International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 3 Issue 2, February - 2014

Secured Data Storage in Cloud Computing

Mrs. K. Vidhya[1]
[1]Assistant professor
Department of Computer Science and Engineering
Sri Shakthi Institute of Engineering and Technology
Coimbatore.

Ms. D. Bala Gayathri[2]
[2]PG Scholar
Department of Computer Science and Engineering
Sri Shakthi Institute of Engineering and Technology
Coimbatore.

ABSTRACT

Cloud computing is a model for enabling ubiquitous, convenient, on demand network access to a shared pool of configurable computing resources such as storage, network applications and services that can be rapidly provisioned and released with minimal management effort. Users can enjoy the benefits of cloud computing once the they are sure about their data security. Since users data are stored in remote locations -which is not under direct control and visibility of users or data owners .So ensuring the security at the remote storage is a big challenge. There are variety of algorithms for ensuring security and integrity. Among them RSA for digital signature generation and SHA for hash code generation are used here efficient in public cryptosystems. Here to provide additional security the sentinels added to the data.

Though cloud provides efficient services, the main challenge service providers and customers facing is data security, integrity maintenance, storage maintenance. For the better public auditing users go in need to the Third Party Providers (TPA).By using the multiple TPA's the invalid responses overcomed.

Key Terms: Data Dynamics, Batch Auditing, Cloud Computing, TPA, MAC, HLA

I. INTRODUCTION

In Previous days most of the companies for storing, maintaining the data go for grid services but those grid services does not suitable for the small scale applications.And for the better and faster services people need for the easy way of computing thus the cloud computing was emerged. Cloud is mainly providing better resource allocation and also resource pooling .[1][2][5]For small scale applications cloud is most suitable for good service.

Cloud Computing has been envisioned as the next-generation architecture of IT Enterprise. It move the application software and databases to the centralized large data centers, where the management of the data and services may not be fully trustworthy. This unique paradigm brings about many new security challenges, which have not been well understood. This work studies the problem of ensuring the integrity of data storage in Cloud Computing.

In particular, we consider the task of allowing a third party auditor (TPA), on behalf of the cloud client, to verify the integrity of the dynamic data stored in the cloud. The introduction of TPA eliminates the involvement of client for the auditing of whether his data stored in the cloud is indeed intact, which can be important in achieving economies of scale for Cloud Computing. The support for data dynamics via the most general forms of data operation, such as block modification, insertion and deletion.[1][2][8]

