A Secure Cloud Storage System with Secure Data Forwarding

Aarti P Pimpalkar, Prof. H.A. Hingoliwala

Absract : A cloud storage system, consisting of a group of storage servers, provides long storage services over the net. Storing information in a third party's cloud system causes serious concern over information confidentiality. General coding schemes protect information confidentiality, however also limit the functionality of the storage system as a result of a couple of operations are supported over encrypted information. Constructing a secure storage system that supports multiple functions is difficult once the storage system is distributed and has no central authority. We have a tendency to propose a threshold proxy re-encryption scheme and integrate it with a decentralized erasure code specified a secure distributed storage system is developed. The distributed storage system not only supports secure and strong information storage and retrieval, however also lets a user forward his information within the storage servers to a different user while not retrieving the info back. The most technical contribution is that the proxy re-encryption theme supports encoding operations over encrypted messages still as forwarding operations over encoded and encrypted messages. Our technique absolutely integrates encrypting, encoding, and forwarding. We have a tendency to analyze and counsel appropriate parameters for the amount of copies of a message sent to storage servers and therefore the number of storage servers queried by a key server. These parameters permit additional flexible adjustment between the amount of storage servers and robustness.

Index Terms: Decentralized erasure code, proxy re-encryption, threshold cryptography, secure storage system.

1 INTRODUCTION

As high-speed networks and present net access become out there in recent years, several services area unit provided on the net specified users will use them from anyplace at any time. As an example, the e-mail service is maybe the foremost widespread one. Cloud computing could be a conception that treats the resources on the net as a unified entity, a cloud. Users simply use services while not caring regarding however computation is completed and storage is managed. During this paper, we tend to specialize in coming up with a cloud storage system for lustiness, confidentiality, and functionality. A cloud storage system is taken into account as a large scale distributed storage system that consists of the many freelance storage servers.

Data strength may be a major demand for storage systems. There are several proposals of storing knowledge over storage servers [1], [2], [3], [4], [5]. A technique to produce knowledge strength is to copy a message such every storage server stores a replica of the message. It's terribly sturdy as a result of the message will be retrieved as long jointly storage server survives. Differently is to write in code a message of k symbols into a code word of n symbols by erasure writing. To store a message, every of its code word symbols are keep in an exceedingly completely different storage server. Storage server failure corresponds to associate degree erasure error of the code word image. As long because the range of failure servers is beneath the tolerance threshold of the erasure code, the message will be recovered from the code word symbols keep within the offered storage servers by the decryption method. This provides a trade-off between the storage size and therefore the tolerance threshold of failure servers. A decentralized erasure code is associate degree erasure code that independently computes each code word symbol for a message. Thus, the encoding process for a message can be split into n parallel tasks of generating code word symbols. A decentralized erasure code is appropriate to be used in an exceedingly distributed storage system. When the message symbols area unit sent to storage servers, every storage server severally computes a code- word image for the received message symbols and stores it. This finishes the coding and storing method. The recovery method is that the same.

Storing information in a third party's cloud system causes serious concern on information confidentiality. So as to produce robust confidentiality for messages in storage servers,

- *Aarti P Pimpalkar is currently pursuing Master's Degree program in Computer Engineering in University Of Pune, India, E-mail: aartipimpalkar@gmail.com*
- *Prof. H.A.Hingoliwala(Guide) is currently working as H.O.D. Computer Engineering, JSCOE, Pune, India, E-mail: ali_hyderi@yahoo.com*

a user will encrypt messages by a cryptographic methodology before applying associate erasure code methodology to encode and store messages. Once he needs to use a message, he must retrieve the code word symbols from storage servers, rewrite them, and so rewrite them by mistreatment cryptographic keys. There square measure 3 issues within the on top of simple integration of coding and coding. First, the user should do most computation and the communication traffic between the user and storage servers are high. Second, the user should manage his cryptographic keys. If the user's device of storing the keys is lost or compromised, the safety is broken. Finally, besides information storing and retrieving, it's hard for storage servers to directly support different functions. For instance, storage servers cannot directly forward a user's messages to a different one. The owner of messages should retrieve, decode, and rewrite so forward them to a different user.

