International Journal on Applications in Information and Communication Engineering
Volume 3: Issue3 : December 2017, pp 17 -22 www.aetsjournal.com *ISSN (Online) : 2394 - 6237*

An Analysis of Cloud Data Storage Integrity Through Internal Auditing

R.Hariharan, Dr. G.Komarasamy

Abstract — Cloud Computing is one of the emerging and recent technologies nowadays for sharing resources. Most commonly used for multiple purposes like storage huge data, independent access of data, fast retrieval of data and resources. Some of the security issues may be raised when we access the data in online. To overcome these security related issues we require a new public authentication method called TPA (Third Party Auditor). To securely use the TPA data assessment, user can check the data integrity without the need of local data storage. This method will not bring additional burden to the cloud users. In this paper we propose a batch auditing to sole the security related issues. This is the best method to handle the cloud data most effectively and efficiently.

Keywords— Data storage, privacy-preserving, public authentication, cloud computing, TPA.

I. INTRODUCTION

Cloud computing has been envisioned as the next generation information technology (IT) architecture for enterprises, due to its long list of unprecedented advantages in the IT history: on-demand self-service, ubiquitous network access, location independent re- source pooling, rapid resource elasticity, usage-based pricing and transference of risk. As a disruptive technology with profound implications, Cloud Computing is transforming the very nature of how businesses use information technology. One fundamental aspect of this paradigm shifting is that data is being centralized or outsourced to the Cloud. From users' perspective, including both individuals and IT enterprises, storing data remotely to the cloud in a flexible on-demand manner brings appealing benefits: relief of the burden for storage management, universal data access with location independence, and avoidance of capital expenditure on hardware, software, and personnel maintenances, etc. While Cloud Computing makes these advantages more appealing than ever, it also brings new and challenging security threats towards users' outsourced data.

Since cloud service providers (CSP) are separate administrative entities, data outsourcing is actually relinquishing user's ultimate control over the fate of their data. As a result, the correctness of the data in the cloud is being put at risk due to the following reasons. First of all, although the infrastructures under the cloud are much more powerful and reliable than personal computing devices, they are still facing the broad range of both internal and external threats for data integrity. Examples of outages and security breaches of noteworthy

cloud services appear from time to time. Secondly, there do exist various motivations for CSP to behave unfaithfully towards the cloud users regarding their outsourced data status. For examples, CSP might reclaim storage for monetary reasons by discarding data that has not been or is rarely accessed, or even hide data loss incidents to maintain a reputation. In short, although outsourcing data to the cloud is economically attractive for long-term large-scale storage, it does not immediately offer any guarantee on data integrity and availability. This problem, if not properly addressed, may impede the success of cloud architecture. As users no longer physically possess the storage of their data, traditional cryptographic primitives for the purpose of data security protection cannot be directly adopted. In particular, simply downloading all the data for its integrity verification is not a practical solution due to the expensiveness in I/O and transmission cost across the network. Moreover, the overhead of using cloud storage should be minimized as much as possible, such that a user does not need to perform too many operations to use the data (in additional to retrieving the data). In particular, users may not want to go through the complexity in verifying the data integrity. Besides, there may be more than one user accesses the same cloud storage, say in an enterprise setting. For easier management, it is desirable that cloud only entertains verification request from a single designated party. To fully ensure the data integrity and save the cloud users'

R.Hariharan, Research scholar, Anna University (Email Id : *hariharanrcse@gmail.com)*

Dr. G.Komarasamy , Assistant Professor (Senior Grade) Department of CSE , Bannari Amman Institute of Technology, India. (Email Id : gkomarasamy@gmail.com)