II.SYSTEM MODEL

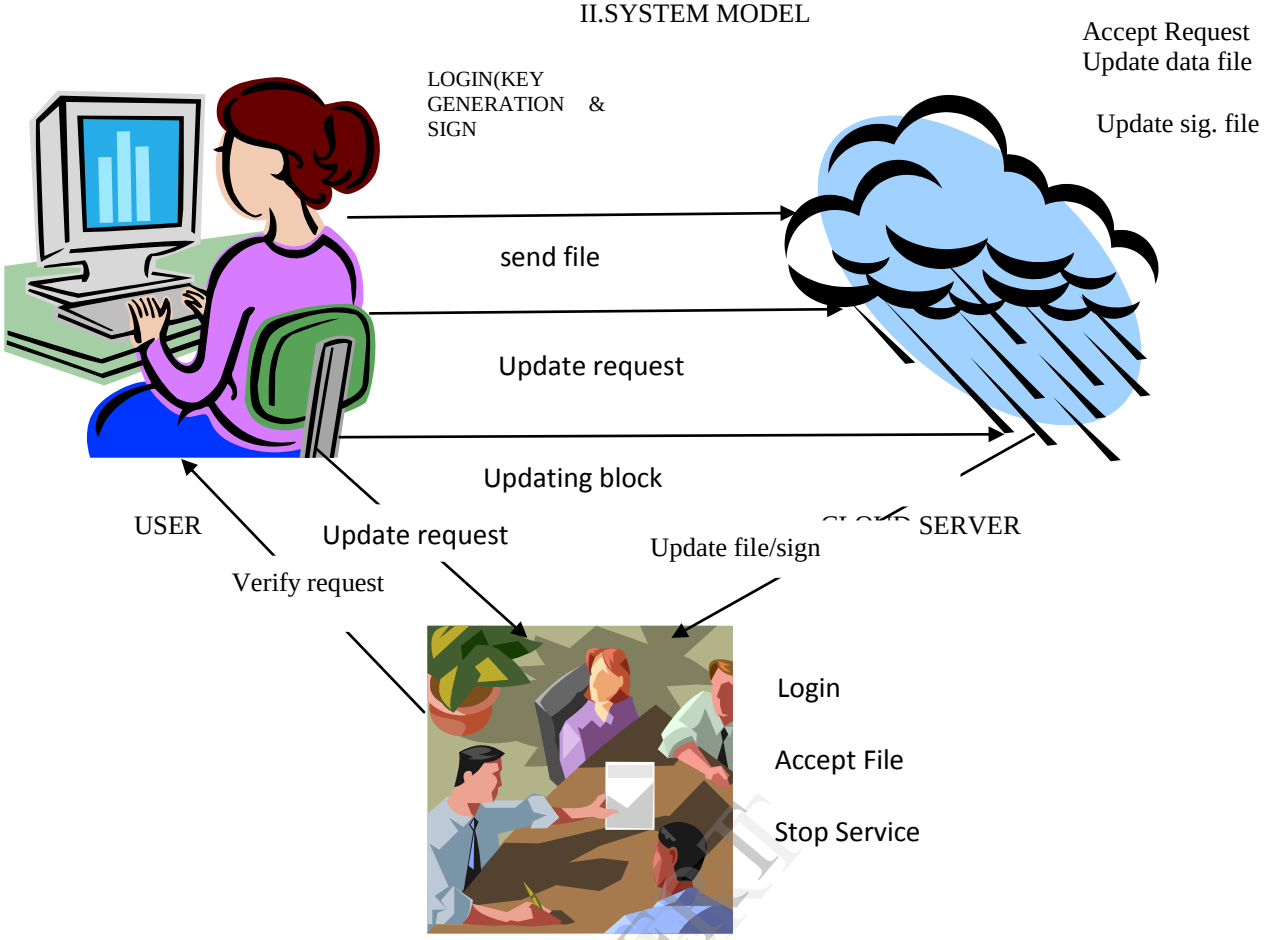


Fig 1:Architecture of cloud data storage service

Here single TPA is found and this provides only single auditing process for the users thus results in increasing the queue length. The cloud user, who has large amount of data files to be stored in the cloud; the cloud server, which is managed by the cloud service provider to provide data storage service and has significant storage space and computation resources third-party auditor, who has expertise and capabilities that cloud users do not have and is trusted to assess the cloud storage service reliability on behalf of the user upon request.

Here, handling data dynamics is serious problem. Data Leakage is happened due to storing the data in single location. The single auditing is done here and therefore it leads to queue waiting.[5][7][2].

In my project there is a general formal model with public verifiability for cloud data storage, where the TPA cannot able to view the users file during the verification and the TPA will be checking only the signature is valid or not.

In my project Data Leakage is solved by storing the data in different location, though if any data is found in any location. Nobody can find the entire data because they are splitted and saved in the public cloud environment.

The operations such as insertion, deletion, append are done in my project and all are done securely that TPA itself cannot able to see any of the files.

III. DESIGN GOALS

1. *Public verification for storage correctness assurance*: to allow anyone, not just the clients who originally stored the file on cloud servers, to have the capability to verify the correctness of the stored data on demand.[1][2]

2. *Dynamic data operation support*: to allow the clients to perform block-level operations on the data files while maintaining the same level of data correctness assurance. The design should be as efficient as possible so as to ensure the seamless integration of public verifiability and dynamic data operation support.

3. Block less verification: no challenged file blocks should be retrieved by the verifier (e.g., TPA) during verification process for both efficiency and security concerns.[1][2]

4. Stateless verification: to eliminate the need for state information maintenance at the verifier side between audits throughout the long term of data storage.[1][2]

SETUP PHASE:
The user initializes the public and secret parameters Of the system by executing KeyGen
and preprocesses the data file F by using SigGen to generate the verification metadata.[1][2] The user then stores the data file F and the verification
metadata at the cloud server and delete its local copy. As part of preprocessing, the user may alter
5. *Multi-User Support by TPA's*.

IV. ALORITHMS USED

Mainly *RSA* and *SHA* algorithms are used.

