ISSN:0975-9646

Ensuring Correctness and Error Localisation In Cloud

Deepitha K R, Animesh Giri

PES Institute of Technology - South Campus,
Bangalore- 560100

Abstract: Cloud Computing has been envisioned as the next generation architecture of IT Enterprise. In contrast to traditional solutions, where the IT services are under proper physical, logical and personnel controls, Cloud Computing moves the application software and databases to the large data centres, where the management of the data and services may not be fully trustworthy. This unique attribute, however, poses many new security challenges which have not been well understood. In this paper, we are implementing the cloud data storage security, which has always been an important aspect of quality of service. To ensure the correctness of users' data in the cloud, we propose an effective and flexible distributed scheme with two salient features, opposing to its predecessors. By utilizing the homomorphism token with distributed verification of erasure-coded data and low density parity check, our scheme achieves the integration of storage correctness insurance and data error localization, i.e., the identification of misbehaving server(s). Unlike most prior works, the new scheme further supports secure and efficient dynamic operations on data blocks, including: data update, delete and append. 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.

Key Words*: Cloud, Security, TPA, Erasure Encoding, Low Density Parity Check*

1. INTRODUCTION:

Several trends are opening up the era of Cloud Computing, which is an Internet-based development and use of computer technology. The ever cheaper and more powerful processors, together with the software as a service (SaaS) computing architecture, are transforming data centers into pools of computing service on a huge scale. The increasing network bandwidth and reliable yet flexible network connections make it even possible that users can now subscribe high quality services from data and software that reside solely on remote data centers.

Moving data into the cloud offers great convenience to users since they don't have to care about the complexities of direct hardware management. The pioneer of Cloud Computing vendors, Amazon Simple Storage Service (S3) and Amazon Elastic Compute Cloud (EC2) [1] are both well known examples. While these internet-based online services do provide huge amounts of storage space and customizable computing resources, this computing platform shift, however, is eliminating the responsibility of local machines for data maintenance at the same time. As a result, users are at the mercy of their cloud service providers for the availability and integrity of their data. Recent downtime of Amazon's S3 is such an example [2].

From the perspective of data security, which has always been an important aspect of quality of service, Cloud Computing inevitably poses new challenging security threats for number of reasons. Firstly, traditional cryptographic primitives for the purpose of data security protection cannot be directly adopted due to the users' loss control of data under Cloud Computing. Therefore, verification of correct data storage in the cloud must be conducted without explicit knowledge of the whole data. Considering various kinds of data for each user stored in the cloud and the demand of long term continuous assurance of their data safety, the problem of verifying correctness [12] of data storage in the cloud becomes even more challenging. Secondly, Cloud Computing is not just a third party data warehouse. The data stored in the cloud may be frequently updated by the users, including insertion, deletion, modification, appending, reordering, etc. To ensure storage correctness under dynamic data update is hence of paramount importance. However, this dynamic feature also makes traditional integrity insurance techniques futile and entails new solutions. Last but not the least, the deployment of Cloud Computing is powered by data centers running in a simultaneous, cooperated and distributed manner. Individual user's data is redundantly stored in multiple physical locations to further reduce the data integrity threats. Therefore, distributed protocols for storage correctness assurance will be of most importance in achieving a robust and secure cloud data storage system in the real world. However, such important area remains to be fully explored in the literature.

A. Problem Statement: In cloud data storage, a user stores his data through a CSP into a set of cloud servers, which are running in a simultaneous, cooperated and distributed manner. Data redundancy can be employed with technique of erasure-correcting code to further tolerate faults or server crash as user's data grows in size and importance. Thereafter, for application purposes, the user interacts with the cloud servers via CSP to access or retrieve his data. In some cases, the user may need to perform block level operations we are considering are block update, delete, insert and append.

