IPHV8I1031X
International Journal Of Advanced Research and Innovation -Vol.8, Issue .I
ISSN Online: 2319 – 9253
Print: 2319 – 9245

EFFICIENT AND SECURE KAC SCHEME FOR DISTRIBUTED CLOUD STORAGE

[#1]RAMESH KATLA, M.Tech Student,
[#2]RAVI MATHEY, Professor& HOD,
Dept of CSE,
VIDYA JYOTHI INSTITUTE OF TECHNOLOGY, HYDERABAD, TELANGANA, INDIA.

Abstract: Data sharing is an important functionality in cloud storage. In this previous work, we show how to securely, efficiently, and flexibly share data with others in cloud storage. The existing work presents the Key- Aggregate Cryptosystem(KAC) used for conveniently sent to others or be stored in a smart card with very limited secure storage. A limitation of existing work is the predefined bound of the number of maximum cipher text classes and key is prompt to leakage. Our proposed work mainly concentrates on above two problems. Our first work dynamically reserve number of maximum cipher text classes in cloud storage. In case of Stream cipher the number of classes decided dynamically, because the cipher text size is too larger than block cipher.We proposes a perfect decentralized access control scheme with aggregate key encryption for data stored in cloud. This scheme provides secure data storage and retrieval. Along with the security the access policy is also hidden for hiding the user's identity. This scheme is so powerful since we use aggregate encryption and string matching algorithms in a single scheme. The scheme detects any change made to the original file and if found clear the error's. The algorithm used here are very simple so that large number of data can be stored in cloud without any problems. The security, authentication, confidentiality are comparable to the centralized approaches.

Keywords: *Cloud Storage, Data Sharing, Asymmetric Encryption, String matching algorithms, Key- Aggregate Cryptosystem*.

I.INTRODUCTION

Cloud storage is gaining popularity recently. In enterprise settings, we see the rise in demand for data outsourcing, which assists in the strategic management of corporate data. It is also used as a core technology behind many online services for personal applications. Now a days, it is easy to apply for free accounts for email, photo album, file sharing and/or remote access, with storage size more than 25 GB (or a few dollars for more than 1 TB). Together with the current wireless technology, users can access almost all of their files and emails by a mobile phone in any corner of the world. Considering data privacy, a traditional way to ensure it is to rely on the server to enforce the access control after authentication, which means any unexpected privilege escalation will expose all data. In a shared-ten an cycloid computing environment, things become even worse. Data from different clients can be hosted on separate virtual machines (VMs) but reside on a single physical machine. Data in a target VM could be stolen by instantiating another VM resident with the target one. Regarding availability of files, there are a series of cryptographic schemes which go as far as allowing a third-party auditor to check the availability of files on behalf of the data owner without leaking anything about the data, or without compromising the data owner's anonymity. Likewise, cloud users probably will not hold the strong belief that the cloud server is doing a good job in terms of confidentiality. A cryptographic solution, for example, with proven security relied on number-theoretic assumptions is more desirable, whenever the user is not perfectly happy with trusting the security of the VM or the honesty of the technical staff. These users are motivated to encrypt their data with their own keys before uploading them to the server.

Clouds can provide several types of services like applications (e.g., Google Apps, Microsoft online), infrastructures (e.g., Amazon's EC2, Eucalyptus, Nimbus), and platforms to help developers write applications (e.g., Amazon's S3, Windows Azure).Security is needed because data stored in clouds is highly sensitive, for example, medical records and social networks. User privacy is also required so that the cloud or other users do not know the identity of the user. Thus it is a complex system which possesses highly securable processes. So it must need a proper systematic scheme to manage data.