Key Generation: Run by client
Input: None
Output: public key rpk, secrete key rsk, generator g
Verify Proof: Run by TPA
Input: Proof P
Output: Boolean value {TRUE, FALSE}
Exec Update: Run by the server
Input: file F, set of signature Φ, update query
Output: new file F, new set of signature Φ, update proof.
Verify Update: Run by the client.
Input : public key, update query, update proof
Output: Boolean value TRUE, FALSE, and signature H(R').
Input: File Blocks F, secret key rsk, generator g.
Output: set of signature Φ.
Generate Proof: Run by cloud storage server
Input: Subset of file blocks mi, coefficient i
Output: Proof P

V. PHASES :

There are mainly two phases in this paper and they are
SETUP PHASE
AUDIT PHASE

There are two possible ways to make use of MAC
To authenticate the data[1].A trivial way is just
Uploading the data blocks with their MACs to the
Server and sends the corresponding secret key sk
to the TPA. Later, the TPA can randomly retrieve
blocks with their MACs and check the correctness via sk.
HOMOMORPHIC LINEAR AUTHENTICATOR:
HLA effectively support public auditability
Without having to retrieve the data blocks themselves, HLA authenticate the integrity of data
block and HLAs can also be aggregated[1][2].

VII. SECURE DATA STORAGE

The secure cloud storage is achieved in this paper by adding sentinels, that is small data fragment can be found to be get added with the normal data and this will enables the point of retrievability.

the data file F by expanding it or including
additional metadata to be stored at the server.

AUDIT PHASE:
The TPA issues an audit message or challenge to the cloud server to make sure that the cloud server has retained the data file F properly
at the time of the audit. The cloud server will derive a response message by executing GenProof using F and its verification metadata as inputs. The TPA then verifies the response via VerifyProof[1]

VI. BASIC SCHEMES USED

There are two basic schemes are used in Privacy preserving. They are MAC(Message Authentication Code),HLA(Homomorphic Linear
Authenticator)
MESSAGE AUTHENTICATION CODE:

Here, normally the data are stored in by separating them in various block and stored in various region in cloud storage.

The sentinels are added by the user with their original data and they are
encrypted[1][2]stored in the multiple locations of the cloud server.

The POR scheme uses special blocks(called sentinels)hidden among other in the data. During the verification phase the client asks for randomly picked sentinels and checks whether they are intact. If the server modifies or deletes parts of the data, then sentinels would also be affected with a certain probability [4][8][9].However, sentinels should be indistinguishable from other regular blocks this implies that blocks must be encrypted.

In case if the sentinels are revealed to the server that sentinels never be used again in the database and also to the blocks. This will increase the POR better than compare to the previous methodology.

X. PERFORMANCE ANALYSIS

In the existing paper, the invalid responses are more during batchauditing so the auditing process is found to be get more affected because there is single auditor performing multiple
delegations and they can't perform better auditing
process though batch auditing is achieved[1][2].

The below figure states that the thought the auditing process is performed the invalid responses is reaching the time of individual process.

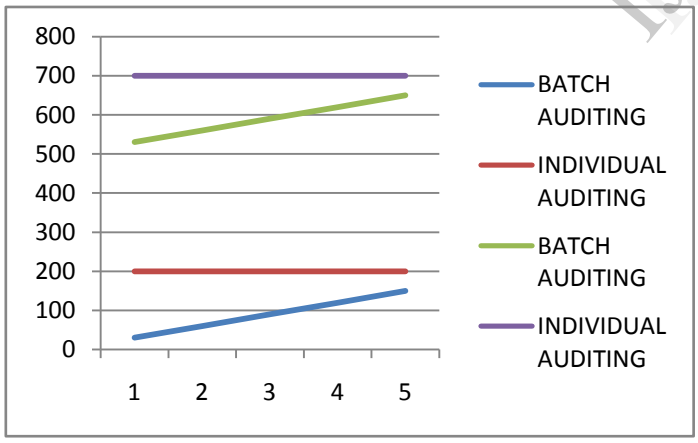
VIII. BATCH AUDITING

With the establishment of privacy-preserving public auditing, the TPA may concurrently handle multiple auditing upon different users' delegation. The individual auditing of these tasks for the TPA can be tedious and very inefficient. Given K auditing delegations on K distinct data files from K different users, it is more advantageous for the TPA to batch these multiple
tasks together and audit at one time. Keeping this natural demand in mind, we slightly modify the protocol in a single user case, and achieves the aggregation of K verification equations (for K auditing tasks) into a single one.[1][2] As a result, a secure batch auditing protocol for simultaneous auditing of multiple tasks is obtained.