As users no longer possess their data locally, it is of critical importance to assure users that their data are being correctly stored and maintained. That is, users should be equipped with security means so that they can make continuous correctness assurance of their stored data even without the existence of local copies. In case those users do not necessarily have the time, feasibility or resources to monitor their data, they can delegate the tasks to an optional trusted TPA of their respective choices. In our model, we assume that the point-to-point communication

channels between each cloud server and the user is authenticated and reliable, which can be achieved in practice with little overhead, we propose Low Density Parity Check – LDPC Encoding technique for ensuring data correctness stored in cloud scenario.

.

2. LITERATURE SURVEY

A. S***ecure cloud:*** 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. The key issues in data storage are:

There are many constraints, discussed as C1, C2 & C3.

C1. How do I ensure that there is no un-authorized access to my cloud by a disgruntled employee, who has left the organization or by an identity thief?

C2. How to ensure proper levels of authentication to cloud services? How do I manage multi-device access?

C3. In multi-cloud scenario, how do I ensure that I provide / delegate access to users to different security domains so that the end-to-end workflow is seamless? Similarly, in hybrid cloud, how do I create a minimum common access control and identity structure?

Implication: Ensure proper access control and identity management.

Synchronizing enterprise and external cloud services access control lists in the context of C1 to ensure right access roles is a very important challenging issue as PaaS and SaaS platforms have complex hierarchies and many fine-grained access capabilities (tenant org level, sub-tenant, and individual user levels). This assumes importance as users, who are no longer part of an enterprise, may still potentially exploit access provided in cloud; unless those credentials are revoked quickly. However, we recognize this as more of a process issue than a technology one. Use of standard languages like Service Provisioning Markup Language] promoted by OASIS, can enable faster user account provisioning and de-provisioning. Cloud service authentication (C2) presents some interesting problems. Cloud services are increasingly getting accessed through browsers and thin mobile devices running new set of applications like HTML-5. Browsers do not have direct means of handling XML signatures and XML encryption, and rely on the underlying SSL layer for handshake. Hence this channel may become a potential threat if not secured properly. This may push enterprises to use VPNs while communicating to cloud. The Cloud Security Alliance recommends cloud provider to provide stronger authentication mechanism and also (optionally) allow users to use third party identity management and single sign- on platforms like Microsoft Passport. This may lead to an added set of authentication complexity. Online open identity management communities like OpenID , OAuth etc. are proliferating and each brings its own set of integration challenges for cloud providers.

There is a growing chorus on 'inter cloud' hand-offs and federated identity management (C3), possibly through assertion tokens like Security Assertion Markup Language (SAML) or privilege management infrastructure based on x.509 certificates. The ongoing standardization work WS-federation may provide some help in this aspect. Cloud federations need to establish a set of common security token services and identity providers. But in dynamic cloud scenario these trust relations may not work. We need to develop more flexible cases of identity federation.

B. Security threats:

The table 1, extracted from a study done in, shows the main cloud computing security issues. As shown in the column Total**,** more than 70 % of the SME (small and medium enterprises) interviewed for this study are concerned by the first six criteria and more specifically by confidentiality of data.

The European Network and Information Security Agency (ENISA) identified thirty-five risks divided in four types: policy and organizational risks, technical risks, legal risks and risks that are not specific to the cloud [6]. From these risks, the ENISA extracted the eight most important risks. Five of these eight risks concerns directly or indirectly the data confidentiality: isolation failure, management interface compromise, data protection, insecure or incomplete data deletion, and malicious insider. Similarly to ENISA, the CSA identified thirteen kinds of risks related to cloud computing [3] and selected the seven most important risks [7]. Five of these risks concerns directly or indirectly data confidentiality: insecure application programming interfaces, malicious insiders, shared technology vulnerabilities data loss/leakage, account service and traffic hijacking. However, some of these security risks are not specific to the cloud computing [8].