Recently S. Yu, C. Wang, K. Ren, and W. Lou proposed a system which is based on attribute based encryption for Fine-Grained Access Control of Encrypted Data. To keep sensitive user data confidential against

unauthenticated servers, existing schemes usually apply cryptographic methods by disclosing data decryption keys only to authorized users. We combine techniques of attribute-based encryption [2] (ABE)and several other techniques. The problem in this latest technique is that Single data owner will be easily be overwhelmed by the key management overhead. So apart from security concerns we have to concentrate on the key distribution also.
Search on encrypted data is also a major concern in cloud. Also [4] hiding of access policy is also needed. So encryption must be done in a perfect manner. Several recent encryption algorithm fails in searching process. But the best encryption algorithm which also makes search better is aggregate type encryption [1].Thus this encryption technique is used mostly. Providing security only is very simple but providing security with privacy[2] is very much difficult. Maintaining the privacy is very much important because it is very easy for intruders to access the confidential data. Since very confidential data's are stored in cloud it is very much needed to maintain the security and privacy. Using homomorphic encryption, the cloud receives cipher text of the data and performs computations on the ciphertext and return's the encoded value. Now the user converts the value, but the cloud does not know what data it has operated on. These are the common problems in cloud. So this area must be concentrated.

Transactions done in the cloud should also be noted periodically. The user should be verified and should give appropriate permission for them. Permission criteria are carefully handled because users may change the data unnecessarily. So this area should be concentrated too much. Adding this kind of feature may automatically reduce the efficiency of the algorithm, so the algorithm designed must be very efficient. It must consider all the additional features and the system should be maintained accordingly. Consider the following situation: A student from a college found out some malpractices done by some employees in college. Then the student takes steps to tell the details about the malpractice done in the college. Now he will report the malpractice done by the employees of the college to the university which controls the college. While reporting there are some conditions to be checked seriously. First the student should prove the identity because the university should believe that the message came from an authorised person. Second there should not be any interference. Also if any change is done for the original message then it should be found out and the file is recovered. Thus in this paper the above problems are described and rectified.
An area where access control is widely being used is health care[14]. Clouds are being used to store sensitive information about patients to enable access to medical professionals, hospital staff, researchers, and policy makers. It is important to control the access of data so that only authorized users can access the data. Using Aggregate key encryption [1], the records are encrypted under some access policy and stored in the cloud. Users are given sets of keys.
Only when the users have matching set of keys, can they decrypt the information stored in the cloud. Access control is also gaining importance in online social networking.

II. RELATED WORK

Attribute based encryption[7][8][12][13](ABE) was proposed by Sahai and Waters [26]. In ABE, a user has a set of attributes based on the user in addition to its unique ID. In Key-policy ABE or KP-ABE (Goyal et al.[27]), the sender has an access policy to encrypt data. A writer whose attributes and keys have been revoked cannot write back stale information. The receiver receives attributes and secret keys from the attribute authority and is able to decrypt information if it has matching attributes. In Ciphertext-policy, CP-ABE ([28],[29]), the receiver has the access policy in the form of a tree, with attributes as leaves and monotonic access structure with AND, OR and other threshold gates.
All the approaches take a centralized approach and allow only one KDC, which is a single point of failure. Chase [2] proposed a multi-authority ABE, in which there are several KDC authorities(coordinated by a trusted authority) which distribute attributes and secret keys to users. Multi-authority ABE protocol was studied in [7], [8], which required no trusted authority which requires every user to have attributes from at all the KDCs. Recently, Lewko and Waters [9] proposed a fully decentralized ABE where users could have zero or more attributes from each authority and did not require a trusted server. In all these cases, decryption at user's end is computation intensive. So, this technique might be inefficient when users access using their mobile devices. However, as mentioned earlier in the previous section it is prone to replay attack.
To reduce or block replay attack we use string matching algorithms [3][5] which is more efficient and perfect in security. It works more efficient than all other matching algorithms.

