IMPROVED SECURE STORAGE AS SERVICE IN CLOUD COMPUTING

Mutharasi N#1, Ajitha P M.E.,#2

1 Student, Department of Information Technology, Sathyabama University, Chennai(T.N), India.
2 Assistant Professor, Department of Information Technology, Sathyabama University, Chennai(T.N), India.

Abstract— Cloud is the vast area where multiple accesses to the storage has been emerging. Data sharing is the important thing which was accessible around the world by multiple entities. In my proposed work, data has been stored and accessed from cloud in an efficient manner by encrypting with asymmetric algorithm and generating the hash value for the encrypted data. To ensure high security we are splitting the encrypted data into 'n' number of parts and storing it in different cloud servers. During the retrieval process integrate the different parts of the encrypted data and decrypt the file in clients place. Thus providing high security to the data stored in the cloud space. Our proposed scheme is highly efficient even if server crashes.

Keywords—Cloud storage, Splitting and backup, Recovery of data

1. INTRODUCTION

Cloud storage services make it easy to securely transfer confidential data. In smaller companies, locally stored files are at risk from theft, software problems and hardware failure. When Internet service provider goes down the entire organization can lose data access and services. But many cloud service providers cache data locally as well as online, even if the ISP goes down we still have access to the data. To provide enhanced security we are splitting the encrypted data into 'n' number of parts and storing it in different cloud servers. During the retrieval process integrate the different parts of the encrypted data and decrypt the data in clients place. Thus providing high security to the data stored in the cloud space. Our proposed scheme is highly efficient even if server crashes.

Nowadays, there is an emerging trend that increasingly more customers are beginning to use the public cloud storage for online data sharing and storing the data. However, these data applications in cloud storage have some security issues such as data confidentiality and information leakage from users data. Once the users publish their private data to the public cloud storage they lose the direct control of their data and have to trust the cloud storage service provider reluctantly. Unfortunately the CSP is usually un-trusted and the user's data gets corrupted. To protect their data, customers need to encrypt the data before sending to the cloud storage and then enforce the self reliant authorization by building a cryptographic access control system.

If the user needs to upload the file into the cloud server it has to encrypt the data. Hashing technique is mainly used to indicate whether the data is corrupted or not. The encrypted data splitted into n number of parts based on the probability and is stored in different cloud servers to protect data from hackers. The backup of the data is stored randomly in different servers and hence it helps to recover the data latter if any server crashes. The authorized user can download the data which will be decrypted to get the corresponding file.

2. RELATED WORK

Encryption is a way to enhance the security of a message or file by scrambling the contents so that only someone who has the right encryption key to unscramble it can read it Attribute based encryption (ABE) is a relatively recent approach that reconsiders the concept of public key cryptography[1]. A distributed system consists of a collection of autonomous computers connected through a network and distribution middleware, which enables computers to coordinate their activities and to share the resources of the system, so that users perceive the system as a single, integrated computing facility. In several distributed systems a user should only be able to access data if a user possesses a certain set of credentials or attributes[2]. A system for realizing complex access control on encrypted data that it call Cipher text Policy Attribute Based Encryption is presented. Most existing public key encryption methods allow a party to encrypt data to a particular user, but are unable to efficiently handle more expressive types of encrypted access control. By using techniques encrypted data can be kept confidential even if the storage server is untrusted; moreover methods are secure against collusion attacks[3,2].

The system allows for a new type of encrypted access control where user's private keys are specified by a set of attributes and a party encrypting data can specify a policy over these attributes specifying which users are able to decrypt[4]. Also allows policies to be expressed as any monotonic tree access structure and is resistant to collusion attacks in which an attacker might obtain multiple private keys. While Key Policy ABE and Cipher text Policy ABE capture two interesting and complimentary types of systems there certainly exist other types of systems. It would be interesting to consider attribute based encryption systems with different types of expressibility. The primary challenge in the line of work is to find new systems with elegant forms of expression that produce more than an arbitrary combination of techniques. One limitation of system is that it is proved secure under the generic group heuristic. The type of work would be interesting even if it resulted in a moderate loss of efficiency from existing system.