Criteria	Very Important	Showstopper	Total
Confidentiality of corporate data	30,9%	63,6%	94,5%
Privacy	43,9%	43,9%	87,8%
Availability of services and/or data	47,3%	40,0%	87,3%
Integrity of services and/or data	42,6%	44,4%	87,0%
Loss control of services and/or data	47,2%	28,3%	75,5%
Lack of liability of providers in case of security incidents	43,1%	29,4%	72,5%
Repudiation	47,9%	8,3%	56,2%

After having seen the main cloud computing security issues, we now focus on the security properties provided by the existing databases and we briefly present the results of the study we have done on these databases taking into account the feedback of final users, personal tests and databases documentation. For this study, we have chosen the most popular freeware and proprietary traditional databases (such as SQL Microsoft, Oracle, DB2, MySQL, PostgreSQL) as well as clouds databases (such as Amazon SimpleDB, Google DataStore and Azure SQL). To compare the existing databases, from a security point of view, we have defined ten security criteria

- **Users' identification and authentication:** identify the database users in a safe and not ambiguous way in order to apply the access controls according to the rights of each user.

- **Identification robustness and authentication:** guarantee that it is difficult to usurp an identity (e.g. password robustness).
- **Rights separation:** distinguish various types of users with predefined actions to separate for example the data exploitation tasks and the database maintenance tasks.
- **Data Access control:** allow access to the stored data only to authorized persons. This access control must allow various access modes (reading and/or writing) and a variable granularity (all databases or one or several database tables).
- **Integrity and confidentiality of the stored data:** ensure that only authorized users can modify the stored data in the server hosting the DBMS (Database Management System) and read critical data.
- **Communications ciphering:** ensure the integrity and if needed the confidentiality of the requests and data exchanged between the various equipments implementing or using the database service i.e. exchanges between clients hosts (end-users or administrators) and the server hosting the DBMS or between servers in case of a distributed DBMS (data replication).
- **Data concealment:** conceal real data in artificial one to falsify the volume of real data, specifically when data are in production, while allowing finding easily the real data.
- **Data masking:** use an irreversible process to replace sensitive data and ensure that original data cannot be found or restored. This property is very important particularly when data are used in the context of application developments and tests.
- **Audit services:** log the events concerning the accesses to the DBMS and ensure the integrity of the logs. This type of service is necessary to control a posteriori the accesses and detects the users who exceed their rights.
- **Certification:** This last criterion concerns the EAL (Evaluation Assurance Level) certification defined in seven levels that allows evaluation of IT applications. For civil applications, the EAL is generally between 1 and 4+ and for military applications between 5 and 7.

Using these ten criteria, we can see that traditional databases are highly secured. This is particularly true for proprietary databases (i.e. Microsoft SQL and Oracle) that are more mature than those used in clouds that are relatively new. We can also see that cloud databases do not ensure data confidentiality as generally cloud providers let the users/clients manage the confidentiality of their data. Moreover, usually in cloud databases such as Amazon SimpleDB, data access control is provided but not at a fine granularity level. Actually, the access to a database is associated to a user account or user role that has all the rights on it and it is not possible to restrict the access rights. Finally, we can see that there is no data concealment mechanism, i.e. allowing protection against the problem of statistics explained in the previous section. Oracle is the only database proposing a similar service but for producing artificial data in the context of data not in production. Therefore, in the following section, we present our solution of data concealment allowing falsification of any kind of statistics done on data in production.

C. Adversary Model:

Security threats faced by cloud data storage can come from two different sources. On the one hand, a CSP can be self-interested, un-trusted and possibly malicious. Not only does it desire to move data that has not been or is rarely accessed to a lower tier of storage than agreed for monetary reasons, but it may also attempt to hide a data loss incident due to management errors, Byzantine failures and so on. On the other hand, there may also exist an economically motivated adversary, who has the capability to compromise a number of cloud data storage servers in different time intervals and subsequently is able to modify or delete users' data while remaining undetected by CSPs for a certain period.