IX. DATA DYNAMICS

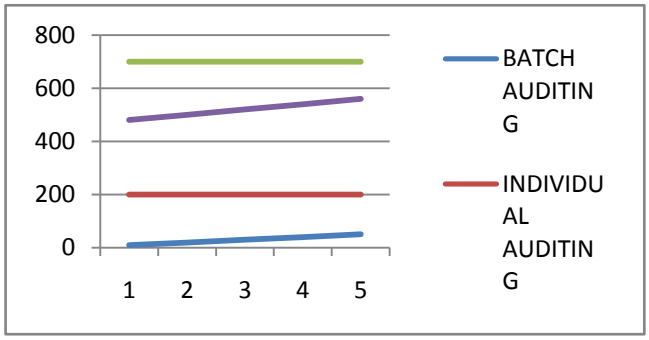
In cloud computing, outsourced data might not only be accessed but also updated frequently by users for various application purposes Hence, supporting data dynamics for privacy-preserving public auditing is also of paramount importance. The data dynamics including block level operations of modification, deletion, and insertion.

In data dynamics support is achieved by replacing the index information i with mi in the computation of block authenticators and using the classic data structure Merkle hash tree (MHT) for the underlying block sequence enforcement and this achieves the privacy preserving public auditing.[1][2]



x-axis: Fraction of invalid responses
y-axis: Auditing time in ms

Fig 2: performance of individual and batch auditing process



x-axis: Fraction of invalid responses
y-axis: Auditing time in ms

Fig 3: performance of individual and batch auditing process

The above graph illustrates that invalid response are found to be get minimized by implementing multiple TPA's with multitasking and therefore the are many auditors for auditing purposes so the auditing process is increased and thereby the invalid responses is minimized. This is achieved in this paper.

XI.CALCULATION FOR VERIFICATION PROCESS:

For authentication purpose the server generates the following equation for the verification purpose of the users

$$\sigma = \prod_{i \in I} \sigma_i^{v_i}$$

The server sends the authentication report to the TPA for the verification is done with the equation

$$R \,.\, e(\sigma^r, g) = e\left(\left(\prod_{i=s_1}^{s_e} H(W_i) v_i\right) . u^{\mu}, v\right)$$

The above equation can be illustrated as follows :

$$R \,.\, e(\sigma^r, g) = e(u,v)^r . e\left(\left(\prod_{i=s_1}^{s_e} (H(W_i) . u^{m_i}) x . {}^{v_i}\right) \gamma, g\right)$$

$$= e(u,v)^r . e\left(\left(\prod_{i=s_1}^{s_e} (H(W_i) . u^{v_i} m_i) x . {}^{v_i}\right) \gamma, g\right)^x$$

$$= e(u,v)^r . e\left(\left(\prod_{i=s_1}^{s_e} (H(W_i) . u^{v_i}\right) . u^{u' r}, v\right)$$

$$= e\left(\left(\prod_{i=s_1}^{s_e} (H(W_i) . u^{v_i}\right)^r . u^{u' r + r}, v\right)$$

$$= e\left(\left(\prod_{i=s_1}^{s_e} (H(W_i) . u^{v_i}\right)^r . u^{u}, v\right)$$

Where the

$H(W_i)$ – Hash value generated for the verification purpose.
σ - summation value generated for the verification.
G – key generation
S_i – No. of blocks

The server check the proof with the above equation by generating the hash values and then sends the result to the TPA for verification .The TPA verifies the with the help of below equation

$$R_1 \ldots R_s \,.\, e(\sigma^{\gamma}, g) = e\left(\left(\prod_{i=s_1}^{s_e} H(W_i) v_i\right)^{\gamma} . \prod_{j=1}^{s} u^{\mu^j}{}_j, v\right)$$

For the effective verification the TPA computes the following equations

$$R.e\left(\prod_{k=1}^{k} \sigma^{\gamma k}{}_k, g\right)$$

$$= \prod_{k=1}^{k} e\left(\left(\prod_{i=s_1}^{s_e} H(W_{k,i}) v^i\right) \gamma k \,.\, \mu_k^{\mu k}, vk\right)$$

To conform the verification process, the server generated value and the TPA generated value
are computed by evaluating the LHS and RHS value of the above equation.

$$LHS = R_1 \,.\, R_2 \ldots R_k \,.\, \prod_{k=1}^{k} e(\sigma_k^{\gamma^k}, g)$$

$= \prod_{k=1}^{k} R_k.e(\sigma_k^{\gamma k},g)$

$=\prod_{k=1}^{k} e((\prod_{i=s_1}^{s_e} H(w_{k,i})v^i)\gamma^k.\ u_k^{\mu k}.vk)$

