

Secure and Dependable Cloud Storage Services for CRM

U. Rama Govinda Reddy

PG Student, Bonam Venkata Chalamayya Engineering College, Odalarevu, Andhra Pradesh, India

Abstract: *Cloud storage allows users to remotely store their knowledge and luxuriate in the on-demand prime quality cloud applications while not the burden of native hardware and software package management. Although the advantages area units clear, such a service is additionally relinquishing users' physical possession of their outsourced knowledge that inevitably poses new security risks toward the correctness of the info in cloud. So as to handle this new downside and additional come through a secure and dependable cloud storage service, we have a tendency to propose during this paper a versatile distributed storage integrity auditing mechanism, utilizing the homomorphic token and distributed erasure-coded knowledge. The projected style permits users to audit the cloud storage with terribly light-weight communication and computation price. The auditing result not solely ensures robust cloud storage correctness guarantee, however additionally at the same time achieves quick knowledge error localization, i.e., the identification of misbehaving server. Considering the cloud knowledge area unit dynamic in nature, the projected style additional supports secure and economical dynamic operations on outsourced knowledge, as well as block modification, deletion, and append. Analysis shows the projected theme is very economical and resilient against Byzantine failure, malicious knowledge modification attack, and even server colluding attacks.*

Keywords: Cloud storage, tendency, error localization

1. Introduction

In recent years, cloud computing has become a hot topic in the global technology business. The initiatives embrace Google's research for building associate degree infrastructure to support analysis wants of top-tier yank universities. Weiss noted that cloud computing services embrace many existing computing technologies, like service-oriented utility computing, grid computing with great amount of computing resources, which mistreatment knowledge centers for knowledge storage services. Prior to the event of the conception of cloud computing, vital industrial knowledge was hold on internally on storage media, protected by security measures as well as firewalls to forestall external access to the information and as well as organizational rules to ban unauthorized internal access. Within the cloud computing atmosphere, storage service providers should have in situ knowledge security practices to confirm that their clients' knowledge is safe from unauthorized access and disclosure.

Additional significantly, the rules and measures for preventing privileged users like system directors from unauthorized access should be strictly established and implemented. Service suppliers follow specific policies and practices to protect their users' knowledge, and these policies are sometimes declared in the contract. Most current network application services have constant follow. As an example a Yahoo! webmail user must scan the contract on-line and show his consent to the contract before he will use the webmail service. The content of the contract covers definitions of service items, service scope, service modification notification, scope of privacy protection, rules on user knowledge assortment, use, sharing and statements relating to user responsibilities. Showing the consent to the contract is an essential step of the service application. In a cloud computing surroundings, the service content offered by service suppliers may be adjusted in step with the needs of the user. as an example, the person will request different

amounts of storage, transmission speeds, levels of data encryption and different services. Additionally to process the service things, the agreement unremarkably additionally notes the time, quality and performance needs given the service. Generally, this service agreements area unit remarked as Service Level Agreements (SLA). By sign language AN SLA, the user shows that he has understood and in agreement to the contents of the application service, and agrees with the provider's information privacy and protection policies.

A common approach to shield user information is that user information is encrypted before it's hold on. during a cloud computing environment, a user's information can even be hold on following additional cryptography, however if the storage and cryptography of a given user's information is performed by an equivalent service supplier, the service provider's internal employees (e.g., system directors and approved staff) will use their coding keys and internal access privileges to access user information. From the user's perspective, this might place his hold on information in danger of unauthorized revealing. Creating user trust through the protection of user's information content is that the key to the widespread acceptance of the cloud computing. This study proposes a business model for cloud computing supported the conception of employing a separate cryptography and coding service. Within the model, information storage and decryption of user information square measure provided singly by 2 distinct providers. Additionally, those operating with the info storage system can don't have any access to decrypted user information, and those working with user encryption and coding can delete all encrypted and decrypted user information once transferring the encrypted information to the system of the info storage service provider. Under the business model planned during this study, the data storage cloud system supplier is allowed to store the user's encrypted information, however doesn't have access to the decipherment Key. Thus, the storage system will solely retrieve encrypted user data, however is unable to decipher it. The cloud system responsible for