EXISTING SYSTEM

Encryption keys also come with two flavors—symmetric key or asymmetric (public) key. Using symmetric encryption, when Alice wants the data to be originated from a third party, she has to give the encryptor her secret key; obviously, this is not always desirable. By contrast, the encryption key and decryption key are different in publickey encryption. The use of public-key encryption gives more flexibility for our applications. For example, in enterprise settings, every employee can upload encrypted data on the cloud storage server without the knowledge of the company's master-secret key. Introducing a special type of public-key encryption which we call key-aggregate cryptosystem (KAC). In KAC, users encrypt a message not only under a public-key, but also under an identifier of cipher text called class. That means the cipher texts are further categorized into different classes. The key owner holds a master-secret called master-secret key, which can be used to extract secret keys for different classes. More importantly, the extracted key have can be an aggregate key which is as compact as a secret key for a single class, but aggregates the power of many such keys, i.e., the decryption power for any subset of cipher text classes. The sizes of cipher text, public-key, master-secret key, and aggregate key in KAC schemes are all of constant size. The public system parameter has size linear in the number of cipher text classes, but only a small part of it is needed each time and it can be fetched on demand from large (but non confidential) cloud storage.

Issues

  This work is the predefined bound of the number of maximum cipher text classes.

  When one carries the delegated keys around in a mobile device without using special trusted hardware, the key is prompt to leakage.

III. AUDIT SYSTEM ARCHITECTURE

The audit system architecture for outsourced data in clouds in which can work in an audit service outsourcing approach. In this architecture, we reflect on a data storage service containing four entities:

1) Data owner (DO): who has data files to be stored in the cloud and relies on the cloud for data maintenance, can be an individual customer or an organization.

2) Cloud Storage Service Provider (CSP): who provides data storage service and has enough storage space to maintain client"s data.

3) Third Party Auditor (TPA): a trusted person who manage or monitor outsourced data under request of the data owner.

4) Authorized Application (AA): who have the right to access and manipulate stored data.

The data which the data owner wants to store in cloud first reaches the authorized application which will create digital signature and sends the data to the cloud storage. If the user needs to verify data means the verification request should be send to third party auditor (TPA), the TPA will retrieve the digital signature from the database and will send the verification request to the management server. The management server in turn will generate the digital signature for the data stored in the cloud and it will send only that digital signature instead of the whole data to the TPA. The TPA will decrypt the digital signature and compares the message digest for verifying correctness of data.

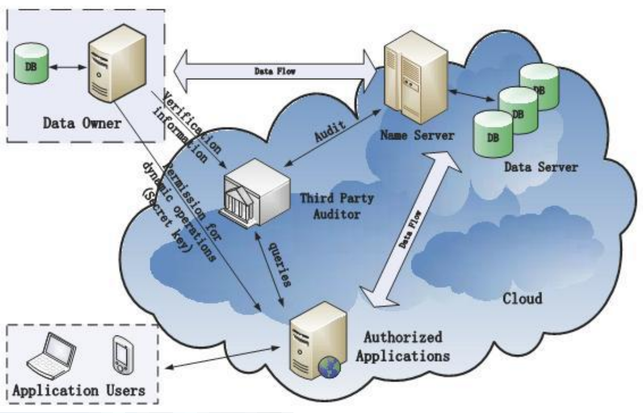

Figure1 : Architecture Diagram

This architecture is known as the audit service outsourcing due to data integrity authentication. Architecture contains the data owner and granted clients need to dynamically interact with cloud service provider to access or update their data for various application purposes. However, we neither assume that cloud service provider is trust to guarantee the security of stored data, or suppose that the data owner has the capability to collect the verifications of cloud service provider"s fault after errors occur. Hence, third party auditor, as a trust third party (TTP), is used to ensure the storage security of their outsourced data. We assume the third party auditor is reliable and independent, and thus has no encouragement to join together with either the cloud service provider or the clients during the auditing process: • TPA must be able to make regular check on the integrity and availability of these delegated data at appropriate intervals; • TPA must be able to take the evidences for the disputes about the inconsistency of data in terms of authentic records for all data operations. To facilitate privacy-preserving public auditing for cloud data storage beneath the architecture, the protocol design should attain subsequent security and performance guarantees:

1) Audit-without-downloading: to allow TPA (or other clients with the help of TPA) to authenticate the correctness of cloud data on demand without recovering a copy of whole data or bring in additional on-line burden to the cloud users;
2) Verification-correctness: to make sure there exists no unethical CSP that can pass the audit from TPA without indeed storing users‟ data intact;
3) Privacy-preserving: to make sure that there exists no way for TPA to derive users‟ data from the in sequence collected during the auditing process;
4) High-performance: to allow TPA to perform auditing with minimum overheads in storage, communication and computation, and to maintain statistical audit sampling and optimized audit schedule with a long enough period of time.

IV. PROPOSED METHOD

A. Framework

The basis or outline of the key-aggregate encryption scheme consists of five polynomial-time algorithms, which are elucidated below: Setup ensures that the owner of the data can construct the public system stricture or parameter. KeyGen, as the name suggests generates a public/master secret (not to be confused with the delegated key explained later) key pair. By using this public and master-secret key cipher text class index he can convert plain text into cipher text via use of Encrypt. Using Extract, the master-secret can be utilized to generate an aggregate decryption key for a set of cipher text classes. These generated keys can be safely transported to the appointees by use of secure mechanisms with proper security measures adhered to. If and only if the cipher text's class index is enclosed in the single key, then every user with an aggregate key can decrypt the given cipher text provided through the use of Decrypt.

B. Algorithm

1. Setup(Security level parameter, number of cipher text classes): Setup ensures that the owner of the data can construct the public system stricture or parameter he create account on cloud. After entering the input, the total of cipher text classes n and a security level parameter 1, the public system parameter is given as output, which usually skipped from the input of other algorithms for the purpose of conciseness.
2. KeyGen: it is for generation of public or master key secret pair.
3. Encrypt(public key,index,message):run any person who want to convert plaintext into cipher text using public and master-secret key
4. Extract(master key, Set): Give input as master secret key and S indices of different cipertext class it produce output aggregate key. This is done by executing extract by the data owner himself. The output is displayed as the aggregate key represented by Ks, when the input is entered in the form the set S of indices relating to the various classes and mastersecret key msk.
5. Decrypt (Ks,S,i,C): When an appointee receives an aggregate key Ks as exhibited by the previous step, it can execute Decrypt. The decrypted original message m is displayed on entering Ks, S, i, and C, if and only if I belongs to the set S.

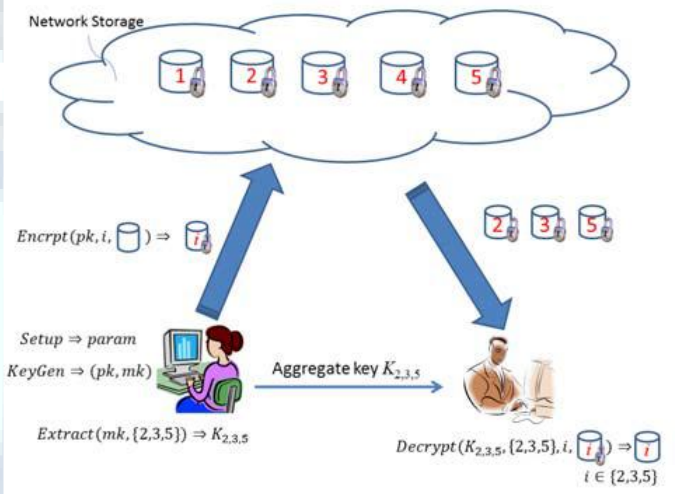
Network Storage
1
2
3
4
5
Encrpt(pk, i,) ⇒
i
2
3
5
Setup ⇒ param
KeyGen ⇒ (pk, mk)
Aggregate key $K_{2,3,5}$
Extract(mk, {2,3,5}) ⇒ $K_{2,3,5}$
Decrypt($K_{2,3,5}$, {2,3,5}, i,) ⇒
$i \in \{2,3,5\}$