There are 2 types of adversary model

Weak Adversary: The adversary is interested in corrupting the user's data files stored on individual servers. Once a server is comprised, an adversary can pollute the original data files by modifying or introducing its own fraudulent data to prevent the original data from being retrieved by the user.

Strong Adversary: This is the worst case scenario, in which we assume that the adversary can compromise all the storage servers so that he can intentionally modify the data files as long as they are internally consistent. In fact, this is equivalent to the case where all servers are colluding together to hide a data loss or corruption incident.

3. EXISTING MECHANISMS: PERFORMANCE COMPARISON

Reed- Solomon technique ensures the correctness and error localization on the stored data; LDPC codes have a great advantage in terms of encoding and decoding time over Reed-Solomon codes

Table 2: Comparison of Reed-Solomon and LDPC

	Reed Solomon code	LDPC
Primary operation	Galois field dot product	XOR
Quality of encoding	Optimal	Sub-optimal
Decoding complexity	O(n^3)	O(n ln(1=)), where 2 R [11]
Overhead factor	1	Roughly 1.15

Table 2 summarizes some of the key differences between Reed-Solomon and LDPC coding. When comparing the two types of coding, the properties of key importance are the encoding and decoding times, and the average number of blocks that are necessary to reconstruct a set. LDPC codes have a great advantage in terms of encoding and decoding time over Reed-Solomon codes; in addition, Reed-Solomon decoding requires n blocks from a set before decoding can begin, while LDPC decoding can take place on-the-fly. However, for small n, the extra blocks that LDPC codes can require for decoding can cause substantial performance degradation. Moreover, for systems where the network connection is slow, Reed-Solomon Codes can sometimes outperform LDPC codes despite the increased decoding penalty

4. PROPOSED MECHANISM FOR ENSURING DATA SECURITY:

Here 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 homomorphism token with distributed verification of erasure-coded data [12], our scheme achieves the storage correctness insurance as well as data error localization, i.e., 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).

- Compared too 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.
- 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.
- 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.

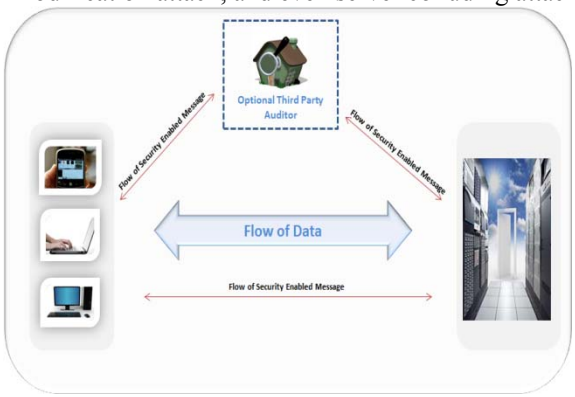


Fig 4.1: Conceptual Architecture

A. **Conceptual network architecture** for cloud data storage is illustrated in Figure 4.1, three different network entities can be identified as follows:

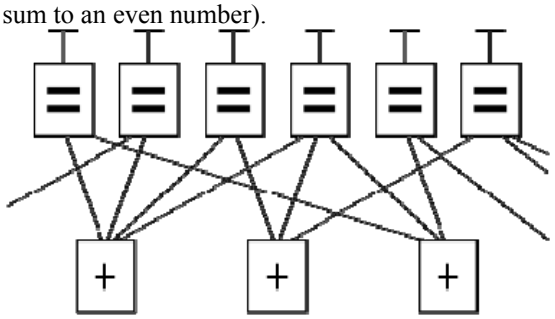
User: Users, who have data to be stored in the cloud and rely on the cloud for data computation, consist of both individual consumers and organizations.

Cloud Service Provider (CSP): A CSP, who has significant resources and expertise in building and managing distributed cloud storage servers, owns and operates live Cloud Computing systems.

Third Party Auditor (TPA): An optional TPA, who has expertise and capabilities that users may not have, is trusted to assess and expose risk of cloud storage services on behalf of the users upon request.