In cryptography, a public key is a value provided by some designated authority as an encryption key that, combined with a private key derived from the public key, can be used to effectively encrypt messages and digital signatures Public Key encryption is a powerful mechanism for protecting the confidentiality of stored and transmitted information Traditionally encryption is viewed as a method for a user to share data to a targeted user or device[5.7].

Public key encryption can be used for authentication, confidentiality and the integrity and the non repudiation. Key Policy ABE, attributes are used to annotate the cipher texts and formulas over these attributes are ascribed to users' secret keys Cipher text Policy ABE, is complementary in that attributes are used to describe the user's credentials and the formulas over these credentials are attached to the cipher text by the encrypting party[6]. A new methodology is presented for realizing Cipher text Policy ABE systems from a general set of access structures in the standard model under concrete and non interactive assumptions. The system allows an encryption algorithm to specify an access formula in terms of any access formula expressing access control by a Linear Secret Sharing Scheme (LSSS) matrix M over the attributes in the system. Do not lose any efficiency by using the more general LSSS representation as opposed to the previously used tree access structure descriptions. Thus, achieving the same performance and functionality as the Bethencourt, Sahai, and Waters construction, but under the standard model. In addition provide two other constructions that tradeoff some performance parameters for provable security under the respective weaker assumptions of decisional Bilinear Diffie Hellman Exponent (d BDHE) and decisional Bilinear Diffie Hellman assumptions. The first cipher text policy attribute based encryption systems that are efficient, expressive and provably secure under concrete assumptions. All of the constructions fall under a common methodology of embedding an LSSS challenge matrix directly into the public parameters. The constructions provide a trade of in terms of efficiency and the complexity of assumptions[8,9].

Attribute based encryption (ABE) is a relatively recent approach that reconsiders the concept of public key cryptography. In an ABE system, a party encrypting data can specify access to the data as a boolean formula over a set of attributes. Each user in the system will be issued a private key from an authority that reflects their attributes (or credentials). A user will be able to decrypt a ciphertext if the attributes associated with their private key satisfy the Boolean formula as described to the ciphertext. A multi authority ABE system consists of any number attribute authorities and any number of users. A set of global public parameters is defined in the system. A user can select an attribute authority and obtain the corresponding decryption keys. The authority executes the corresponding attribute key generation algorithm and the result is returned to the user. A Multi Authority Attribute Based Encryption (ABE) system is proposed. In the system, any party can become an authority and there is no requirement for any global coordination other than the creation of an initial set of common reference parameters. A party can simply act as an ABE authority by creating a public key and issuing private keys to different users that reflect their attributes. A user can encrypt data in terms of any Boolean formula over attributes issued from any chosen set of authorities. The system does not require any central authority. The ultimate goal of designing a MA ABE scheme is to develop a secure, robust expressive and efficient multi authority attribute based encryption system. The field of MA ABE scheme is a vast and ever evolving one with its wings stretched to the areas of IT and Social Network. System secure using the recent dual system encryption methodology where the security proof works by first converting the challenge cipher text and private keys to a semi functional form and then arguing security is proved. The system does not require any central authority. Thus avoid the performance bottleneck incurred by relying on a central authority which makes the system more scalable Also avoid placing absolute trust in a single designated entity which must remain active and uncorrupted throughout the lifetime of the system. This is a crucial improvement for efficiency as well as security, since even a central authority that remains uncorrupted may occasionally fail for benign reasons, and a system that constantly relies on its participation will be forced to remain stagnant until it can be restored