In this paper, we have a tendency to address the matter of forwarding information to a different user by storage servers directly below the command of the information owner. We have a tendency to think about the system model that consists of distributed storage servers and key servers. Since storing cryptologic keys in a very single device is risky, a user distributes his cryptologic key to key servers that shall perform cryptographic functions on behalf of the user. These key servers area unit extremely protected by security mechanisms. To well match the distributed structure of systems, we have a tendency to need that servers severally perform all operations. With this thought, we have a tendency to propose a brand new threshold proxy re-encryption theme and integrate it with a secure localized code to make a secure distributed storage system.

The encoding theme supports cryptography operations over encrypted messages and forwarding operations over encrypted and encoded messages. The tight integration of cryptography, encryption, and forwarding makes the storage system expeditiously meet the wants of knowledge robustness, information confidentiality, and information forwarding. Accomplishing the mixing considerately of a distributed structure is difficult. Our system meets the wants that storage servers severally perform cryptography and re encryption and key servers severally perform partial decoding. Moreover, we have a tendency to consider the system in a very a lot of general setting than previous works. This setting permits a lot of versatile adjustment between the quantity of storage servers and robustness.

Our contribution: Assume that there are n distributed storage servers and m key servers within the cloud storage system. A message is split into k blocks and drawn as a vector of k symbols. Our contributions are as follows:

We have a tendency to construct a secure cloud storage system that supports the perform of secure information forwarding by employing a threshold proxy re-encryption theme. The cryptography theme supports decentralized erasure codes over encrypted messages and forwarding operations over encrypted and encoded messages. Our system is extremely distributed wherever storage servers severally encode and forward messages and key servers severally perform partial coding.

2 RELATED WORKS

We concisely review distributed storage systems, proxy re- encryption schemes, and integrity checking mechanisms.

2.1 Distributed Storage Systems

At the early years, the Network-Attached Storage (NAS) [6] and the Network File System (NFS) [7] provide extra storage devices over the network such that a user can access the storage devices via network connection. Afterward, many improvements on scalability, robustness, efficiency, and security were proposed [1], [2]. A decentralized architecture for storage systems offers good scalability, because a storage server can join or leave without control of a central authority. To provide robust- ness against server failures, a simple method is to make replicas of each message and store them in different servers. However, this method is expensive as z replicas result in z times of expansion. One way to reduce the expansion rate is to use erasure codes to encode messages [5]. A message is encoded as a code word, which is a vector of symbols, and each storage server stores a code word symbol. A storage server failure is modeled as an erasure error of the stored code word symbol. Random linear codes support distributed encoding, that is, each code word symbol is independently computed. To store a message of k blocks, each storage server linearly combines the blocks with randomly chosen coefficients and stores the code word symbol and coefficients. To retrieve the message, a user queries k storage servers for the stored code word symbols and coefficients and solves the linear system.

2.2 Proxy Re-Encryption Schemes

Proxy re-encryption schemes are proposed by Mambo and Okamoto and Blaze et al. [8]. During a proxy re-encryption theme, a proxy server will transfer a cipher text under a public key PKA to a new one under another public key PKB by victimization the re-encryption key $RK_{A\text{->}B}$. The server doesn't recognize the plaintext throughout transformation. Ateniese et al. [9] planned some proxy re-encryption schemes and applied them to the sharing operate of secure storage systems. In this scheme, messages are 1st encrypted by the owner then keep during a storage server.

Once a user desires to share his messages, he sends a re-encryption key to the storage server. The storage server re-encrypts the encrypted messages for the approved user. Thus, their system has information confidentiality and supports the information forwarding operates. Our work more integrates encryption, re-encryption, and cryptography specified storage robust- terra firma is reinforced.