B. Function of low density parity check codes:

LDPC codes are defined by a sparse parity-check matrix. This sparse matrix is often randomly generated, subject to the sparsity constraints. These codes were first designed by Gallager in 1962. Below is a graph fragment of an example LDPC code using Forney's factor graph notation. In this graph, n variable nodes in the top of the graph are connected to (n-k) constraint nodes in the bottom of the graph. This is a popular way of graphically representing an (n, k) LDPC code. The bits of a valid message, when placed on the **T's** at the top of the graph, satisfy the graphical constraints. Specifically, all lines connecting to a variable node (box with an '=' sign) have the same value, and all values connecting to a factor node (box with a '+' sign) must sum, modulo two, to zero (in other words, they must sum to an even number).



$$\mathbf{H} = \begin{pmatrix} 1&1&1&1&0&0 \\ 0&0&1&1&0&1 \\ 1&0&0&1&1&0 \end{pmatrix} \sim \begin{pmatrix} 1&1&1&1&0&0 \\ 0&0&1&1&0&1 \\ 0&1&1&0&1&0 \end{pmatrix} \sim \begin{pmatrix} 1&1&1&1&0&0 \\ 0&1&1&0&1&0 \\ 0&0&1&1&0&1 \end{pmatrix} \sim \begin{pmatrix} 1&1&1&1&0&0 \\ 0&1&1&0&1&0 \\ 1&1&0&0&0&1 \end{pmatrix}.$$

Ignoring any lines going out of the picture, there are 8 possible 6-bit strings corresponding to valid code words: (i.e., 000000, 011001, 110010, 101011, 111100, 100101, 001110, 010111). This LDPC code fragment represents a 3-bit message encoded as six bits. Redundancy is used, here, to increase the chance of recovering from channel errors. This is a (6, 3) linear code, with $n = 6$ and $k = 3$.

Once again ignoring lines going out of the picture, the parity-check matrix representing this graph fragment is

$$\mathbf{H} = \begin{pmatrix} 1&1&1&1&0&0 \\ 0&0&1&1&0&1 \\ 1&0&0&1&1&0 \end{pmatrix}.$$

In this matrix, each row represents one of the three parity-check constraints, while each column represents one of the six bits in the received codeword.

In this example, the eight code words can be obtained by putting the parity-check matrix **H** into this form through basic row operations:

From this, the generator matrix **G** can be obtained as (noting that in the special case of this being a binary code), or specifically:

$$\mathbf{G} = \begin{pmatrix} 1&0&0&1&0&1 \\ 0&1&0&1&1&1 \\ 0&0&1&1&1&0 \end{pmatrix}.$$

Finally, by multiplying all eight possible 3-bit strings by **G**, all eight valid code words are obtained. For example, the codeword for the bit-string '101' is obtained by:

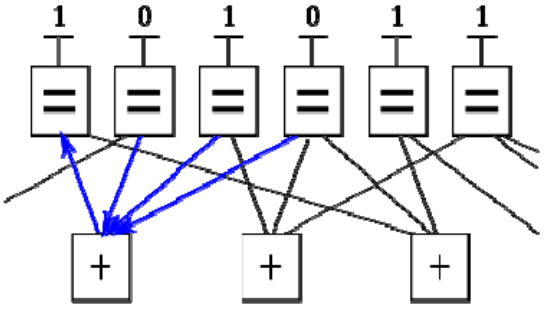
$$\begin{pmatrix} 1&0&1 \end{pmatrix} \cdot \begin{pmatrix} 1&0&0&1&0&1 \\ 0&1&0&1&1&1 \\ 0&0&1&1&1&0 \end{pmatrix} = \begin{pmatrix} 1&0&1&0&1&1 \end{pmatrix}.$$

Decoding

As with other codes, optimally decoding an LDPC code on the binary symmetric channel is an NP-complete problem, although techniques based on iterative belief propagation used in practice lead to good approximations. In contrast, belief propagation on the binary erasure channel is particularly simple where it consists of iterative constraint satisfaction.

