FLEXIBLE DISTRIBUTED STORAGE INTEGRITY AUDITING MECHANISM TO ACHIEVE A SECURE AND DEPENDABLE CLOUD STORAGE SERVICE IN CLOUD COMPUTING

N Chandrakala

Christu Jyothi Institute of Technology and Science,Jangaon,TS,India.

Abstract: *Cloud storage allows users to remote storage of their data and enables on-demand high quality cloud applications without the local hardware and software management. Though this type of storage have many benefits, such a service is also suffers with the problems like physical possession of their outsourced data, which causes new security risks toward the correctness of the data in cloud. In order to remove this new problem and to obtain a secure and dependable cloud storage service, this paper proposes a flexible distributed storage integrity auditing mechanism, which utilises the homomorphism token and distributed erasure-coded data. The proposed method ensures users to audit the cloud storage with very lightweight communication and computation cost. This type of auditing mechanism ensures strong cloud storage correctness guarantee and also achieves fast data error localization. If the cloud data is dynamic, the proposed method further supports secure and efficient dynamic operations on outsourced data, including block modification, deletion, and append. The analysis shows the proposed scheme is efficient and resilient against byzantine failure, malicious data modification attack, and server colluding attacks.*

1. Introduction:

Cloud computing has the potential to dramatically change the landscape of the current IT industry (Armbrust et al., Goldberg 1989). It is the most powerful processor and even cheaper which is an internet based technology with Infrastructure as a service (IaaS) computing architecture .For companies that only have to process large amount of data occasionally running their own data centres is obviously not an option .Cloud computing has emerged as a promising approach to rent a large IT infrastructure on a short term pay-per-usage basis .Amazon simple storage service (S3) and Amazon Elastic Compute Cloud. Moving the data in the cloud provide the user with a great efficiencies, since it don't have to care about the complexities of direct hardware management .By internet based online services, using the huge amount of storage space and customizable computing resources, this platforms eliminates the responsibilities of the local machines for data maintenance . So for the integrity of the data users are depended on the cloud service providers. Vendors like Amazons S3 is such an example [2]

Generally users‟ data is protected using different cryptographic algorithms, in order to protect data from intruders .This approach also used in cloud computing environment. By using the algorithms the user loses control over the data. Therefore without the knowledge of the data in cloud we should provide security. So for the long term data safety these methods become even more challenging for the correctness f the data in the cloud. Secondly in cloud the data is not static, the data is updated, deleted inserted, appended. So the dynamic data update is the most important challenge in cloud .In previously in cloud only static operations are done, so by using dynamic operations in the cloud entails to new solution. The third challenge is that the data should be stored multiple locations so that we can reduce data integrity threats. The deployment in the cloud computing is powered by data centres running simultaneous, cooperated and distributed manner [1]. So the distributed data in the cloud is most important for storage correctness of data in the cloud and also to achieve robust and secure cloud data in the real world. In the existing system the ensuring data integrity has been highlighted by the following researchers [3]-[5].These techniques ensures the storage correctness in the cloud but it can’t address all the security threats in the cloud, since they are focus on single server scenario and most them uses static data it does not consider dynamic data pupation .

2. System Analysis

After analysing the requirements of the task to be performed, the next step is to analyze the problem and understand its context. The first activity in the phase is studying the existing system and other is to understand the requirements and domain of the new system. Both the activities are equally important, but the first activity serves as a basis of giving the functional specifications and then successful design of the proposed system. Understanding the properties and requirements of a new system is more difficult and requires creative thinking and understanding of existing running system is also difficult, improper understanding of present system can lead diversion from solution.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 trust worthy. This unique attribute, however, poses many new security challenges which have not been well understood like no user data privacy, Security risks towards the correctness of the data in cloud.

3 Proposed System design :

We focus on 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 holomorphic token with distributed verification of erasure-coded data, 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. 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.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, 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.

In this paper we are using Reed-Solomon erasure-correcting code to create k redundancy parity vectors from m data vectors in such a way that the original m data vectors can be reconstructed from any m out of the m+k vectors on a different server, the original data file can survive the failure of any k of m+k servers without any data loss, with a space overhead of k/m. For support of original file layout I/O our file layout is systematic, i.e., the unmodified m data file vectors together with k parity vectors are distributed across m+k different servers. Linear m/n codes are defined over Galois field with 2f elements. Let F=(F1 ,F2, ………..,Fm) and Fi=(f1i ,f2i,………fmi)T (i € {1,…….,m}),where l≤2p-1.Code defined by a generator matrix G with m rows and n columns G has a special form (Im |P) Im is an identity matrix .Every m×m identity matrix and G is invertible Generation of generator matrix with n=2f GF elements in a given order: f1, f2,…,fm..It form Vandermonde matrix.