encrypting user information has authority over all encryption keys needed for cryptography or encoding or encryption however, only if the encryption supplier doesn't store the user's information, internal mismanagement of the decipherment keys still poses no risk of unauthorized revelation of the user's information. Given that encoding is AN freelance cloud computing service, a novel feature of the business model is that completely different services square measure provided by multiple operators. As an example, the Encryption as a Service supplier and therefore the "Storage as a Service" provider collaborate to supply a Cloud Storage System with effective information protection. This study provides a draft SLA for this type of business model of mixing multiple suppliers in a single service, which may establish the cooperation model between operators and therefore the division of responsibility for the services they together offer to the user.

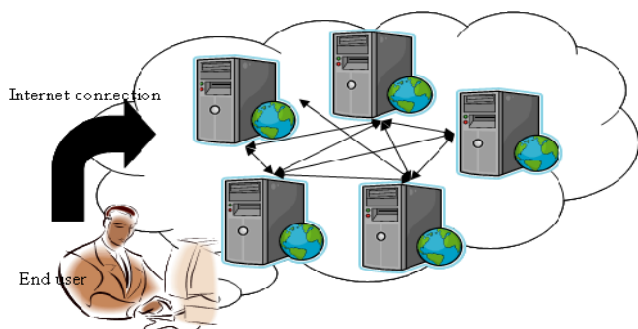


Figure 1: Cloud Computing Concept Map

2. Ensuring Cloud Data Storage

In cloud data storage system, users store their data in the cloud and no longer possess the data locally. Thus, the correctness and availability of the data files being stored on the distributed cloud servers must be guaranteed. One of the key issues is to effectively detect any unauthorized data modification and corruption, possibly due to server compromise and/or random Byzantine failures. Besides, in the distributed case when such inconsistencies are successfully detected, to find which server the data error lies in is also of great significance, since it can always be the first step to fast recover the storage errors and/or identifying potential threats of external attacks.

To address these problems, our main scheme for ensuring cloud data storage is presented in this section. The first part of the section is devoted to a review of basic tools from coding theory that is needed in our scheme for file distribution across cloud servers. Then, the homomorphic token is introduced. The token computation function we are considering belongs to a family of universal hash function [5], chosen to preserve the homomorphic properties, which can be perfectly integrated with the verification of erasure coded data [4]. Subsequently, it is shown how to derive a challenge-response protocol for verifying the storage correctness as well as identifying misbehaving servers. The procedure for file retrieval and error recovery based on erasure-correcting code is also outlined. Finally, we describe how to extend our scheme to third party auditing with only slight modification of the main design.

3. Providing Dynamic Data Operation Support

So far, we assumed that F represents static or archived data. This model may fit some application scenarios, such as libraries and scientific data sets. However, in cloud data storage, there are many potential scenarios where data stored in the cloud is dynamic, like electronic documents, photos, or log files, etc. Therefore, it is crucial to consider the dynamic case, where a user may wish to perform various block-level operations of update, delete, and append to modify the data file while maintaining the storage correctness assurance. Since data do not reside at users' local site but at cloud service provider's address domain, supporting dynamic data operation can be quite challenging. On the one hand, CSP needs to process the data dynamics request without knowing the secret keying material. On the other hand, users need to ensure that the entire dynamic data operation request has been faithfully processed by CSP. To address this problem, we briefly explain our approach methodology here and provide the details later. For any data dynamic operation, the user must first generate the corresponding resulted file blocks and parities. This part of operation has to be carried out by the user, since only he knows the secret matrix P . Besides, to ensure the changes of data blocks correctly reflected in the cloud address domain, the user also needs to modify the corresponding storage verification tokens to accommodate the changes on data blocks. Only with the accordingly changed storage verification tokens, the previously discussed challenge-response protocol can be carried on successfully even after data dynamics. In other words, these verification tokens help to ensure that CSP would correctly execute the processing of any dynamic data operation request. Otherwise, CSP would be caught cheating with high probability in the protocol execution later on. Given this design methodology, the straightforward and trivial way to support these operations is for user to download all the data from the cloud servers and recompute the whole parity blocks as well as verification tokens. This would clearly be highly inefficient. In this section, we will show how our scheme can explicitly and efficiently handle dynamic data operations for cloud data storage, by utilizing the linear property of Reed-Solomon code and verification token construction.