For example, consider that the valid codeword, 101011, from the example above, is transmitted across a binary erasure channel and received with the first and fourth bit erased to yield ?01?11. Since the transmitted message must have satisfied the code constraints, the message can be represented by writing the received message on the top of the factor graph.

In this example, the first bit cannot yet be recovered, because all of the constraints connected to it have more than one unknown bit. In order to proceed with decoding the message, constraints connecting to only one of the erased bits must be identified. In this example, either the second or third constraint suffices. Examining the second constraint, the fourth bit must have been 0, since only a 0 in that position would satisfy the constraint.



This procedure is then iterated. The new value for the fourth bit can now be used in conjunction with the first constraint to recover the first bit as seen below. This means that the first bit must be a 1 to satisfy the leftmost constraint.

Thus, the message can be decoded iteratively. For other channel models, the messages passed between the variable nodes and check nodes are real numbers, which express probabilities and likelihoods of belief.

This result can be validated by multiplying the corrected

$$\mathbf{z} = \mathbf{H}\mathbf{r} = \begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}.$$

codeword **r** by the parity-check matrix **H**:

Because the outcome **z** (the syndrome) of this operation is the 3 × 1 zero vector, the resulting codeword **r** is successfully validated.

5. EXPERIMENTAL RESULTS:

On successful execution of the encoding techniques, we will display various operations done on client, TPA and the server.

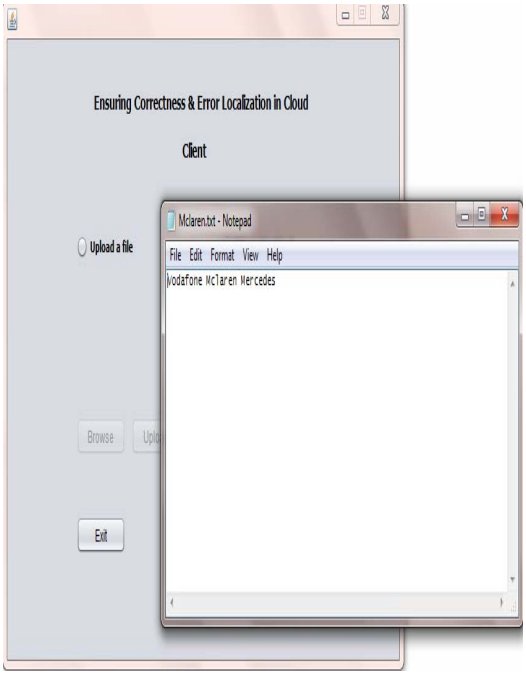


Fig 5.1: Client viewing the contents of the file stored on cloud

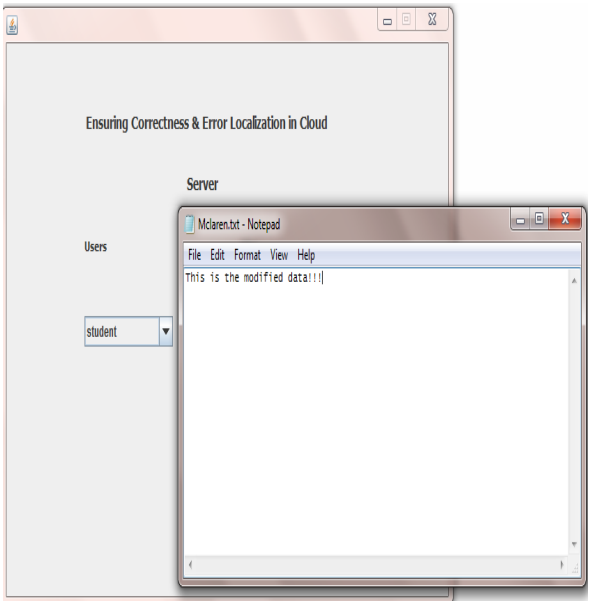


Fig 5.2: Server modifying the data

Once the file has successfully uploaded by the client, then client can download it by selecting down Client can perform various operations such as Upload a file & files in the cloud, view the files uploaded in the cloud. Download, Delete & Verify & correct are the objects used to see the files stored load option and then the user can download the files which are stored on to the cloud. This Screen Shot display the contents of the file upon downloading the particular file by the client.