Type-based proxy re-encryption schemes planned by Tang[10] provide a stronger graininess on the granted right of a re encryption key. A user can decide which type of messages and with whom he desires to share during this reasonably proxy re-cryptography schemes. Key-private proxy re-encryption schemes square measure planned by Ateniese et al. [11].In a key-private proxy re-encryption theme, given a re-encryption key, a proxy server cannot verify the identity of the recipient. This type of proxy re-encryption schemes provides higher privacy guarantee against proxy servers. Though most proxy re-encryption schemes use pairing operations, there exist proxy re-encryption schemes while not pairing [12].

2.3 Integrity Checking Functionality

Another vital functionality concerning cloud storage is that the perform of integrity checking. When a user stores information into the storage system, he now not possesses the info at hand. The user might want to see whether or not are properly kept in storage servers. The idea of obvious information possession [13] and therefore the notion of proof of storage [14] are planned. Later, public auditability of keep information is addressed in [15]. However all of them consider the messages within the clear text kind.

3 PROGRAMMERS DESIGN:

3.1 System Model

As shown in Fig. 1, our system model consists of users, n storage servers SS1,SS2,...,SSn, and m key servers KS1, KS2,...,KSm. Storage servers give storage services and key servers give key management services. They work severally. Our distributed storage system consists of four phases: system setup, data storage, data forwarding, and data retrieval. These four phases are represented as follows.

Within the system setup section, the system manager chooses system parameters and publishes them. Every user A is assigned a public-secret key try (PKA,SKA). User A distributes his secret key SKA to key servers specified every key server KSi holds a key share SKAi, 1< i < m. The key is shared with a threshold t.

In the data storage phase, user A encrypts his message M and dispatches it to storage servers. A message M is decomposed into k blocks m1,m2,...,mk and has an symbol ID. User A encrypts every block mi into a cipher text Ci and sends it to v randomly chosen storage servers. Upon receiving cipher texts from a user, every storage server linearly combines them

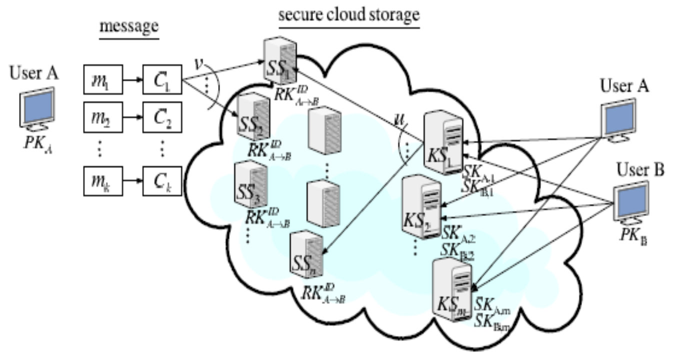

Fig. 1. System Model for Cloud Storage [16]

with randomly chosen coefficients into a code word image and stores it. Note that a storage server could receive but k message blocks and that we assume that each one storage servers know the value k in advance.

In the data forwarding phase, user A forwards his encrypted message with an identifier ID stored in storage servers to user B such that B can decrypt the forwarded message by his secret key. To do so, A uses his secret key SKA and B's public key PKB to reckon a re-encryption key RK^{ID} A->B so sends RK^{ID} A->B to all or any storage servers. Every storage server uses the re-encryption key to re-encrypt its code word image for later retrieval requests by B. There-encrypted code word symbol is the mixture of cipher texts below B's public key. So as to tell apart re-encrypted code word symbols from intact ones, we tend to decision them original code word symbols and re- encrypted code word symbols, severally.

Within the information retrieval section, user A requests to retrieve a message from storage servers. The message is either keeps by him or forwarded to him. User A sends a retrieval request to key servers. Upon receiving the retrieval request and executing a correct authentication method with user A, every key server KSi requests u randomly chosen storage servers to induce code word symbols and will partial decryption on the received code word symbols by using the key share SK_{Ai}. Finally, user A combines the part decrypted code word symbols to get the initial message M.

System recovering. When a storage server fails, a new one is added. The new storage server queries k on the market storage servers linearly combines the received code word symbols as a replacement one and stores it. The system is then recovered.