The extractor that is cloud user is giving one challenge to the cloud server for retrieving the data from the cloud. For that, the extractor verifies the following equation

$\mathbf{R\ .\ e(\sigma^{\gamma}, g)=e((\prod_{i=s_1}^{s_e} H(w_i)^{\vartheta_i})^{\gamma}.u^{\mu},v)}$

The below equation for the n number of users

$\mathbf{R\ .\ e(\sigma^{\gamma}, g)=e((\prod_{i=s_1}^{s_e} H(w_i)^{\vartheta_i})^{\gamma}.u^{\mu}\ v)}$

Dividing the above equation as

$(\sum_{i=s_1}^{s_e} m_i\, v_i) = (\mu - \mu^*)/(\gamma - \gamma^*)$

Thus the challenge is solved by evaluating the above equations.

XII RSA THE PUBLIC KEY CRYPTOSYSTEMS

Deciphering an enciphered message gives you the original message

$D(E(M)) = M$

For encrypting the message

$E(D(M)) = M$

The encryption and decryption can be done with the below equations

$C \equiv E(M) \equiv Me\ (mod\ n)$

$M \equiv D(C) \equiv Cd\ (mod\ n)$

Now we want to obtain the appropriate e and d. We pick d to be a random large integer, which must be coprime to (p - 1) (q - 1), meaning the following equation has to be satisfied:

$$gcd(d, (p - 1) (q - 1)) = 1$$

We will want to compute e from d, p, and q, where e is the multiplicative inverse of d. That means we need to satisfy

$e\ .\ d = 1\ (mod\ \emptyset(n))$ ----------(i)

$\emptyset\ (n) = \emptyset\ (p)\ .\ \emptyset\ (q)$

$= (p\ .\ 1)\ \_\ (q\ .\ 1)$

$= n - (p + q) + 1$

Then we substitute $\emptyset(n)$ value to the (i) eqn

$e\ .\ d \equiv 1\ (mod\ \emptyset\ (n))$

which is equivalent to

$e\ .\ d = k\ .\ \emptyset\ (n) + 1$

Thus we safely can assure that

$D(E(M)) \equiv (E(M))d \equiv\ (Me)d\ (mod\ n) = Me\_d(mod\ n)$

$E(D(M)) \equiv (D(M))e \equiv (Md)e\ (mod\ n)$

$= Me\_d\ (mod\ n)$

Proof of correctness:

Proof using Fermat's little theorem

The proof of the correctness of RSA is based on Fermat's little theorem. This theorem states that if p is prime and p does not divide an integer a then

$A^{(p-1)} \equiv 1\ (mod\ p)$

We want to show that $(m^e)^d \equiv m\ (mod\ pq)$ for every integer m when p and q are distinct prime numbers and e and d are positive integers satisfying

$ed \equiv 1\ (mod\ (p - 1)\ (q - 1))$

We can write

$ed\ - 1 \equiv h(p - 1)(q - 1)$

for some nonnegative integer h.

XIII. SIGNATURE GENERATION USING SHA ALGORITHM

The signature of a message M is the pair of numbers r and s computed according to the equations below:

$r = g^k\ mod\ p)\ mod\ q$

$s = (k^{-1}\ (SHA\text{-}1(M)+xr))\ mod\ q$

For verification through the signature the following calculations are done

$w = (s')^{-1}\ mod\ q$

$u1 = ((SHA\text{-}192(M'))\ w)\ mod\ q$

$u2 = ((r')\ w)\ mod\ q$

$v = (((g)^{u1}\ (y)^{u2})\ mod\ p)\ mod\ q$

XIV COMPARISON GRAPH FOR AUDITING BY TPA

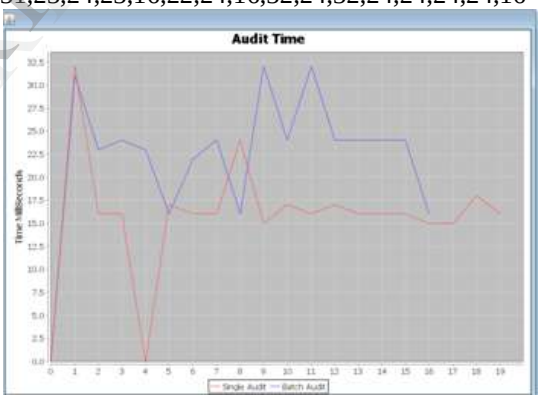
The graph stated below is for the comparison for the single auditing and multiple auditing .