4. Methods

4.1 Challenge Token Pre-computation

In order to achieve assurance of data storage correctness and data error localization simultaneously, our scheme entirely relies on the pre-computed verification tokens. The main idea is as follows: before file distribution the user pre-computes a certain number of short verification tokens on individual vector $G(j)$ ($j \in \{1, \dots, n\}$), each token covering a random subset of data blocks. Later, when the user wants to make sure the storage correctness for the data in the cloud, he challenges the cloud servers with set randomly generated block indices. Upon receiving challenge, each cloud server computes a short "signature" over the specified blocks and returns them to the user. The values of these signatures should match the corresponding tokens pre-computed by the user. Meanwhile, as all servers operate over the same subset of the indices, the requested response values for integrity

check must also be a valid codeword determined by secret matrix P .

Algorithm 1 Token Pre-computation

```
1: procedure
2: Choose parameters  $l, n$  and function  $f, \emptyset$ ;
3: Choose the number  $t$  of tokens;
4: Choose the number  $r$  of indices per verification;
5: Generate master key  $KPRP$  and challenge key  $kchal$ ;
6: for vector  $G(j), j \leftarrow 1, n$  do
7: for round  $i \leftarrow 1, t$  do
8: Derive  $\alpha_i = fkchal(i)$  and  $k(i)$  prp from  $KPRP$ .
9: Compute  $v(j) = \sum_{q=1}^r \alpha_q i * G(j)[\emptyset k(i) \text{ prp } (q)]$ 
10: end for
11: end for
12: Store all the  $v_i$ 's locally.
13: end procedure
```

Suppose the user wants to challenge the cloud servers t times to ensure the correctness of data storage. Then, he must pre-compute t verification tokens for each $G(j)$ ($j \in \{1, \dots, n\}$), using a PRF $f(\cdot)$, a PRP $_()$, a challenge key $kchal$ and a master permutation key $KPRP$. Specifically, to generate the i th token for server j , the user acts as follows:

- 1) Derive a random challenge value α_i of $GF(2^p)$ by $\alpha_i = fkchal(i)$ and a permutation key $k(i)$ prp based on $KPRP$.
- 2) Compute the set of r randomly-chosen indices: $\{I_q \in [1, \dots, l] \mid 1 \leq q \leq r\}$, where $I_q = k(i)$ prp (q) .
- 3) Calculate the token as: $v(j)_i = \sum_{q=1}^r \alpha_q i * G(j)[I_q]$, where $G(j)[I_q] = g(j)I_q$.

Note that $v(j)_i$, which is an element of $GF(2^p)$ with small size, is the response the user expects to receive from server j when he challenges it on the specified data blocks.

4.2 File Retrieval and Error Recovery

Since our layout of file matrix is systematic, the user can reconstruct the original file by downloading the data vectors from the first m servers, assuming that they return the correct response values. Notice that our verification scheme is based on random spot-checking, so the storage correctness assurance is a probabilistic one. However, by choosing system parameters (e.g., r, l, t) appropriately and conducting enough times of verification, we can guarantee the successful file retrieval with high probability.