computation resources as well as online burden, it is of critical importance to enable public auditing service for cloud data storage, so that users may resort to an independent third party auditor (TPA) to audit the outsourced data when needed. The TPA, who has expertise and capabilities that users do not, can periodically check the integrity of all the data stored in the cloud on behalf of the users, which provides a much more easier and affordable way for the users to ensure their storage correctness in the cloud. Moreover, in addition to help users to evaluate the risk of their subscribed cloud data services, the audit result from TPA would also be beneficial for the cloud service providers to improve their cloud based service platform, and even serve for independent arbitration purposes. In a word, enabling public auditing services will play an important role for this nascent cloud economy to become fully established; where users will need ways to assess risk and gain trust in the cloud. Therefore, how to enable a privacy-preserving third party auditing protocol, independent to data encryption, is the problem we are going to tackle in this paper. Our work is among the first few ones to support privacy preserving public auditing in Cloud Computing, with a focus on data storage. Besides, with the prevalence of Cloud Computing, a foreseeable increase of auditing tasks from different users may be delegated to TPA. As the individual auditing of these growing tasks can be tedious and cumbersome, a natural demand is then how to enable the TPA to efficiently perform multiple To address these problems, our work utilizes the technique of public key based homomorphism linear authenticator (or HLA for short), which enables TPA to perform the auditing without demanding the local copy of data and thus drastically reduces the communication and computation overhead as compared to the straightforward data auditing approaches.

II. PROBLEM STATEMENT

A. The System and Threat Model

We consider a cloud data storage service involving three different entities, as illustrated in Fig. 1: the cloud user, who has large amount of data files to be stored in the cloud; the cloud server (CS), which is managed by the cloud service provider (CSP) to provide data storage service and has significant storage space and computation resources (we will not differentiate CS and CSP hereafter); the third party auditor (TPA), 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. Users rely on the CS for cloud data storage and maintenance. They may also dynamically interact with the CS to access and update their stored data for various application purposes. As users no longer possess their data locally, it is of critical importance for users to ensure that their data are being correctly stored and maintained. To save the computation resource as well as the online burden potentially brought by the periodic storage correctness verification, cloud users may resort to TPA for ensuring the storage integrity of their outsourced data, while hoping to keep their data private from TPA. We assume the data integrity threats towards users' data can come from both internal and external attacks at CS. These may include: software bugs, hardware failures, bugs in the network path, economically motivated hackers, malicious or accidental management errors, etc. Besides, On the one hand, there are regulationsTherefore, we assume that neither CS nor TPA has motivations to collude with each other during the auditing process. In other words, neither entities will deviate from the prescribed protocol execution in the following presentation. To authorize the CS to respond to the audit delegated to TPA's, the user can issue a certificate on TPA's public key, and all audits from the TPA are authenticated against such a certificate.

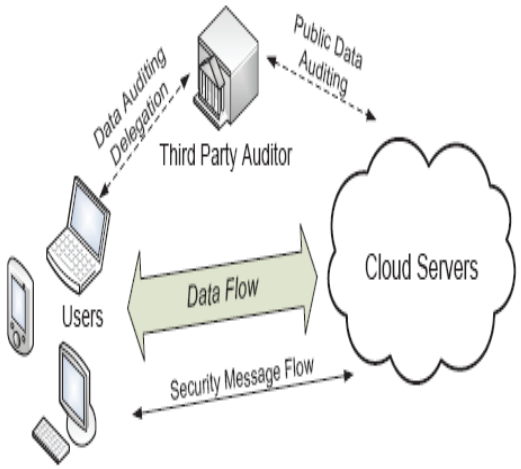


Fig. 1: The architecture of cloud data storage service

B. Design Goals

To enable privacy-preserving public auditing for cloud data storage under the aforementioned model, our protocol design should achieve the following security and performance guarantees. 1) Public auditability: to allow TPA to verify the correctness of the cloud data on demand without retrieving a copy of the whole data or introducing additional online burden to the cloud users. 2) Storage correctness: to ensure that there exists no cheating cloud server that can pass the TPA's audit

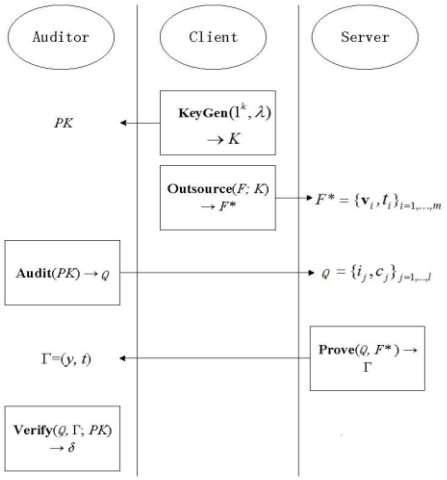
without indeed storing users' data intact. 3) Privacy-preserving: to ensure that the TPA cannot derive users' data content from the information collected during the auditing process. 4) Batch auditing: to enable TPA with secure and efficient auditing capability to cope with multiple auditing delegations from possibly large number of different users simultaneously. 5) Lightweight: to allow TPA to perform auditing with minimum communication and computation overhead.