3.2 Threat Model

We consider information confidentiality for each data storage and data forwarding. During this threat model, associate attacker desires to interrupt information confidentiality of a target user. To do so, the attacker colludes with all storage servers, nontarget users, and up to (t -1) key servers. The attacker analyzes keep messages in storage servers, the key keys of nontarget users, and therefore the shared keys keep in key servers. Note that the storage servers store all re-encryption keys provided by users. The attacker might try and generate a replacement re-encryption key from keep re-encryption keys. We tend to formally model this attack by the quality chosen plaintext attack1 of the proxy re-encryption theme in a very threshold version, as shown in Fig. 2[16]. The rival C provides the system parameters. Once the aggressor A chooses a target user T, the rival provides him (t – 1) key shares of the key SK_T of the target user T to model (t – 1) compromised key servers. Then, the attacker will query secret keys of different users and every one re- encryption keys except those from T to different users. These models compromised nontarget users and storage servers. Within the challenge part, the attacker chooses 2 messages M_0 and M_1 with the identifiers ID_0 and ID_1, respectively. The challenger throws a random coin b and encrypts the message Mb with T's public key PK_T. once obtaining the cipher text from the challenger, the attacker outputs slightly b_0 for guessing b. during this game, the attacker wins if and providing b'= b. The advantage of the attacker is outlined as |½- Pr[b'= b]|.

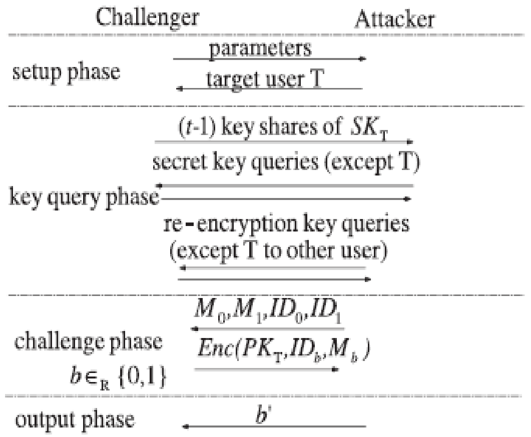

Fig. 2. Security for choosen Plaintext Attack [16]

A cloud storage system model within the above is secure if no probabilistic polynomial time attacker wins the sport with a non-negligible advantage. A secure cloud storage system implies that associate unauthorized user or server cannot get the content of keep messages, and a storage server cannot generate re-encryption keys by himself. If a storage server will generate a re-encryption key from the target user to a different user B, the attacker will win the safety game by re-encrypting the cipher text to B and decrypting the re- encrypted cipher text using the key SK_B. Therefore, this model addresses the security of knowledge storage and data forwarding.

3.3 A Straightforward Solution

A straightforward solution to supporting the data forwarding perform in a very distributed storage system is as follows: once the owner A desires to forward a message to user B, he downloads the encrypted message and decrypts it by using his secret key. He then encrypts the message by using B's public key and uploads the new cipher text. Once B wants to retrieve the forwarded message from A, he downloads the cipher text and decrypts it by his secret key. The complete information forwarding method wants 3 communication rounds for A's downloading and uploading and B's downloading. The communication cost is linear within the length of the forwarded message. The computation cost is that the decryption and encryption for the owner A, and therefore the decryption for user B. Proxy re-encryption schemes will considerably decrease communication and computation price of the owner [16]. In a very proxy re-encryption scheme, the owner's ends are-encryption key to storage servers such that storage servers perform the re-encryption operation for him. Thus, the communication cost of the owner is independent of the length of forwarded message and therefore the computation price of re-encryption is taken care of by storage servers. Proxy re-encryption schemes significantly reduce the overhead of the data forwarding function in a very secure storage system.