Algorithm 2 Error Recovery

```
1: procedure
% Assume the block corruptions have been detected among
the specified  $r$  rows;
% Assume  $s \leq k$  servers have been identified misbehaving
2: Download  $r$  rows of blocks from servers;
3: Treat  $s$  servers as erasures and recover the blocks.
4: Resend the recovered blocks to corresponding servers.
5: end procedure
```

On the other hand, whenever the data corruption is detected, the comparison of pre-computed tokens and received response values can guarantee the identification of misbehaving server(s) (again with high probability). Therefore, the user can always ask servers to send back blocks of the r rows specified in the challenge and in

Algorithm 2, as long as the number of identified misbehaving servers is less than k . (otherwise, there is no way to recover the corrupted blocks due to lack of redundancy, even if we know the position of misbehaving servers.) The newly recovered blocks can then be redistributed to the misbehaving servers to maintain the correctness of storage.

4.3 Proposed System

In this paper, we propose an effective and flexible distributed scheme with explicit dynamic data support to ensure the correctness of users' data in the cloud. We rely on erasure correcting code in the file distribution preparation to provide redundancies and guarantee the data dependability. This construction drastically reduces the communication and storage overhead as compared to the traditional replication-based file distribution techniques. By utilizing the homomorphic token with distributed verification of erasure-coded data, our scheme achieves the storage correctness insurance as well as data error localization: whenever data corruption has been detected during the storage correctness verification, our scheme can almost guarantee the simultaneous localization of data errors, i.e., the identification of the misbehaving server(s).

Advantages

1. Compared to many of its predecessors, which only provide binary results about the storage state across the distributed servers, the challenge-response protocol in our work further provides the localization of data error.
2. Unlike most prior works for ensuring remote data integrity, the new scheme supports secure and efficient dynamic operations on data blocks, including: update, delete and append.
3. Extensive security and performance analysis shows that the proposed scheme is highly efficient and resilient against Byzantine failure, malicious data modification attack, and even server colluding attacks.

4.4 User Registration and Control

This module can be also used to register users for custom modules that support personalization and user specific handling. If the users wish to create their own user accounts, i.e. register, then registration checks for the username availability and assign unique ID. User Control means controlling the login with referring the username and password which are given during the registration process. After login, the user can encrypts the original data and stored it in database, and the user can retrieve the original data which gets decrypted after checking the unique ID and searched data. Based on their logins, they have rights to view, or edit or update or delete the contents of resources. Part of the stored data are confidential, but when these institutions store the data to equipment afforded by cloud computing service provider, priority accessing to the data is not the owner, but cloud computing service provider. Therefore, there is a possibility that stored confidential data cannot rule out being leaked. Also there is no possibility to track the original data for the hackers.

4.5 CRM Service

This module is customer relationship management, where the user can interact with the application. CRM is concerned with the creation, development and enhancement of individualised customer relationships with carefully targeted customers and customer groups resulting in maximizing their total customer life-time value. CRM is a business strategy that aims to understand, anticipate and manage the needs of an organisation's current and potential customers. It is a comprehensive approach which provides seamless integration of every area of business that touches the customer- namely marketing, sales, customer services and field support through the integration of people, process and technology. CRM is a shift from traditional marketing as it focuses on the retention of customers in addition to the acquisition of new customers. The expression Customer Relationship Management (CRM) is becoming standard terminology, replacing what is widely perceived to be a misleadingly narrow term, relationship marketing (RM). The main purpose of CRM is:

- The focus [of CRM] is on creating value for the customer and the company over the longer term.
- When customers value the customer service that they receive from suppliers, they are less likely to look to alternative suppliers for their needs.
- CRM enables organisations to gain 'competitive advantage' over competitors that supply similar products or services.

CRM consists of index page, registration page, login page, etc. Through this, the user can register with the user details, after registration the user can send the original data, which gets encrypted and stored in database; also the user can retrieve the original data which they stored only after decrypting the encrypted data by giving the decryption key.

