$100.00	$10.00
Apr - Jun	$20,00,000.00	$100.00	$10.00
Total	$20,30,010.00	$400.00	$40.00

EXPENSES

DATE	#REF!	#REF!	#REF!	#REF!	#REF!	#REF!	#REF!	#REF!	#REF!	#REF!	#REF!	#REF!	#REF!
Jul - Sept	$10,000.00	$10,000.00	$10,000.00	$10,000.00	$10,000.00	$10,000.00	$10,000.00	$10.00	#REF!	$5.00	#REF!	$100.00	$50.00
Oct - Dec	$10,000.00	$10,000.00	$10,000.00	$10,000.00	$10,000.00	$10,000.00	$10,000.00	$10.00	#REF!	$5.00	#REF!	$100.00	$50.00
Jan - Mar	$10,000.00	$10,000.00	$10,000.00	$10,000.00	$10,000.00	$10,000.00	$10,000.00	#REF!	#REF!	$5.00	#REF!	$100.00	$50.00
Apr - Jun	$10,000.00	$10,000.00	$10,000.00	$10,000.00	$10,000.00	$10,000.00	$10,000.00	#REF!	#REF!	$5.00	#REF!	$100.00	$50.00
Total	$40,000.00	$40,000.00	$40,000.00	$40,000.00	$40,000.00	$40,000.00	$40,000.00	#REF!	#REF!	$20.00	#REF!	$400.00	$200.00

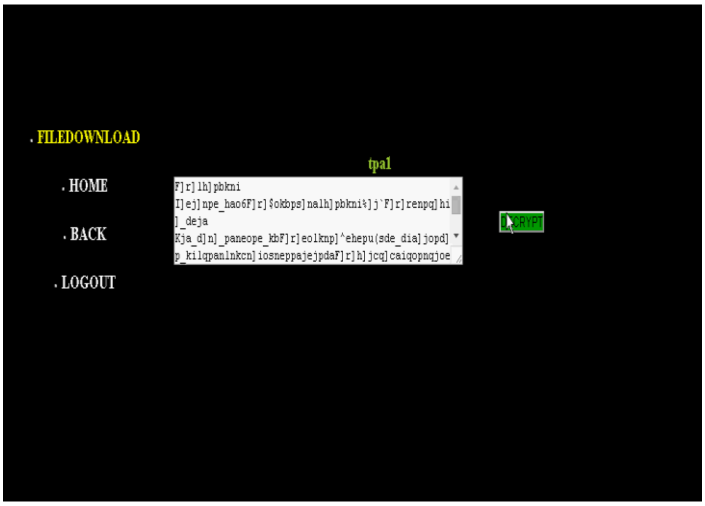
WAGES

DATE	GROSS	TAX	NET	SUPER
Jul - Sept	$10,000.00	$2,000.00	$8,000.00	#REF!
Oct - Dec	$10,000.00	$2,000.00	$8,000.00	#REF!
Jan - Mar	$10,000.00	$2,000.00	$8,000.00	#REF!
Apr - Jun	$10,000.00	$2,000.00	$8,000.00	#REF!
Total	$40,000.00	$8,000.00	$32,000.00	#REF!

Total Sales	#########
GST Collected	$400.00
Total Purchases	#REF!
GST Paid	#REF!

Figure 5.1: Sample Input

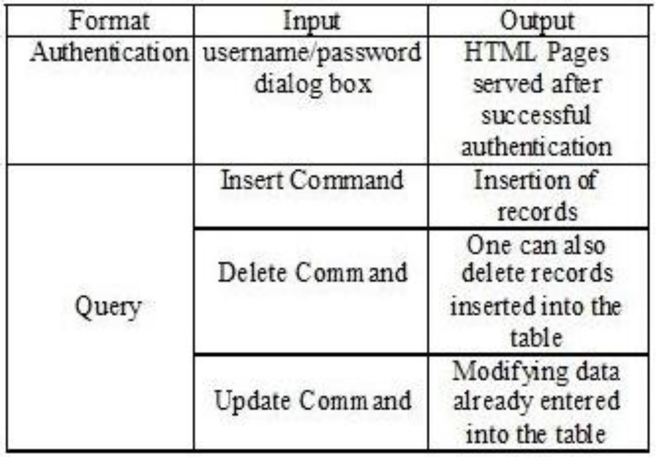
1. The file is exactly stored in encrypted format. The "userdata" in storage cloud is the table which stores the data for each user; Figure 8.2 shows the encrypted file which was stored by a user.



```
F]r]lh]pbkni
I]ej]npe_hao6F]r]$okbps]nalh]pbkni%]j`F]r]renpq]hi
]_deja
Kja_d]n]_paneope_kbF]r]eolknp]^ehepu(sde_dia]jopd]
p_kilqpanlnkcn]iosneppajejpdaF]r]h]jcq]caiqopnqjoe
```

Figure 5.2: Encrypted Confidential File Stored In Storage Cloud

2. Secondly, the keys, hash of data are stored in security cloud server in "userkey" table, while the hash of corresponding data is stored in "filehash" table.