A server can view the number of users subscribed and each of the individual users files which has been stored on the server. The server has opted to view the files of the user "student". The server upon opening the users file he modifies the content and save the file back as shown in the above screenshot.

Fig 5.3: Client viewing the saved file on the cloud

This Screen Shot displays the data that is modified by the server and the client downloads it and he does not know if the contents of the file are modified or not.

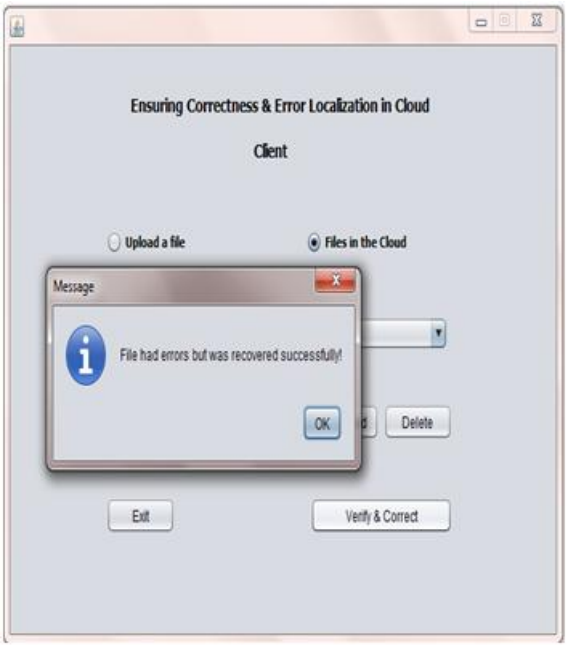


Fig 5.4: Recovery of the modified file.

This Screen Shot displays the data had errors and was recovered successfully. This is the result of Verify & Correct button.

Fig 5.5: Recovery of the modified file.

This Screen Shot displays the recovered data file. This is the result of ok button.

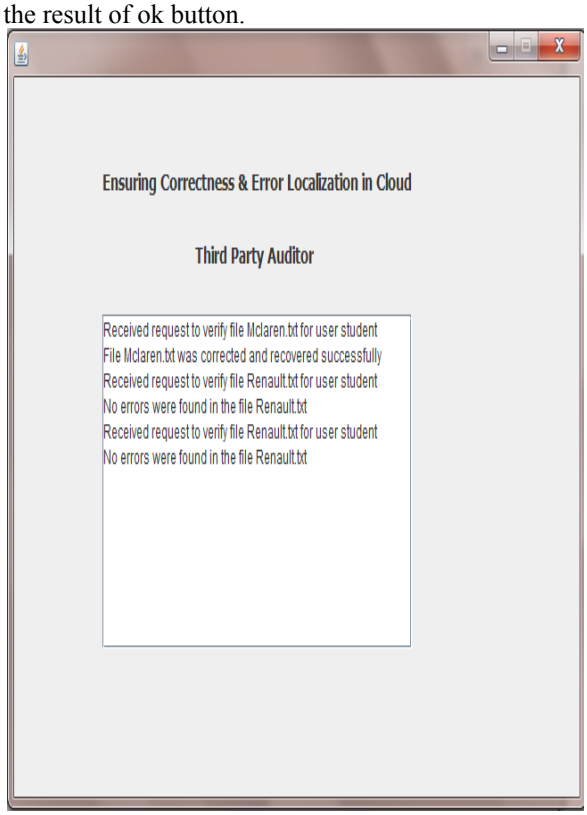


Fig 5.6: TPA Status after a user requested for Verify & Correct.