III. THE PROPOSED SCHEMES

In this section, we firstly introduce a cloud storage auditing protocol, which is based on a recent homomorphic network coding signature technique under the strong RSA assumption [4] [6], and then extend and enhance it to support privacy preserving public-auditing and dynamic data operations.

A. Basic Scheme for Integrity Verification

As illustrated in Fig. 2, we show this basic auditing protocol by filling all the semantics of a cloud storage auditing protocol CSA = (KeyGen, Outsource, Audit, Prove, Verify) according to Definition 2. Details are as follows:



- KeyGen(1k,) ! K: The client generates two random (safe) primes p, q of length k=2 each and then sets N = pq. In addition to k, here we assume an additional security parameter to generate the file identifier e, a prime number of (exactly) + 1 bits, greater than 2 . Then the client determines the block length n and the total number of blocks m. The client also chooses g; g1; : : : ; gn; h1; : : : ; hm at random (in Z N). The public key is PK = (N; e; g; g1; : : : ; gn; h1; : : : ; hm) and the secret key is SK = (p; q). Denote the key by K = (SK; PK).
- Outsource(F; K) ! F_: On input the data F to be outsourced, the client divides F into a collection of vectors fvi = [vi1; : : : ; vin]gi=1;2;:::;m in Zn e . For each vi, compute its signature as follows. First, generate a random integer si 2 Ze uniformly. Use the Chinese remainder theorem to calculate xi 2 ZN by solving

xe i = gsi (n Yj=1gvj). hi mod N

B. Support for Privacy-Preserving Public Auditing

In this section, we show how to enhance the basic protocol in Subsection A to obtain an advanced protocol which supports third party auditing. It is desired that the third party auditor should not know the client's data, but at the same time the third party auditor can indeed verify the integrity of the client's data in the cloud server. In fact, only two algorithms CSA.Outsource and CSA.Prove in our protocol CSA = (KeyGen, Outsource, Audit, Prove, Verify) need to be enhanced.

C. Support for Dynamic Data Operation with Integrity Assurance

Without employing index tables [14], we design a new data dynamic protocol based on a public key updating system, which excludes index information from the signatures and uses the unique structure of the public key to authenticate the positions of data blocks. Now we show how our scheme can explicitly and efficiently handle fully dynamic data operations for cloud data storage.

Step 1: Data Update.

There are three types of data update

operations that can be used by the client: data modification (M), data insertion (I), and data deletion (D).

- Data Modification: Suppose the client wants to modify the i-th block vi to v0i . At start, based on the new block v0i, the client generates the corresponding signature t0i. Specifically, the client (1) generates a new random number h0i in Z N; (2) generates a random number s0i in Ze; (3) gets x0i 2 ZN by solving the equation x0i e = gs0i (Qn j=1 g v0j j) h0i mod N. The signature for

v0i is t0i = (s0i; x0i). Note that the public key updates from PK = (N; e; g; g1; : : : ; gn; h1; : : : ; hm) to PK0 = (N; e; g; g1; : : : ; gn; h1; : : : ; hi□1; h0i; : : : ; hm).

Then, the client constructs an update request message "update = (M; i; v0i; t0i)" and sends it to the server, where M denotes the modification operation. Upon receiving the request, the server runs ExecUpdate(F ; update), i.e.,

the sever replaces the pair (vi; ti) with (v0i; t0i). Finally, the server responses the client with a feedback for this operation is done, UpdateDone(M; i;m), where m is the total number of data blocks stored in the server after the operation, and in this case, m = m. Upon receiving the feedback, the client sends an update notification update = (M; i; h0i;m) to the auditor.

Step 2: Update Verification.

In order to guarantee the server updates the client's data with correct value and position, it's necessary to verify the update operation by the auditor.

Note that every block has a uniquely corresponding hi 2 ZN, and whenever there is a dynamic operation, the public key PK updates correspondingly. We find that we can use the default verification algorithm "Verify(q, □; PK) ! _" to verify the update operation, which means the update verification can be part of the integrity verification. The proof will be given in Section IV.