Secure sharing of personal data

In recent years, personal health record (PHR) has emerged as a patient centric model of health information exchange. A PHR service allows a patient to create, manage, and control her personal health data in one place through the web which has made the storage retrieval, and sharing of the medical information more efficient. Especially, each patient is promised the full control of her medical records and can share her health data with a wide range of users, including healthcare providers, family members or friends. Due to the high cost of building and maintaining specialized data centers, many PHR services are outsourced to or provided by third party service providers, for example, Microsoft Health Vault Recently, architectures of storing PHRs in cloud computing have been proposed While it is exciting to have convenient PHR services for everyone, there are many security and privacy risks, which could impede its wide adoption. The main concern is about whether the patients could actually control the sharing of their sensitive personal health information (PHI), especially when they are stored on a third party server which people may not fully trust. A novel framework of secure sharing of personal health records in cloud computing is proposed. Considering partially trustworthy cloud servers, argue that to fully realize the patient centric concept patients shall have complete control of their own privacy through encrypting their PHR files to allow fine grained access. The framework addresses the unique challenges brought by multiple PHR owners and users, in that it greatly reduce the complexity of key management while enhance the privacy guarantees compared with previous works. It utilize ABE to encrypt the PHR data, so that patients can allow access not only by personal users, but also various users from public domains with different professional roles Qualifications, and affiliations. The scheme also enables dynamic modification of access policies or file attributes supports efficient on demand user/attribute revocation and break glass access under emergency scenarios.

Data sharing with revocation

Cipher text policy attribute based encryption (CP ABE) is a public key cryptography primitive that was proposed to resolve the exact issue of fine grained access control on shared data in one to many communications. In CP ABE, each user is assigned a set of attributes which are embedded into the user‘s secret key. A public key component is defined for each user attribute. CP ABE is resistant to collusion attacks from unauthorized users. All these nice properties make CP ABE extremely suitable for fine grained data access control on untrusted storage. Existing CP ABE schemes are not able to simultaneously achieve provable security expressiveness of access structure, and efficient construction; user management user revocation in particular, is extremely hard to realize in an efficient way. When current researches are mainly focusing on solving the former, the later has drawn less attention. User revocation is a challenge issue in many one to many communication systems. In attribute based systems, this issue is even more difficult since each attribute is conceivably shared by multiple users. Either it may require to revoke the entire user access privilege, or just partial access right of the user, i.e., a subset of his/her attributes. Existing CP ABE schemes suggest associating expiration time attributes to user secret keys. In CP ABE, each user is associated with a set of attributes and data are encrypted with access structures on attributes. A user is able to decrypt a cipher text if and only if his attributes satisfy the cipher text access structure. An important issue of attribute revocation which is cumbersome for CP ABE schemes is focused. In particular, resolve the challenging issue by considering more practical scenarios in which semi trustable on line proxy servers are available It achieve this by uniquely integrating the technique of proxy re encryption with CP ABE, and enable the authority to delegate most of laborious tasks to proxy servers. As compared to existing schemes, the proposed solution enables the authority to revoke user attributes with minimal effort. Formal analysis shows that the proposed scheme is provably secure against chosen cipher text attacks. One nice property of the proposed scheme is that it places minimal load on authority upon attribute revocation events.

3. EXISTING SYSTEM

In existing system it will check the attributes of the users whether the receiver have the same attributes as the sender mentioned or not. It will avoid the hackers or unauthorized users to access the data. The sender gives the attributes of the receiver while sending the file to the receiver. Then the data gets encrypted as per the given attributes. The receiver receives the encrypted data and if the attribute matched, then the original data gets decrypted for the receiver. This allows them to access the data without authorization and thus poses a risk to information privacy.

Before uploading the file to the cloud server it has to encrypt the data along with attributes. Those attributes randomly generated in third party auditor based on the users details. The encrypted file along with attributes is stored in the cloud server. The cloud server acts as a open source application which checks whether the authorized user is retrieving the data or not. The user decrypts the data latter to get the corresponding file.

Limitations

- Users with same attribute and hence unauthorized users can access the data.
- Key updation on user addition/revocation.
- Storage overhead will occur due to more number of keys.