The main work of TPA is who has expertise and capabilities that users may not have, i.e. trusted to assess and expose risk of cloud storage services on behalf of the users upon request. This Screen Shot displays the information regarding the particular user requesting to verify the file. And would also display if the particular data file had errors or not.

6. CONCLUSION:

We investigated the problem of data security in cloud data storage, which is essentially a distributed storage system. To ensure the correctness of users' data in cloud data storage, we proposed an effective and flexible distributed scheme with explicit dynamic data support, including block update, delete, and append. We rely on erasure-correcting code in the file distribution preparation to provide redundancy parity vectors and guarantee the data dependability. 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). Through detailed security and performance analysis, we show that our scheme is highly efficient and resilient to Byzantine failure, malicious data modification attack, and even server colluding attacks

.Future Enhancement: We believe that data storage security in Cloud Computing, an area full of challenges and of paramount importance, is still in its infancy now, and many research problems are yet to be identified. We envision several possible directions for future research on this area. The most promising one we believe is a model in which public verifiability is enforced. Public verifiability, supported in allows TPA to audit the cloud data storage without demanding users' time, feasibility or resources. An interesting question in this model is if we can construct a scheme to achieve both public verifiability and storage correctness assurance of dynamic data. Besides, along with our research on dynamic cloud data storage, we also plan to investigate the problem of fine-grained data error localization.

REFERENCES

[1]. Amazon.com, "Amazon Web Services (AWS)," Online at http://aws.amazon.com, 2008.

[2]. [2] N. Gohring, "Amazon's S3 down for several hours," Online at http://www.pcworld.com/businesscenter/article/142549/amazons s3 down for several hours.html, 2008.

[3] A. Juels and J. Burton S. Kaliski, "PORs: Proofs of Retrievability for Large Files," Proc. of CCS '07, pp. 584–597, 2007.

[4] H. Shacham and B. Waters, "Compact Proofs of Retrievability," Proc. of Asiacrypt '08, Dec. 2008.

[5] K. D. Bowers, A. Juels, and A. Oprea, "Proofs of Retrievability: Theory and Implementation," Cryptology ePrint Archive, Report 2008/175, 2008, http://eprint.iacr.org/.

[6] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song, "Provable Data Possession at Untrusted Stores," Proc. Of CCS '07, pp. 598–609, 2007.

[7] G. Ateniese, R. D. Pietro, L. V. Mancini, and G. Tsudik, "Scalable and Efficient Provable Data Possession," Proc. of SecureComm '08, pp. 1– 10, 2008.

[8] T. S. J. Schwarz and E. L. Miller, "Store, Forget, and Check: Using Algebraic Signatures to Check Remotely Administered Storage," Proc. of ICDCS '06, pp. 12–12, 2006.

[9] M. Lillibridge, S. Elnikety, A. Birrell, M. Burrows, and M. Isard, "A Cooperative Internet Backup Scheme," Proc. of the 2003 USENIX Annual Technical Conference (General Track), pp. 29–41, 2003.

[10] K. D. Bowers, A. Juels, and A. Oprea, "HAIL: A High-Availability and Integrity Layer for Cloud Storage," Cryptology ePrint Archive, Report 2008/489, 2008, http://eprint.iacr.org/.

[11] L. Carter and M. Wegman, "Universal Hash Functions," Journal of Computer and System Sciences, vol. 18, no. 2, pp. 143–154, 1979.

[12] J. Hendricks, G. Ganger, and M. Reiter, "Verifying Distributed Erasurecoded Data," Proc. 26th ACM Symposium on Principles of Distributed Computing, pp. 139–146, 2007.

[13] J. S. Plank and Y. Ding, "Note: Correction to the 1997 Tutorial on Reed-Solomon Coding," University of Tennessee, Tech. Rep. CS-03- 504, 2003

.