4 Constructions of Secure Cloud Storage Systems

We use a threshold proxy re-encryption theme with multiplicative homomorphic property. An encryption theme is multiplicative homomorphic if it supports a bunch operation ϴ on encrypted plaintexts while not decryption

$$D (SK, E (PK, m1) \Theta E (PK, m2)) = m1. m2$$

Where E is that the encryption perform, D is that the decryption perform, and (PK,SK) may be a pair of public key and secret key. Given 2 coefficients g1 and g2, 2 message symbols M1 and m2 will be encoded to a code word symbol $m1^{g1}$ one $m2^{g2}$ within the encrypted type

$$C = E (PK, m1)^{g1} \Theta E (PK, m2)^{g2} = E (PK, m1^{g1.} m2^{g2}),$$

Thus, a multiplicative homomorphic encr1yption theme supports the encoding operation over encrypted messages. We then convert a proxy re-encryption theme with multiplicative homomorphic property into a threshold version. A key's secret is shared to key servers with a threshold value t via the Shamir secret sharing theme, wherever $t \geq k$. In our system, to decrypt for a group of k message symbols, every key server severally queries two storage servers and part decrypts 2 encrypted code word symbols. As long as t key servers are offered, k code word symbols are obtained from the part decrypted cipher texts.

4.1 A Secure Cloud Storage System with Secure Forwarding

As described in Section 3.1, there are four phases of our storage system.

System setup.

The algorithm SetUp(1^T) generates the system parameters μ. A user uses KeyGen(μ) to come up with his public and secret key pair and ShareKeyGen() to share his secret key to a group of m key servers with a threshold t, wherever $k \leq t \leq m$. The user locally stores the third component of his secret key.

Data storage.

When user A desires to store a message of k blocks m1,m2,...,mk with the symbol ID, he computes the identity token T= $h^{f(a3,ID)}$and performs the encryption rule Enc() on T and k blocks to get k original cipher texts C1,C2,...,Ck. a creative cipher text is indicated by a leading bit b = 0. User A sends every cipher text C_i to v arbitrarily chosen storage servers. A storage server receives a group of original cipher texts with a similar identity token T from A. once a cipher text Ci isn't received, the storage server inserts C_i = (0,1,T,1)to the set. The special format of (0,1,T,1)could be a mark for the absence of C_i. The storage server performs Encode() on the set of k cipher texts and stores the encoded result (code word symbol).

Data forwarding.

User A desires to forward a message to a different user B. He desires the first element a1 of his secret key. If A doesn't possess a1, he queries key servers for key shares. Once at least t key servers respond, A recovers the primary element a1 of the secret key SKA via the KeyRecover() algorithm. Let the symbol of the message be ID. User A computes the reencryption key $RK^{ID}_{A\to B}$ via the ReKeyGen() algorithm and firmly sends the re- encryption key to every storage server. By using $RK^{ID}_{A\to B}$, a storage server re-encrypts the first code word image C0 with the symbol ID into a re-encrypted code word image C00 via the ReEnc() rule such C" is decryptable by using B's secret key. A re-encrypted code word image is indicated by the leading bit b=1. Let the public key PKB of user B be ($g^{b1,} h^{b2}$).

Data retrieval.

There are 2 cases for the info retrieval section. The primary case is that a user A retrieves his own message. When user A wants to retrieve the message with the symbol ID, he informs all key servers with the identity token T. A key server 1st retrieves original code word symbols from u arbitrarily chosen storage servers then performs partial decryption ShareDec() on each retrieved original code word symbol C0.The result of partial decryption is termed a partly decrypted code word image. The key server sends the partially decrypted code word symbols and therefore the coefficients to user A. when user A collects replies from at least t key servers and at least k of them are originally from distinct storage servers, he executes Combine() on the t part decrypted code word symbols to recover the blocks m1,m2,...,mk. The second case is that a user B retrieves a message forwarded to him. User B informs all key servers directly. the gathering the collection elements are an equivalent because the 1st case except that key servers retrieve re-encrypted code word symbols and perform partial decryption ShareDec() on reencrypted code word symbols.

5 Methods and Results:

We have implemented following modules according to the given details

5.1 Cloud Data Storage Module

In Admin Module the admin will login to provide his username and password. Then the server setup methodology is often opened. In server setup method the admin 1st set the remote servers Ip-address for send that Ip-address to the receiver. Then the server will skip the method to activate or Dis-activate the method. For activating the process the storage server will show the Ip-address. For Dis-activating the process the storage server cannot show the Ip-address. These details are often viewed by clicking the key server. The activated Ip-addresses are keep in available storage server. By clicking the offered storage server button we will view the presently offered Ip-addresses.