4. PROPOSED WORK

In our proposed system we eliminate all tedious works done in existing system and improve the performance tremendously through high security by three distinguished techniques. Asymmetric encryption is used for encryption and decryption. After encryption we use hashing technique for data verification. If the data gets corrupted, hash value will be changed such that we can know the data has been accessed by unauthorized person. Thus data integrity can be verified using this process. The encrypted file splits into n number of parts so that it can be stored as part of files in various cloud servers. The data will be integrated into single file only when it is requested by the authorized user and then it gets decrypted.

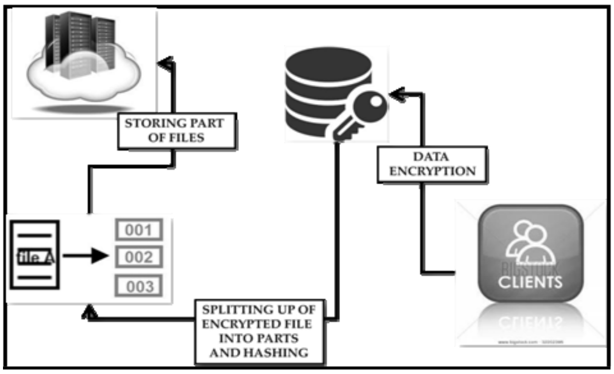


Fig. Architecture Diagram

Implementation of the proposed paper includes the following modules:

- Data Encryption
- Splitting up of Data
- Data Backup
- Storage and Retrieval

Data Encryption

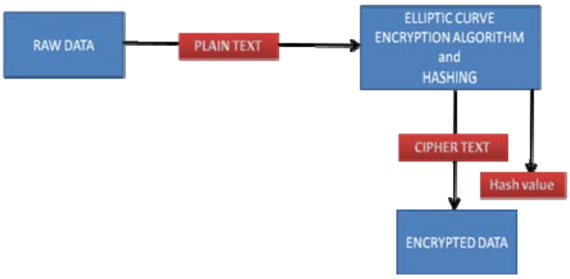
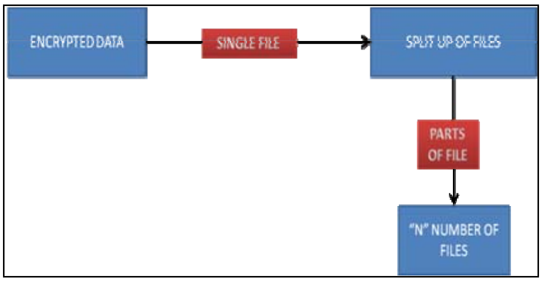


Fig. Data Encryption with ECC

The data will be encrypted with elliptic curve cryptography algorithm. The main purpose of this algorithm is that it is very difficult to understand by unauthorized users. So it is difficult to break the data by the hackers. In ECC(Elliptic Curve Cryptography) point multiplication will be an added advantage which multiplies a point on the curve by a number will produce another point on the curve but it is difficult to find what number was used , even if we know original point and result.

Splitting Up Of Data



Splitting up of data

The encrypted data split into n number of parts based on various different cloud servers. The n value represents the number of cloud servers where each part of file will be stored in different cloud server. Data splitting requires accurate information for integrating the files back. These information about splitting of data will be stored locally to the client system for security reasons.

Data Backup

The files splitted will be stored in n number of servers to secure the data from hackers. This backup data doesn't affect the security because only two different parts of file which cannot be integrated with one another is stored in one particular server. If a server crashes the data recovery will be done through this backup data.