Fig2. Proposed KAC for data sharing in cloud storage system

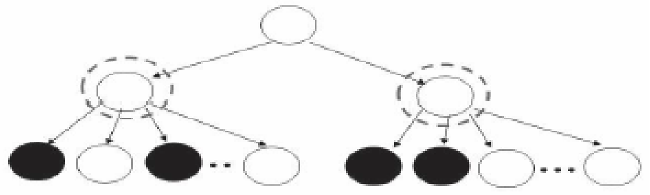

Fig.3.Key Assignment

V. CRYPTOSYSTEM ME6

Plaintext is readable data (for example, a spreadsheet file), and ciphertext is the result of encrypting plaintext. A cryptosystem is a set of procedures and conventions for hiding and revealing information in a controlled manner. A cryptosystem generally has two distinct components:
(a) the processes used to encipher and decipher data and
(b) the set of keys used to influence the operation of these processes so that the ciphertext is dependent on the key used for encryption.
The security of a cryptosystem lies not in the secrecy of the methods used to encipher and decipher the data but rather in the difficulty of decrypting ciphertext in the absence of

knowledge of the key used to produce it. Cryptosystem ME6 encrypts data in files stored on disk. A file may be considered as a sequence of at least one byte and perhaps many millions of bytes. ME6 reads in plaintext from a file in blocks whose size is between 6 KB and 10 KB (the exact size of each block depends on the encryption key), encrypts each block and writes the resulting ciphertext to disk. This is done for each of the blocks making up the file. Each block is first compressed, if possible, before being encrypted, so normally the ciphertext blocks are smaller than the plaintext blocks, with the result that the file containing the encrypted data is usually smaller than the input file.

VI.RESULT AND DISCUSSION

Our approaches change the compression issue F (F =n in our schemes) to be a tunable parameter, at the cost of O(n)-sized system parameter. cryptography is tired constant time, whereas coding is tired O(|S|) cluster multiplications (or purpose addition on elliptic curves) with 2 pairing operations, where S is that the set of ciphertext classes decryptable by the granted mixture key and |S| ≤ n. of course, key extraction wants O(|S|) cluster multiplications additionally, that a replacement advance on the stratified key assignment (a ancient approach) that preserves areas providing the entireties of the key-holders share similar edges is our approach of “compressing” secret keys in public key cryptosystems. These public key cryptosystems manufacture cipher texts of constant size nominal economical delegation of secret writing rights for any set of cipher texts is possible. This not exclusively enhances user privacy and confidentiality of data in cloud storage, but it'll this by supporting the distribution or appointing of secret keys varied for diverse} cipher text classes and generating keys by numerous derivation of cipher text class properties of the information and its associated keys. This sums up the scope of our paper. As there is a limit attack selection the quantity the quantity} of cipher text classes beforehand & in addition to the exponential growth inside the quantity of cipher texts in cloud storage, there is a demand for reservation of cipher text classes for future use. As for potential modifications and enhancements to our current cause, in future, the parameter size area unit usually altered nominal it's freelance of the utmost style of cipher text classes. to boot, a specially designed cryptosystem, with the employment of an accurate security formula, as associate degree example, the Diffie-Hellman Key-Exchange methodology, which can then be imperviable, or at the foremost proof against outpouring at the aspect of economical key appointing, will confirm that one can transport same keys on mobile devices without fear of outpouring.