5. Encryption / Decryption Service

This module describes about the encryption and decryption process for the original data. The encryption process is needed while storing the data, and the data decryption is needed while retrieving the data. After the user's login has been successfully verified, if the CRM Service System requires client information from the user, it sends a request the information (for encryption and decryption) to the Storage Service System.

Encryption: In this (data storage service), the CRM Service System transmits the user ID to the Storage Service System where it searches for the user's data. This original data, once found, a request must be sent to the Encryption/Decryption Service System along with the user ID. It shows the Storage Service System executing the transmission of client data and the user ID to the Encryption/Decryption Service System. Here, the users send original data gets encrypted and stored in storage service as per the user request. That data cannot be hacked by unauthorized one that is more confidential and encrypted.

Decryption: In this (data retrieval service), if the user request the CRM service to retrieve the data which are stored in Storage service, the CRM sends the user ID and the

search data to the Encryption/Decryption Service System. It authenticates whether the user ID and search data are owned by the same user. If authenticated, the encrypted data from the storage service system is send to the Encryption/Decryption Service System for the decryption process. In that process, it checks for decryption key, if it OK, then decrypts the encrypted data and the original data retrieved, and send to the user.

5.1 Accessing Storage Service

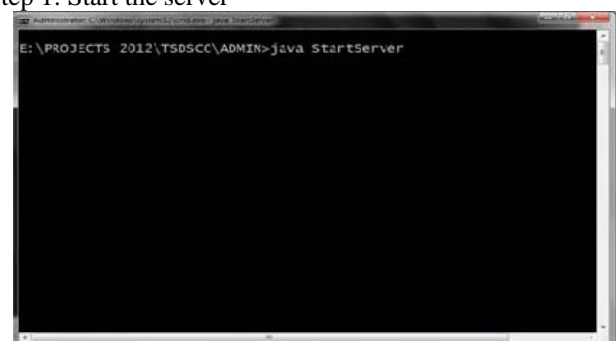
This module describes about how the data gets stored and retrieved from the database. The original data which given by the user gets encrypted and request for the storage, the storage service system store the encrypted data with the user ID for avoiding the misuse of data. Also during retrieval, the user request for retrieving the data by giving the search data, the storage service system checks for user ID and search data are identical, if so it sends the encrypted data to the Encryption/Decryption Service System for the decryption process, it decrypts the data and sends to the user. The user interacts with the database every time through the CRM service only.

The user's goal in logging into the CRM Service System is possibly to maintain part of the client data, thus the system design must take data maintenance into consideration. Feasible design methods include matching the encrypted client data with the corresponding user ID and client ID, thus allowing for the indexing of the user ID to obtain the corresponding client data. Then the client ID can be used to index the client data the user wishes to maintain. Considering the massive amount of client data, search efficiency could be improved by combining the user ID and client ID to form a combined ID used for searching for a specific client's data.

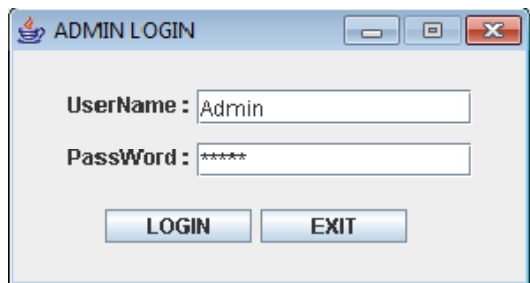
In the new business model, multiple cloud service operators jointly serve their clients through existing information technologies including various application systems such as ERP, accounting software, portfolio selection and financial operations which may require the user ID to be combined with other IDs for indexing stored or retrieved data. In addition, the foregoing description of the two systems can use Web Service related technology to achieve operational synergies and data exchange goals.