D. Hierarchical Cloud Structure

Firstly, hierarchical cloud architecture is given according to the different types of provided services. Upon each layer, we organize the nodes in a proper sequence and build up the BAT structure to assist location and recovery protocols. Service layer. HCSP performs as an honest organizer of multicloud.

During verification, it aggregates the proofs from all clouds and responses the final proof to the designated auditors. In the recovery procedure, HCSP assists owners to locate corruptions and recover with the lowest cost.

Computation layer. This layer relies on the high computation capacity of multi-cloud. We can divide this layer into K levels. Each node on k level computes an aggregation of sub proofs on k 1 level − recursively, till the nodes on the top level send the aggregated proofs to the HCSP. We use (σ(k),ϑ(k))to represent the proof of a node on k level . Storage layer. This layer is consisted of physical storage nodes, which are leaf nodes under the structure of BAT. This layer offers an interface of data manipulation while it does not participate in computing.

E. The Basic Protocol

The basic protocol consists of nine phases: GlobeSetup, UserSetup, OffTagGen, OnTagGen, Audit, Batch Audit, Modification, Insertion and Deletion. In the first phase (GlobeSetup), the TA generates the system global parameter.

In the second phase (UserSetup), a user's public private key pair and the corresponding certificate are generated. In the third phase (OffTagGen), a pool of offline tags are generated by the user through calculation in the background. Those tags will be used in the next phase. In the fourth phase (OnTagGen), the user generates the online tags based on the offline tags when the data file to be outsourced is available. In the fifth phase (Audit), the TPA launches the public auditing task by sending a challenge message to the CSP. The CSP will generate a response and send it to the TPA. Finally, the TPA verifies the response to check whether the data is intact. The sixth phase (Batch Audit) considers the case when the TPA receives multiple auditing tasks. The last three phases (Modification, Insertion and Deletion) allow the user to modify, insert and delete his file respectively. Our protocol is as follows:

- GlobeSetup: On input a security parameter , the TA generates multiplicative cyclic groups G1, G2 and GT of prime order p, and a bilinear map e : G1 G2 ! GT , selects a generator g 2 G1 and a generator h 2 G2, chooses a hash function H2 : GT ! Zp; chooses a secure signature scheme Sigprivate key()=V erpublic key(); generates a private-public key pair (msk; mpk). The system global parameter is param = (e;G1;G2; g; h; Sigprivate key()=V erpublic key();H2; mpk).
- UserSetup: A user Ul randomly chooses xl; yl 2 Zp and computes ql = gyl and dl = hxl . Ul also generates a private-public key pair (sskl; spkl) corresponding to Sigprivate key()=V erpublic key(). The full private key of the user is (xl; yl; sskl) and the public key is (ql; dl; spkl). Finally, a certificate signed by the TA using msk is issued for the user.
- OffTagGen: The user Ul chooses a set of random values fwi;l; ri;lgi2f1;:::;Blg 2 Zp, computes fWi;l = wi;lxl;Ri;l = ri;lxlgi2f1;:::;Blg and generates the offline tags fToff i;l gi2f1;:::;Blg in the background, where Bl is the number of the tags that the user wants to generate,

Toff i;l = qWi;l l gRi;l :

- OnTagGen: Given a file Fl with filename namel, similar to [25], we split Fl into nl blocks

{mjgj2f1;:::;nl};

Batch Audit:

The above Audit algorithm only allows the TPA to perform the auditing tasks from different users independently. However, since the TPA will receive huge requests from different users, it is inefficient for the TPA to perform the auditing tasks independently. Here, we propose a batch auditing technique, i.e., the TPA may perform the auditing tasks simultaneously. Assume that there exist K auditing tasks requested by K users fUlgl2f1;:::;Kg

on K data files. For simplicity, we assume that all the K files have the same number of n blocks. Let

the private-public key pair of Ul be
((xl; yl; sskl); (ql; dl; spkl));