5.2 Data Encryption Module

In cloud login module the user will login his own details. If the user cannot have the account for that cloud system 1st the user will register his details for using and entering into the cloud system. The Registration method details are Username, Email, and password, make sure password, date of birth, gender and additionally the situation. After coming into the registration method the main points may be keep in information of the cloud system. Then the user needs to login to offer his corrected username and password the code needs to be send his/her E-mail. Then the user can move to open his account and consider the code that may be generated from the cloud system.

In transfer Module the new folder is produce for storing the files. In folder creation method the cloud system could raise one question for that user. The user should answer the question and should keep in mind that declares any usage. Then enter the folder name for produce the folder for that user. In file transfer method the user needs to select one file from browsing the system and enter the transfer choice. Now, the server from the cloud will offer the encrypted type of the uploading file.

5.3 Data Forwarding Module

In forward module initial we will see the storage details for the uploaded files. When click the storage details choice we will see the file name, question, answer, folder name, forward price (true or false), forward E-mail. If the forward column show the forwarded price is true the user cannot forward to a different person. If the forward column show the forwarded price is fake the user will forward the file into another person. In file forward processes contains the chosen file name, Email address of the forwarder and enter the code to the forwarder. Now, another user will check his account properly and consider the code forwarded from the previous user. Then this user has login to the cloud system and to see the receive details. In receive details the forwarded file is gift then the user can go to the transfer method.

5.4 Data Retrieval Module

In download module contains the subsequent details. There are username and file name. First, the server method will be run which implies the server will be connected with its specific client. Now, the client needs to download the file to download the file key. In file key downloading method the fields are username, filename, question, answer and also the code. Currently clicking the transfer possibility the client will view the encrypted key. Then using that key the client will read the file and use that file suitably.

5.5 Advantages

Our system provides following advantages,

1) It provides robustness of data, confidentiality and data forwarding using combination of encoding, encryption and forwarding which makes an efficient system.
2) Encoding and re-ecryption are performed independently by storage servers and partial decryption is performed by key servers.
3) It is flexible for the no of storage servers and its robustness.
4) While user login an additional image login will be provided to user to provide an extra security.
5) Instead of blocking a particular user an IP blocking method can be used to provide login to original user.

5 DISCUSSIONS AND CONCLUSION

In this paper, we tend to think about a cloud storage system consists of storage servers and key servers. We integrate a new proposed threshold proxy re-encryption theme and erasure codes over exponents. The threshold proxy re- encryption theme supports encoding, forwarding, and partial decryption operations in a very distributed manner. To decrypt a message of k blocks that are encrypted and encoded to n code word symbols, every key server only must partly decipher 2 code word symbols in our system. By using the threshold proxy re-encryption theme, we present a secure cloud storage system that gives secure information storage and secure data forwarding practicality during a decentralized structure. Moreover, every storage server severally performs encoding and re-encryption and every key server severally perform partial decryption. Our storage system and a few new planned content available file systems and storage system [17] are extremely compatible. Our storage servers act as storage nodes in a very content available storage system for storing content available blocks. Our key servers act as access nodes for providing a front-end layer like a conventional filing system interface. Additional study on careful cooperation is needed.

REFERENCES

[1] J. Kubiatowicz, D. Bindel, Y. Chen, P. Eaton, D. Geels, R. Gummadi, S. Rhea, H. Weatherspoon, W. Weimer, C. Wells, and B. Zhao, "Oceanstore: An Architecture for Global-Scale Persistent Storage," Proc. Ninth Int'l Conf. Architectural Support for Programming Languages and Operating Systems (ASPLOS), pp. 190-201, 2000.
[2] P. Druschel and A. Rowstron, "PAST: A Large-Scale, Persistent Peer-to-Peer Storage Utility," Proc. Eighth Workshop Hot Topics in Operating System (HotOS VIII), pp. 75-80, 2001.
[3] A. Adya, W.J. Bolosky, M. Castro, G. Cermak, R. Chaiken, J.R. Douceur, J. Howell, J.R. Lorch, M. Theimer, and R. Wattenhofer, "Farsite: Federated, Available, and Reliable Storage for an Incompletely Trusted Environment," Proc. Fifth Symp. Operating System Design and Implementation (OSDI), pp. 1-14, 2002.
[4] A. Haeberlen, A. Mislove, and P. Druschel, "Glacier: Highly Durable, Decentralized Storage Despite Massive Correlated Failures," Proc. Second