6. Implementation and Results

Step 1: Start the server



Step 2: Now go to admin page. Enter user name and password and click login button



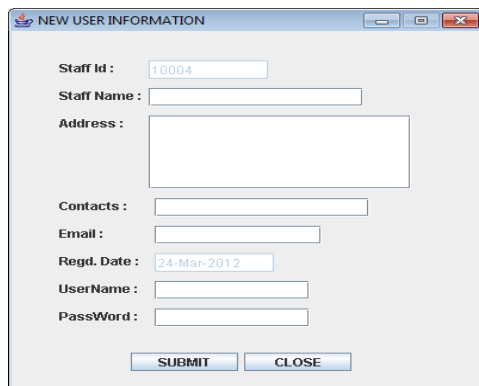
ADMIN LOGIN

UserName : Admin

PassWord : *****

LOGIN EXIT

Step 3: Main form will be displayed where the users are created. Create a user by filling the details in the below page.



NEW USER INFORMATION

Staff Id : 10004

Staff Name :

Address :

Contacts :

Email :

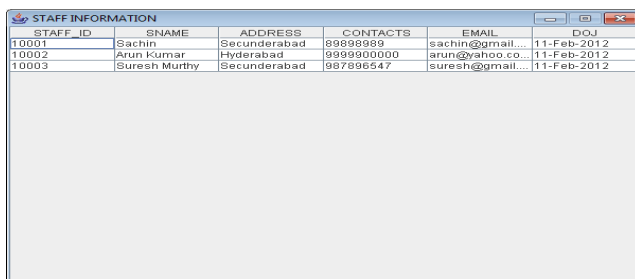
Regd. Date : 24-Mar-2012

UserName :

PassWord :


SUBMIT CLOSE

Step 4: After filling the details click submit button. In order to view the list of users go to view users option and list of users will be displayed as shown below:



STAFF_ID	SNAME	ADDRESS	CONTACTS	EMAIL	DOJ
10001	Sachin	Secunderabad	99999999	sachin@gmail...	11-Feb-2012
10002	Arun Kumar	Hyderabad	9999900000	arun@yahoo.co...	11-Feb-2012
10003	Suresh Murthy	Secunderabad	987896547	suresh@gmail...	11-Feb-2012

Step 5: Now Start user to perform the transactions. First enter the user id and password and click connect button.



USER LOGIN

UserName : sachin

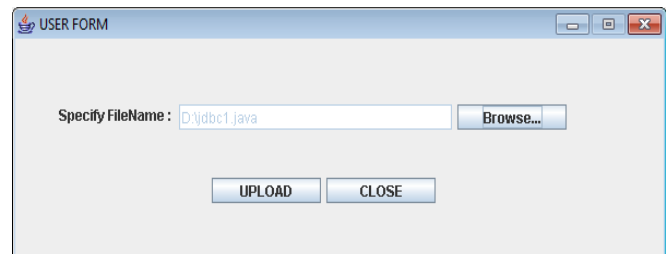
PassWord : *****

Server Name : honey

CONNECT EXIT

A user form will be displayed. Go to upload option and upload the file

Step 6: Now the user uploads the file by browsing the file from the drive



USER FORM

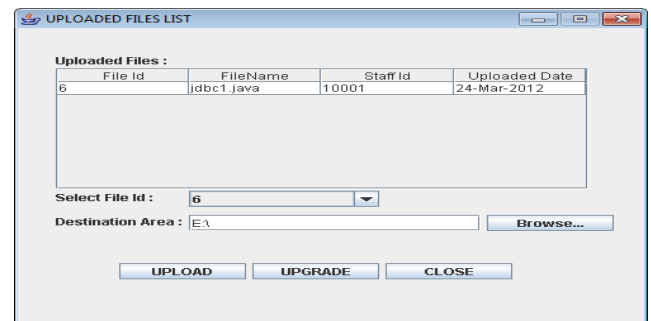
Specify FileName : D:\jdbc1.java

Browse...