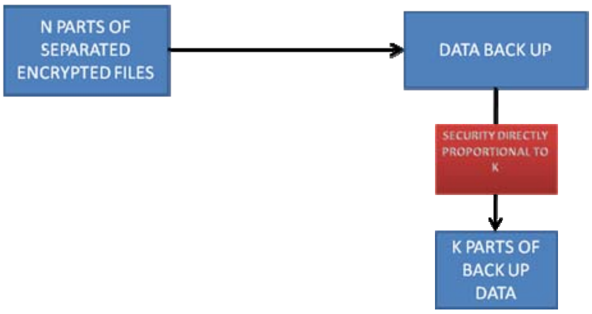

N PARTS OF SEPARATED ENCRYPTED FILES
DATA BACK UP
SECURITY DIRECTLY PROPORTIONAL TO K
K PARTS OF BACK UP DATA

Fig. Data Backup

Storage And Retrieval

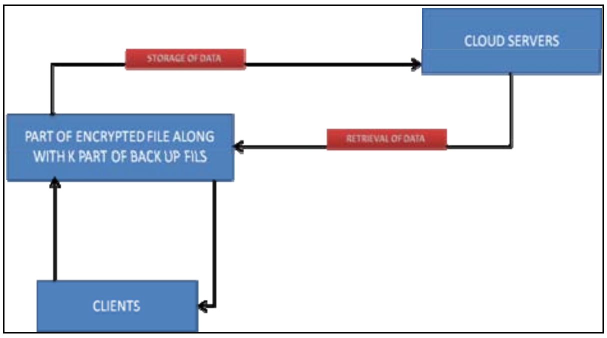


Fig Storage and retrieval of data

In this module, data is ready to store in cloud server. The number of cloud server needed to store file is directly proportional to the security given to the file. The number of backup file needed to retrieve the data is directly proportional to data recovery. After retrieving data these data will be integrated according to the information stored locally. Then the integrated data will be now decrypted to convert them into original data.

5. EXPERIMENTAL STUDY, RESULTS AND DISCUSSION

Description of the Experiments conducted

Experiment 1:

Storing the data in the cloud server (Normal scenario)

Project executed and the data need to be uploaded into the Cloud server has been uploaded in it. Uploaded file has been encrypted and splited and stored across the Cloud servers. Download the uploaded file and it gets decrypted in the client system and file downloaded in it.

Experiment 2:

Stored Data in the cloud has been altered

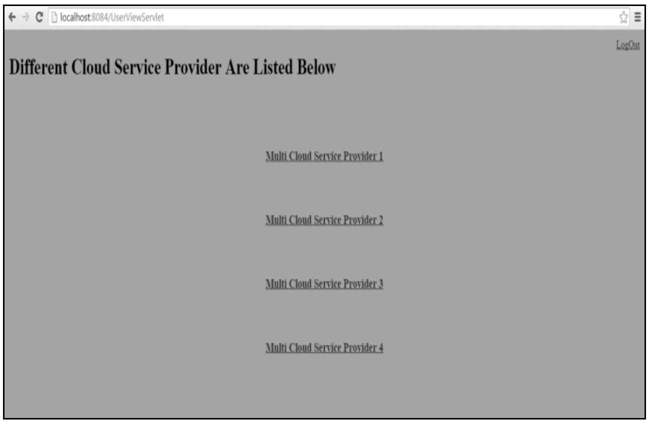
Project executed and data stored in the Cloud server has been altered using any text editor. Now download the data from the Cloud server in the client system, due to file get altered it will not download. Then recover the data from other Cloud server for the exact data which we stored earlier.

Experiment 3:

Cloud server attacked by external entity

Project executed and try to access the data stored in the Cloud server. But the Cloud server has been attacked by external entity. Even though the data has been lost in one cloud server, we can get the actual data from other server where we splited the data and the backup of that data was available in another Cloud server. Thus we can access the actual data which we uploaded earlier.

Different Cloud server’s which are available to provide service.