where xl; yl 2 Zp, ql = gyl and dl = hxl ,
(sskl; spkl) is a private-public key pair corresponding to Sigprivate key()=V erpublic key(). Assume Ul's file is Fl and the corresponding filename is namel;
the offline tags corresponding to file Fl are
Toff
j;l = qWj;l
l gRj;l ;
where fWj;l = wj;lxl;Rj;l = rj;lxlgj2f1;:::;ng,
fwj;l; rj;lgj2f1;:::;ng 2 Zp; the online tags corresponding
to file Fl are
Ton
j;l = (wj;l □ mj;l)yl + rj;l:
The batch auditing technique works as follows:
1) The TPA chooses the indices of the chosen blocks J = fs1; :::scg and random values V = fvs1 ; :::; vscg, where vsi 2 Zp. Then the TPA sends the challenge message
chal = (fnamelgl2f1;:::;Kg; J;V)
to the CSP for the K users.
2) On receiving chal from the TPA, for l 2 f1; :::;Kg, the CSP calculates
0l =Xj2J
vjmj;l and l =Yj2J
(Toff
j;l)vj :

IV. EVALUATION

A. Security Analysis

We evaluate the security of the proposed scheme by analyzing its fulfilment of the security guarantee described in Section 2.2, namely, the storage correctness and privacy-preserving property. We start from the single user case, where our main result is originated. Then we show the security guarantee of batch auditing for the TPA in multi-user setting.

1) Storage Correctness Guarantee

We need to prove that the cloud server cannot generate valid response for the TPA without faithfully storing the Data .

2) Privacy Preserving Guarantee

The below theorem shows that TPA cannot derive users' data from the information collected during auditing.

3) Security Guarantee for Batch Auditing

Now we show that our way of extending our result to a multi-user setting will not affect the aforementioned security insurance

B. Performance Analysis

Asymptotic efficiency analysis on the batch auditing, by considering only the total number of pairing operations. However, on the practical side, there are additional less expensive operations required for batching, such as modular exponentiations and multiplications. Thus, whether the benefits of removing pairings significantly outweighs these additional operations remains to be verified. To get a complete view of batching efficiency, we conduct a timed batch auditing test, where the number of auditing tasks is increased from 1 to approximately 200 with intervals of 8. Note that we only focus on the choice of s = 1 here, from which similar performance results can be directly obtained for the choice of s = 10. The performance of the corresponding non-batched (individual) auditing is provided as a baseline for the measurement. Following the same settings c = 300 and c = 460, the average per task auditing time, which is computed by dividing total auditing time by the number of tasks, is given in Fig. 2 for both batch and individual auditing. It can be shown that compared to individual auditing, batch auditing indeed helps reducing the TPA's computation cost, as more than 15% of per-task auditing time is saved.

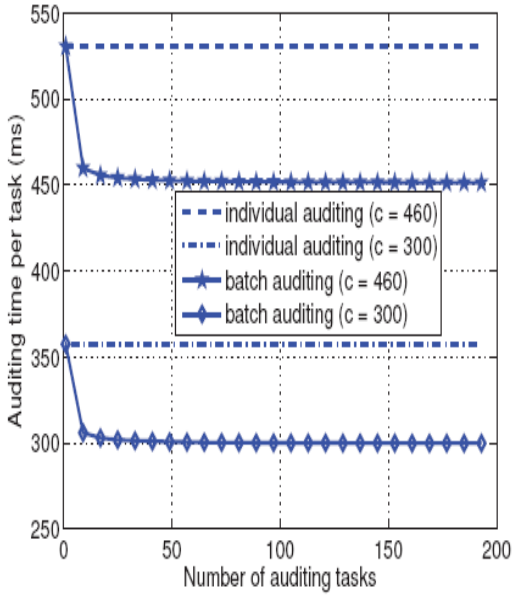


Fig. 2: Comparison on auditing time between batch and individual auditing: Per task auditing time denotes the total auditing time divided by the number of tasks.

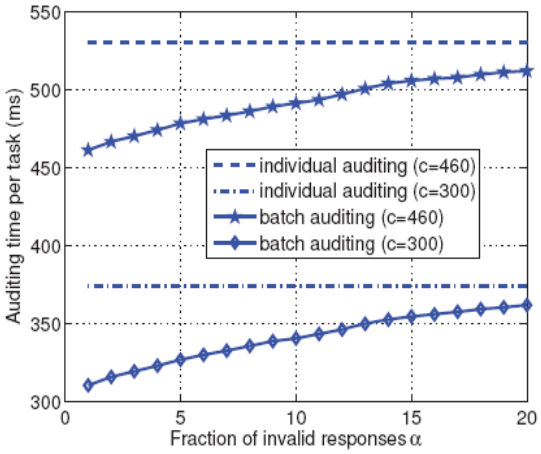


Fig. 3: Comparison on auditing time between batch and individual auditing, when α-fraction of 256 responses are invalid: Per task auditing time denotes the total auditing time divided by the number of tasks