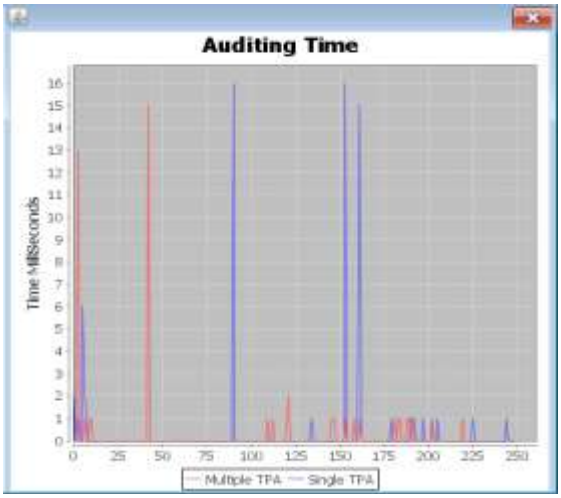
The values for single auditing as follows

32,16,16,0,17,16,16,24,15,17,16,16,16,15,15,18,16

The values for the batch auditing as follows

31,23,24,23,16,22,24,16,32,24,32,24,24,24,24,16





XVI. RELATED WORK

Ateniese et al [4]are the first to consider public auditability in their "provable data possession"(PDP)model for ensuring possession of data files on untrusted storages. They utilize the RSA-based homomorphic linear authenticators for
auditing outsourced data and suggest randomly sampling a few blocks of the file. However, among
their two proposed schemes, the one with public auditability exposes the linear combination of sampled blocks to external auditor. When used
directly ,their protocol is not provably privacy preserving and thus may leak user data information to the external auditor. proof of retrievability[8][9]
(POR) model, where spot-checking and error-correcting codes are used to ensure both
"possession" and "retrievability" of data files on remote archieve service systems storage efficiently.

XVII. CONCLUSION

In this paper, we propose a privacy-preserving public auditing system for data storage security in cloud computing using sentinels. We utilize the homomorphic linear authenticator and random masking to guarantee that the TPA would not learn any knowledge about the data content stored on the cloud server during the efficient auditing process, which not only eliminates the burden of cloud user from the tedious and possibly expensive auditing task, but also alleviates the users' fear of their outsourced data leakage. Considering TPA may concurrently handle multiple audit sessions from different users for their outsourced data files and the invalid responses are also minimized here. Extensive analysis shows that our schemes are provably secure and highly efficient. Our preliminary experiment conducted on ASPOSE instance further demonstrates the fast performance of our design on both the cloud and the auditor side. We leave the full-fledged implementation of the mechanism on commercial public cloud as an important future extension, which is expected to robustly cope with very large scale data.

XVIII. REFERENCE PAPERS

1. C.Wang, Q.Wang ,K.Ren and W.Lou, "Privacy-Preserving Public Auditing for storage Security in Cloud Computing," Proc. IEEE INFOCOM '10, Mar. 2010.
2. 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 and Distributed Systems, vol. 3. 3)T.Schwarz and E.L.Miller,"store,Forget,and Check:Using Algebraic Signatures to check Remotely Administered Storage," Proc.IEEE int'l Conf.Distributed computing Systems(ICDCS'06), 2006.
4. 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.
5. R. Curtmola, O. Khan, R. Burns, and G. Ateniese, "MR-PDP: Multiple-Replica
6. C. Wang, K. Ren, W. Lou, and J. Li, "Towards Publicly Auditable Secure Cloud Data Storage Services," IEEE Network Magazine, vol. 24, no. 4, pp. 19-24, July/Aug. 2010.
7. D. Boneh, B. Lynn, and H. Shacham, "Short Signatures from the Weil Pairing," J. Cryptology, vol. 17, no. 4, pp. 297-319, 2004.
8. G. Ateniese, R.D. Pietro, L.V. Mancini, and G. Tsudik, "Scalable and Efficient Provable Data Possession," Proc. Int'l Conf. Security and Privacy in Comm. Networks (SecureComm '08), pp. 1-10, 2008.
9. C. Erway, A. Kupcu, C. Papamanthou, and R. Tamassia, "Dynamic Provable Data Possession," Proc. ACM Conf. Computer and Comm. Security (CCS '09), pp. 213-222, 2009.