Different Cloud Service Provider Are Listed Below

Multi Cloud Service Provider 1

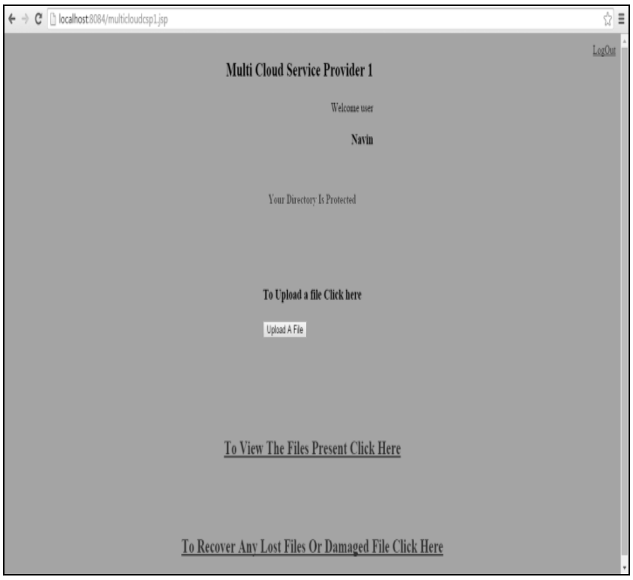
Multi Cloud Service Provider 2

Multi Cloud Service Provider 3

Multi Cloud Service Provider 4

Fig: Different cloud service providers

Client can select the service either to upload the data or to view the data already uploaded in it.



Multi Cloud Service Provider 1

Welcome user

Navin

Your Directory Is Protected

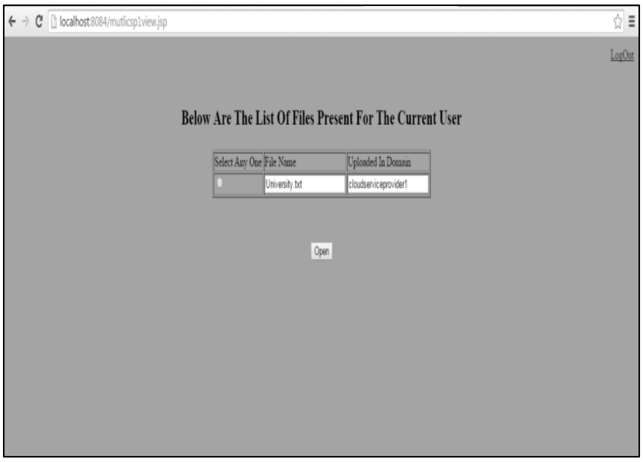
To Upload a file Click here

To View The Files Present Click Here

To Recover Any Lost Files Or Damaged File Click Here

Fig : List of services provided in cloud server

File already uploaded can be viewed. Select the file which is required to be downloaded.

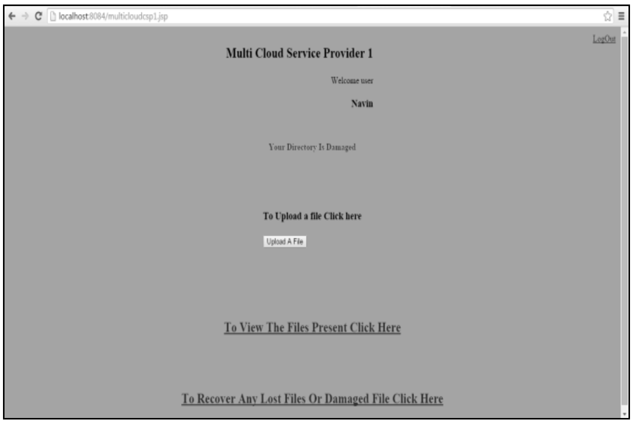


Below Are The List Of Files Present For The Current User

Select Any One	File Name	Uploaded In Domain
○	University.txt	cloudserviceprovider1

Fig : File Uploading

Damaged server will be reported if you try to access that server.



Multi Cloud Service Provider 1

Welcome user

Navin

Your Directory Is Damaged

To Upload a file Click here

To View The Files Present Click Here

To Recover Any Lost Files Or Damaged File Click Here

Fig : Cloud service file system damaged

Retrieving the actual data from Cloud storage due to problems like attack from external entity, data integrity. In such cases of security breech in a Cloud server an effective mechanism can be implemented retrieve the data which was lost. In this project ECC has been used to encrypt the data and hashing value generated for the encrypted data and splitted it into “n” number of parts and stored in the “n” number of cloud server. It also ensures data integrity by using hashing in which it uses an effective measure for finding each and every data stored in a Cloud. All these techniques used here to protect the data from security breech and to retrieve the data in case of Cloud server failure.