Symp. Networked Systems Design and Implementation (NSDI), pp. 143-158, 2005.

[5] Z. Wilcox-O'Hearn and B. Warner, "Tahoe: The Least-Authority Filesystem," Proc. Fourth ACM Int'l Workshop Storage Security and Survivability (StorageSS), pp. 21-26, 2008.

[6] D.R. Brownbridge, L.F. Marshall, and B. Randell, "The Newcastle Connection or Unixes of the World Unite!," Software Practice and Experience, vol. 12, no. 12, pp. 1147-1162, 1982.

[7] R. Sandberg, D. Goldberg, S. Kleiman, D. Walsh, and B. Lyon, "Design and Implementation of the Sun Network Filesystem," Proc. USENIX Assoc. Conf., 1985.

[8] M. Blaze, G. Bleumer, and M. Strauss, "Divertible Protocols and Atomic Proxy Cryptography," Proc. Int'l Conf. Theory and Application of Cryptographic Techniques (EUROCRYPT), pp. 127-144, 1998.

[9] G. Ateniese, K. Fu, M. Green, and S. Hohenberger, "Improved Proxy Re-Encryption Schemes with Applications to Secure Distributed Storage," ACM Trans. Information and System Security, vol. 9, no. 1, pp. 1-30, 2006.

[10] Q. Tang, "Type-Based Proxy Re-Encryption and Its Construction," Proc. Ninth Int'l Conf. Cryptology in India: Progress in Cryptology(INDOCRYPT), pp. 130-144, 2008.

[11] G. Ateniese, K. Benson, and S. Hohenberger, "Key-Private Proxy Re-Encryption," Proc. Topics in Cryptology (CT-RSA), pp. 279-294,2009.

[12] J. Shao and Z. Cao, "CCA-Secure Proxy Re-Encryption without Pairings," Proc. 12th Int'l Conf. Practice and Theory in Public Key Cryptography (PKC), pp. 357-376, 2009.

[13] G. Ateniese, R.D. Pietro, L.V. Mancini, and G. Tsudik, "Scalable and Efficient Provable Data Possession," Proc. Fourth Int'l Conf. Security and Privacy in Comm. Netowrks (SecureComm), pp. 1-10, 2008.

[14] K.D. Bowers, A. Juels, and A. Oprea, "HAIL: A High-Availabilityand Integrity Layer for Cloud Storage," Proc. 16th ACM Conf.Computer and Comm. Security (CCS), pp. 187-198, 2009.

[15] C. Wang, Q. Wang, K. Ren, and W. Lou, "Privacy-Preserving Public Auditing for Data Storage Security in Cloud Computing,"Proc. IEEE 29th Int'l Conf. Computer Comm. (INFOCOM), pp. 525-533, 2010.

[16] Hsiao-Ying Lin, and Wen-Guey Tzeng, "A Secure Erasure Code-Based CloudStorage System with Secure Data Forwarding" IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, VOL. 23, NO. 6, JUNE 2012.

[17] C. Ungureanu, B. Atkin, A. Aranya, S. Gokhale, S. Rago, G.Calkowski, C. Dubnicki, and A. Bohra, "Hydrafs: A High-Throughput File System for the Hydrastor Content-Addressable Storage System," Proc. Eighth USENIX Conf. File and Storage Technologies (FAST), p. 17, 2010. "Tradeoffs in Scalable Data Routing for Deduplication Clusters," Proc. Ninth USENIX Conf. File and Storage Technologies (FAST), p. 2,2011.