UPLOAD CLOSE

Click on upload button. Now the file will be stored in temp folder

Step 7: Now go to admin page and view the uploaded files and specify the destination path where the files must be saved press



UPLOADED FILES LIST

Uploaded Files :

File Id	FileName	Staff Id	Uploaded Date
6	jdbc1.java	10001	24-Mar-2012

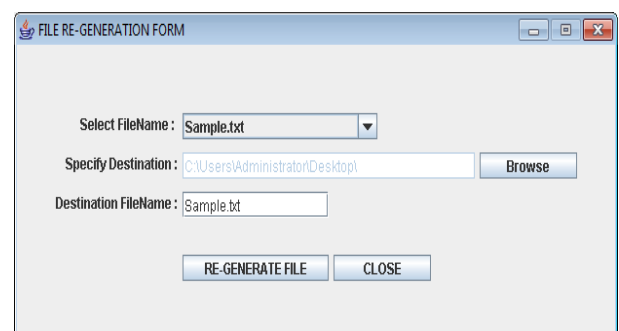
Select File Id : 6

Destination Area : E:\

Browse...

UPLOAD UPGRADE CLOSE

Step 8: If the file has been deleted or modified, there is a chance to regenerate the original file using regenerate option. Enter the filename in regenerate form and specify the destination in order to save the file at that specified location.



FILE RE-GENERATION FORM

Select FileName : Sample.txt

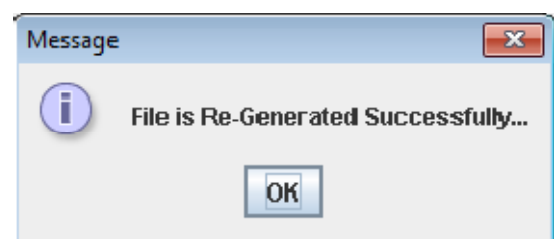
Specify Destination : C:\Users\Administrator\Desktop\

Browse

Destination FileName : Sample.txt

RE-GENERATE FILE CLOSE

Step 9: If the file has been regenerated successfully, a successful message will be displayed as shown below



Message

File is Re-Generated Successfully...

OK

7. Conclusion and Future Scope

By using this project we are performing encryption and decryption. We will hide the details of users. In this we are using AES algorithm and two databases. In future we will apply this project for more number of databases. To achieve the assurances of cloud data integrity and availability and enforce the quality of dependable cloud storage service for users, we propose an effective and flexible distributed

scheme with explicit dynamic data support, including block update, delete, and append. By utilizing the homomorphic token with distributed verification of erasure-coded data, our scheme achieves the integration of storage correctness insurance and data error localization, i.e., whenever data corruption has been detected during the storage correctness verification across the distributed servers, we can almost guarantee the simultaneous identification of the misbehaving server(s).

Considering the time, computation resources, and even the related online burden of users, we also provide the extension of the proposed main scheme to support third-party auditing, where users can safely delegate the integrity checking tasks to third-party auditors and be worry-free to use the cloud storage services. Through detailed security and extensive experiment results, we show that our scheme is highly efficient and resilient to Byzantine failure, malicious data modification attack, and even server colluding attacks.

References

- [1] 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, July 2009.
- [2] Amazon.com, "Amazon Web Services (AWS)," <http://aws.amazon.com>, 2009.
- [3] Sun Microsystems, Inc., "Building Customer Trust in Cloud Computing with Transparent Security," https://www.sun.com/offers/details/sun_transparency.xml, Nov. 2009.
- [4] K. Ren, C. Wang, and Q. Wang, "Security Challenges for the Public Cloud," IEEE Internet Computing, vol. 16, no. 1, pp. 69-73, 2012.
- [5] M. Arrington, "Gmail Disaster: Reports of Mass Email Deletions," <http://www.techcrunch.com/2006/12/28/gmail-disasterreportsof-mass-email-deletions>, Dec. 2006.