6. CONCLUSION

Some set of data’s have high level of confidentiality due to its nature and purpose serving it. For this class of data, security and backup schemes are crucial since disjoint design strategies lead to compromise the data stored in it or data loss due to Cloud server failure/hack. By using the steps encryption with hashing and splitting the encrypted data, It was shown that the proposed design scheme can achieve data integrity there by giving better performance and data security.

FUTURE WORK

Cloud storage space can be further reduced by compressing the data which occupies less data and Security can be further enhanced by blocking the attacker before entering into the Cloud system.

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,” Comm. ACM, vol. 53, no. 4, pp. 50-58, Apr. 2010.

[2] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song, “Provable Data Possession at Untrusted Stores,” Proc. 14th ACM Conf. Computer and Comm. Security (CCS ’07), pp. 598-610, 2007.

[3] B. Chen, R. Curtmola, G. Ateniese, and R. Burns, “Remote Data Checking for Network Coding-Based Distributed Storage Systems,” Proc. ACM Workshop Cloud Computing Security Workshop (CCSW’10), pp. 31-42, 2010.

[4] C. Erway, A. Kupcu, C. Papamanthou, and R. Tamassia, “Dynamic Provable Data Possession,” Proc. 16th ACM Conf. Computer and Comm. Security (CCS’09), pp. 213-222, 2009.

[5] K. Ren, C. Wang, and Q. Wang, “Security Challenges for the Public Cloud,” IEEE Internet Computing, vol. 16, no. 1, pp. 69-73, 2012.

[6] R. Rivest, A. Shamir, and L. Adleman, “A Method for Obtaining Digital Signatures and Public Key Cryptosystems,” Comm. ACM, vol. 21, no. 2, pp. 120-126, 1978.

[7] D. Song, E. Shi, I. Fischer, and U. Shankar, “Cloud Data Protection for the Masses,” Computer, vol. 45, no. 1, pp. 39-45, 2012.

[8] H. Shacham and B. Waters, “Compact Proofs of Retrievability,” Proc. 14th Int’l Conf. Theory and Application of Cryptology and Information Security: Advances in Cryptology (ASIACRYPT ’08), pp. 90- 107, 2008.

[9] The MD5 Message-Digest Algorithm (RFC1321). https://tools. ietf.org/html/rfc1321, 2014.

[10] B. Wang, B. Li, and H. Li, “Oruta: Privacy-Preserving Public Auditing for Shared Data in the Cloud,” Proc. IEEE Fifth Int’l Conf. Cloud Computing, pp. 295-302, 2012.

[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] B. Wang, M. Li, S.S. Chow, and H. Li, “Computing Encrypted Cloud Data Efficiently under Multiple Keys,” Proc. IEEE Conf. Comm. and Network Security (CNS ’13), pp. 90-99, 2013.

[13] Q. Wang, C. Wang, J. Li, K. Ren, and W. Lou, “Enabling Public Verifiability and Data Dynamic for Storage Security in Cloud Computing,” Proc. 14th European Conf. Research in Computer Security (ESORICS’09), pp. 355-370, 2009.

[14] C. Wang, Q. Wang, K. Ren, and W. Lou, “Ensuring Data Storage Security in Cloud Computing,” Proc. 17th Int’l Workshop Quality of Service (IWQoS’09), pp. 1-9, 2009.

[15] Y. Zhu, H. Wang, Z. Hu, G.-J. Ahn, H. Hu, and S.S Yau, “Dynamic Audit Services for Integrity Verification of Outsourced Storages in Clouds,” Proc. ACM Symp. Applied Computing (SAC’11), pp. 1550-1557, 2011.