C. Computational Cost on User Side

We compare the computational cost of our protocols on user side with that of the protocol in this section. The theoretical comparison result between our protocols and the protocol. When a file is given, since the most complex part is computed as a background computation, our protocols only need to compute one MulZ and two Add operations to generate the final tag. However, one H, two Exp and one MulG operations are needed in the protocol

D. Efficiency of Audit

The efficiency of the whole protocol is also dominated by the Audit phase. In this section, we evaluate the performance of our Audit phase. As mentioned before, the TPA only needs to select c file blocks to be checked rather than all the file blocks. In order to achieve the high assurance, the value of c is usually selected to be 300 and 460 for the probability of 95% and 99% respectively. we show the time cost for the TPA to verify the response from the CSP. TPA only needs 8.65ms for 300 sampled blocks and 8.86ms for 460 sampled blocks in our protocols.

E. Efficiency of Batch Audit

we show the efficiency of the batch auditing protocol. We compare the efficiency of Batch Audit with Audit of our protocols and the batch auditing protocol.

F. Transmission Overhead

we compare the transmission overheads of our protocols and the protocols. Since the GlobeSetup, UserSetup, OffTagGen and OnTagGen algorithms only needs to be run once, we only compare the Audit, Batch Audit, Modification, Insertion and Deletion algorithms.

V. CONCLUSION

In this paper we depict the cloud architecture and the basic functionality of cloud services related to the data storage security. Additionally we have seen the several related study of the cloud storage security issues. Through this study, we can identify the security issues of cloud users when they working with the cloud data. Mainly security issues can be avoided by using the internal data auditing method without modifying the integrity of the stored data in the cloud.

ACKNOWLEDGEMENT

Corresponding Author: This paper is created by the support of mentors and professors of Bannari Amman Institute of Technology, Sathyamangalam, India.

REFERENCES

[1] Wang, Q. Wang, K. Ren, and W. Lou, "Privacy-preserving public auditing for storage security in cloud computing," in Proc. of IEEE INFOCOM'10, March 2010.
[2] P. Mell and T. Grance, "Draft NIST working definition of cloud computing," Referenced on June.3rd,2009.http:// csrc.nist.gov/groups/ SNS/cloudcomputing/index.html.
[3] 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, "Above the clouds: A Berkeley view of cloud computing," University of California, Berkeley, Tech. Rep. UCBEECS- 2009-28, Feb 2009.
[4] Cloud Security Alliance, "Top threats to cloud computing,"2010, http://www.cloudsecurityalliance.org.
[5] M. Arrington, "Gmail disaster: Reports of mass email-deletions,"2006, http://www.techcrunch.com/2006/12/28/gmail-disasterreports-of-mass-email-deletions/.
[6] J. Kincaid, "MediaMax/TheLinkup closes its doors,"http:// www.techcrunch.com /2008/07/10/mediamaxthelinkup-closes-its-doors/.
[7] Q. Wang, C. Wang, K. Ren, W. Lou, and J. Li, "Enabling public auditability and data dynamics for storage security in cloud computing," IEEE Transactions on Parallel and Distributed Systems, vol. 22, no. 5, pp. 847–859, 2011.
[8] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song, "Provable data possession at untrusted stores," in Proc. of CCS'07, 2007, pp. 598–609.
[9] M. A. Shah, R. Swaminathan, and M. Baker, "Privacy-preserving audit and extraction of digital contents," Cryptology ePrint Archive, Report 2008/186, 2008.
[10] A. Juels and J. Burton S. Kaliski, "PORs: Proofs of retrievability for large files," in Proc. of CCS'07, October 2007, pp. 584–597.
[11] Cloud Security Alliance, "Security guidance for critical areas of focus in cloud computing," 2009, http://www.cloudsecurityalliance.org.
[12] 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, 2010.
[13] 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," Communications of the ACM, vol. 53, no. 4, pp. 50–58, Apirl 2010.