Format	Input	Output
Authentication	username/password dialog box	HTML Pages served after successful authentication
Query	Insert Command	Insertion of records
	Delete Command	One can also delete records inserted into the table
	Update Command	Modifying data already entered into the table

6. Conclusion

In this paper, a privacy-conserving public appraising mechanism for shared data in the cloud has been proposed. With this mechanism, the TPA is able to proficiently audit the integrity of shared data, yet cannot differentiate who is the signer on each block, which can maintain identity privacy for users. It offers user ambiguity in authentication phase, data integrity and confidentiality and the fair revocation process for all users. To improve the efficiency of verifying multiple auditing tasks, this mechanism supports batch auditing.

An interesting problem in future work is how to efficiently audit the integrity of shared data with dynamic groups while still preserving the identity of the signer on each block from the third party auditor.

References

[1] Boyang Wang, Baochun Li and Hui Li, “Oruta: Privacy-Preserving Public Auditing for Shared Data in the Cloud”, IEEE Transactions on Cloud Computing, Vol. 2, No. 1, 2014.

[2] Mohamed Nabeel and Elisa Bertino, “Privacy Preserving Delegated Access Control in Public Clouds”, IEEE Transactions on Knowledge and Data Engineering, Vol. 26, No. 9, 2014

[3] Lukas Malina and Jan Hajny, “Efficient Security Solution for Privacy-PreservingCloud Services”, 6th International Conference on Telecommunications Signal Processing Year 2013

[4] Cheng-Kang Chu, Sherman S.M. Chow, Wen-GueyTzeng, Jianying Zhou, Robert H. Deng, “Key-Aggregate Cryptosystem for Scalable Data Sharing in Cloud Storage”, IEEE Transactions on Parallel and Distributed Systems, Vol. 25, No. 2, 2014

[5] Yong Yu, Jianbing Ni, Man Ho Au, Yi Mu, Boyang Wang, and Hui Li, “On the Security of a Public Auditing Mechanism for Shared Cloud Data Service”, IEEE Transactions on Services Computing, 2014

[6] Hong Liu, HuanshengNing, QingxuXiong, Member, Laurence T. Yang, “Shared Authority Based Privacy-Preserving Authentication Protocol in Cloud Computing”, IEEE Transactions on Parallel and Distributed Systems, Vol. 26, No. 1, 2015

[7] M. Nabeel, N. Shang, and E. Bertino, “Privacy Preserving Policy Based Content Sharing in Public Clouds,” IEEE Trans. Knowledge and Data Eng., vol. 25, no. 11, pp. 2602-2614, Nov. 2013.

[8] C. Wang, S.S.M. Chow, Q. Wang, K. Ren, and W. Lou, “Privacy-Preserving Public Auditing for Secure Cloud Storage,” IEEE Trans. Computers, vol. 62, no. 2, pp. 362-375, Feb. 2013.

[9] Y. Zhu, G.-J. Ahn, H. Hu, S. S. Yau, H. G. An, and S. Chen, “Dynamic Audit Services for Outsourced Storage in Clouds,” IEEE Transactions on Services Computing, vol. 6, no. 2, pp. 227–238, 2013.

[10] B. Wang, B. Li, and H. Li, “Panda: Public Auditing for Shared Data with Efficient User Revocation in the Cloud,” IEEE Transactions on Services Computing, doi: 10.1109/TSC.2013.2295611, 2013.

[11] I.T. Lien, Y.H. Lin, J.R. Shieh, and J.L. Wu, “A Novel Privacy Preserving Location-Based Service Protocol with Secret Circular Shift for K-NN Search,” IEEE Trans. Information Forensics and Security, vol. 8, no. 6, pp. 863-873, June 2013.

[12] A. Barsoum and A. Hasan, “Enabling Dynamic Data and Indirect Mutual Trust for Cloud Computing Storage Systems,” IEEE Trans. Parallel and Distributed Systems, vol. 24, no. 12, pp. 2375-2385, Dec. 2013.

[13] H.Y. Lin and W.G. Tzeng, “A Secure Erasure Code-Based Cloud Storage System with Secure Data Forwarding,” IEEE Trans. Parallel and Distributed Systems, vol. 23, no. 6, pp. 995-1003, June 2012.

[14] J.Yu,P.Lu,G.Xue,andM.Li, “Towards Secure Multi-Keyword Top k Retrieval over Encrypted Cloud Data,” IEEE Trans. Dependable and Secure Computing,vol.10, no.4, pp.239-250, July/Aug.2013.