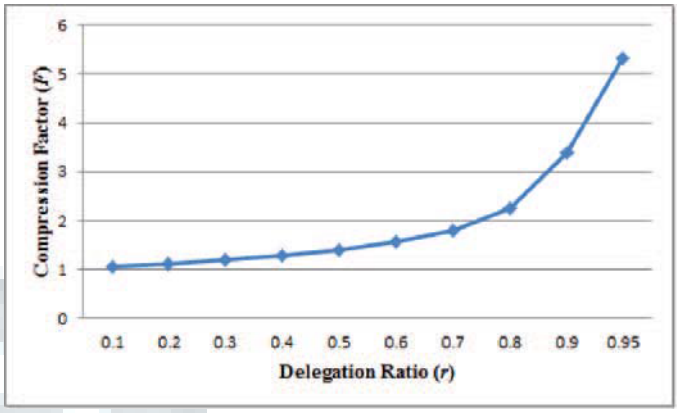

Fig 4. (A) Compression achieved by the tree-based approach for delegating different ratio of the classes

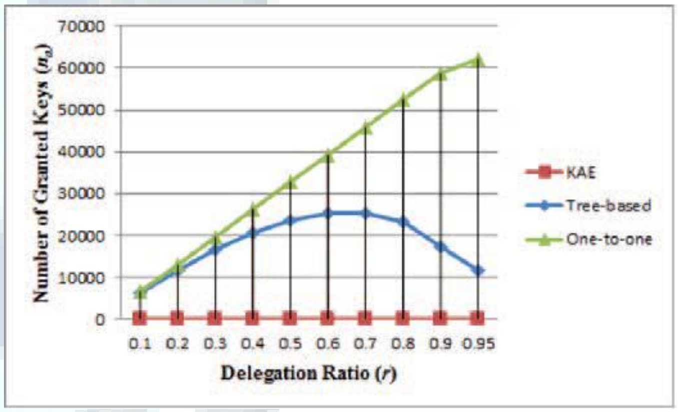

Fig 4. (B) Number of granted keys (*na*) required for different approaches in the case of 65536 classes of data.

VII.CONCLUSION

We consider how to ―compress‖ secret keys in public-key cryptosystems which support delegation of secret keys for different cipher text classes in cloud storage. No matter which one among the power set of classes, the delegate can always get an aggregate key of constant size. Our approach is more flexible than hierarchical key assignment which can only save spaces if all key-holders share a similar set of privileges. The work is giving an efficient privacy-preserving storage compared to other works. Even though there are many approaches in the literature for mitigating the concerns in privacy, no approach is fully sophisticated to give a privacy-preserving storage that overcomes all the other privacy concerns. Thus to deal with the concerns of privacy, we need to develop privacy–preserving framework that overcomes the worries in privacy security and encourage users to adopt cloud storage services more confidently. Our approach is more flexible than hierarchical key assignment which can only save spaces if all key-holders share a similar set of privileges. A limitation in our work is the predefined bound of the number of maximum ciphertext classes. In cloud storage, the number of ciphertexts usually grows rapidly. So we have to reserve enough ciphertext classes for the future extension.

REFERENCES

[1] Yan Zhua,b, Hongxin Huc, Gail-Joon Ahnc, Stephen S. Yauc. "Efficient audit service outsourcing for data integrity in clouds". In "The Journal of Systems and Software 85 (2012) 1083– 1095".

[2] 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," Comm. ACM, vol. 53, no. 4, pp. 50-58, 2010.

[3] T. Velte, A. Velte, and R. Elsenpeter, Cloud Computing: A Practical Approach, first ed., ch. 7. McGraw-Hill, 2010.

[4] A. Juels and B.S. Kaliski Jr., "PORs: Proofs of Retrievability for Large Files," Proc. 14th ACM Conf. Computer and Comm. Security (CCS "07), pp. 584-597, Oct. 2007.

[5] 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. 598-609, Oct. 2007.

[6] M.A. Shah, M. Baker, J.C. Mogul, and R. Swaminathan, "Auditing to Keep Online Storage Services Honest," Proc. 11th USENIX Workshop Hot Topics in Operating Systems (HotOS "07), pp. 1-6,2007.

[7] 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. 598-609, 2007.

[8] M.A. Shah, R. Swaminathan, and M. Baker, "Privacy-Preserving Audit and Extraction of Digital Contents," Cryptology ePrint Archive, Report 2008/186, 2008.

[9] A. Juels and J. Burton, S. Kaliski, "PORs: Proofs of Retrievability for Large Files," Proc. ACM Conf. Computer and Comm. Security (CCS "07), pp. 584-597, Oct. 2007.

[10] Q. Wang, C. Wang, K. Ren, W. Lou, and J. Li, "Enabling Public Auditability and Data Dynamics for Storage Security in Cloud Computing," IEEE Trans. Parallel Distributed Systems, vol. 22, no. 5,pp. 847-859, May 2011.

[11] 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.

[12] C. Wang, K. Ren, W. Lou, and J. Li, "Toward Publicly Auditable Secure Cloud Data Storage Services," IEEE Network, vol. 24, no. 4, pp. 19-24, July/Aug. 2010.

[13] K. Yang and X. Jia, "Data Storage Auditing Service in Cloud Computing: Challenges, Methods and Opportunities," World Wide Web, vol. 15, no. 4, pp. 409-428, 2012.

[14] Q. Wang et al., "Enabling Public Verifiability and Data Dynamics for Storage Security in Cloud Computing," Proc. ESORICS „09, Sept. 2009, pp. 355–70.

[15] C. Erway et al., "Dynamic Provable Data Possession," Proc. ACM CCS „09, Nov. 2009, pp. 213–222.

[16] C. Wang et al., "Privacy-Preserving Public Auditing for Storage Security in Cloud Computing," Proc. IEEE INFOCOM „10, Mar. 2010.

[17] Cong Wang and Kui Ren, Illinois Institute of Technology Wenjing Lou, Worcester Polytechnic Institute Jin Li, Illinois Institute of Technology "Toward Publicly Auditable Secure Cloud Data Storage Services". 0890-8044/10/2010 IEEE.

[18] Cong Wang, Student Member, IEEE, Qian Wang, Student Member, IEEE, Kui Ren, Senior Member, IEEE, Ning Cao, and Wenjing Lou, Senior Member, IEEE "Toward Secure and Dependable Storage Services in Cloud Computing" IEEE TRANSACTIONS ON SERVICES COMPUTING, VOL. 5, NO. 2, APRIL-JUNE 2012.

[19] Kan Yang, Student Member, IEEE, and Xiaohua Jia, Fellow, IEEE "An Efficient and Secure Dynamic Auditing Protocol for Data Storage in Cloud Computing" IEEE ON PARALLEL AND DISTRIBUTED SYSTEMS, VOL. 24, NO. 9, SEPTEMBER 2013.

[20] Cong Wang, Member, IEEE, Sherman S.M. Chow, Qian Wang, Member, IEEE, Kui Ren, Senior Member, IEEE, and Wenjing Lou, Senior Member, IEEE "Privacy-Preserving Public Auditing for Secure Cloud Storage" IEEE TRANSACTIONS ON COMPUTERS, VOL. 62, NO. 2, FEBRUARY 2013.

[21] Cheng-Kang Chu, Sherman S. M. Chow, Wen-Guey Tzeng, Jianying Zhou, and Robert H. Deng, Senior Member, IEEE," Key-Aggregate Cryptosystem for Scalable Data Sharing in Cloud Storage," IEEE Transactions on Parallel and Distributed Systems,IEEE,2013.

[22] 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, 2013.

[23] B. Wang, S. S. M. Chow, M. Li, and H. Li, "Storing Shared Data on the Cloud via Security-Mediator," in *International Conference on Distributed Computing Systems - ICDCS 2013*. IEEE, 2013.

[24] M. J. Atallah, M. Blanton, N. Fazio, and K. B. Frikken,"Dynamic and Efficient Key Management for Access Hierarchies," *ACMTransactions on Information and System Security (TISSEC)*, vol. 12,no. 3, 2009.

[25] W.-G. Tzeng, "A Time-Bound Cryptographic Key Assignment Scheme for Access Control in a Hierarchy," *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, vol. 14, no. 1, pp. 182–188,2002.