Store vi s locally 12 end procedure The element vi (j) belongs to GF (2p) with small size, it is the response where the user expect from the server when he challenges in a specified blocks. The user keeps the token locally or in encrypted form of the pre-computed tokens in the cloud server. But in our case the user stores the tokens locally, to avoid lower bandwidth which occur due to dynamic data operation

The final step before file distribution is to blind each parity block gi(j) in (G(m+1),…………,G(n)) by gi(j) ← gi(j) +fkj (sij) , i € {1,……….,l}, where kj is the secret key for parity vector .This key is used for protection. After blinding the parity information, the user disperses all the encoded vectors G(j) (j €{1,….,n}) across the cloud servers S1,S2,……,Sn .

3.2 Correctness Verification and Error Localization Correctness

Verification and Error localization is the main idea for eliminating the data errors in the system. In the existing system they do not explicitly consider the error localization, thus they give only binary results. But in this scheme we get find the misbehaving servers by using challenge –response protocol Specially, the procedure of the i-th challenge-response for a cross-check over the n servers is described as follows : 1) The user reveals the άi as well as the i-th permutation key kprp(i) to each servers. 2) The server storing vector G(j) aggregates those r rows specified by index kprp(i) into a linear combination 3) Upon receiving Ri (j) all the servers, the user takes away blind values in R(j) (j € {m+ 1, . . . , n}) 4) Then the user verifies the code word with the received value which is generated but the secret matrix P. As the entire server operates over the same subset of indices .So these row specified code word should be in the encoded file. If the above equation holds the challenge is passed otherwise it indicates that the specified row has an error which is passed to the server that there is a corruption

The file matrix is symmetric, so the user can get the original file by downloading the data vectors from the first m servers, assume that they return the correct response value .Generally our verification is random spot checking that the storage correctness is probabilistic .By choosing system parameters (e.g., r, l, t) appropriately and conducting enough times of verifications, we can guarantee the successful file retrieval with high probability 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, which will be discussed shortly. Therefore, the user can always ask server to send back blocks of the r specified rows in the challenge and regenerate the correct blocks. The newly recovered file should be sent to the misbehaving servers to maintain the correctness of storage

A quality output is one, which meets the requirements of the end user and presents the information clearly. In any system results of processing are communicated to the users and to other system through outputs. In output design it is determined how the information is to be displaced for immediate need and also the hard copy output. It is the most important and direct source information to the user. Efficient and intelligent output design improves the system’s relationship to help user decision-making.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. We can guarantee the successful file retrieval with high probability. 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).As discussed in our architecture, in case the user does not have the time, feasibility or resources to perform the storage correctness verification, he can optionally delegate this task to an independent third party auditor, making the cloud storage publicly verifiable. However, as pointed out by the recent work, to securely introduce an effective TPA, the auditing process should bring in no new vulnerabilities towards user data privacy. Namely, TPA should not learn user's data content through the delegated data auditing.

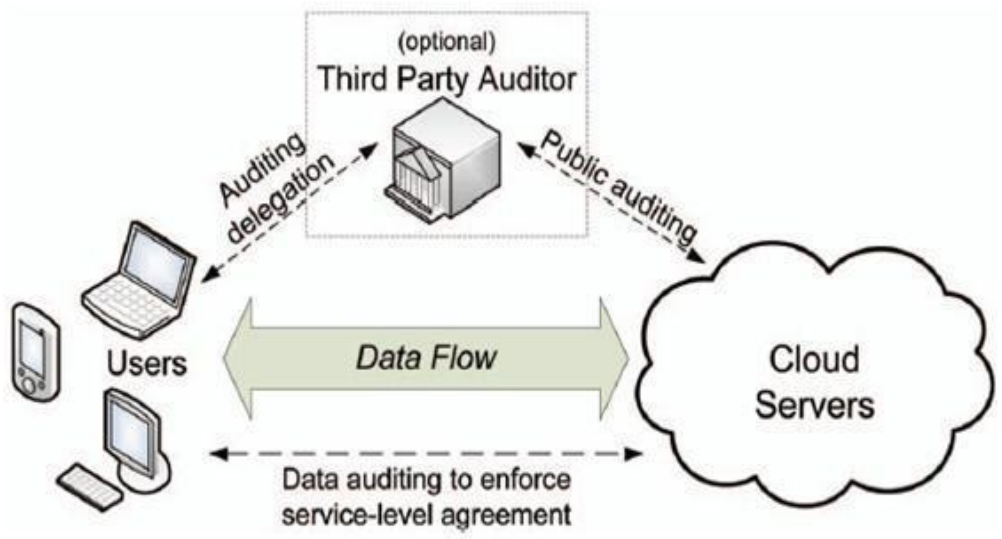


Fig. 1: Cloud storage service architecture

Representative network architecture for cloud storage service architecture is illustrated in Figure 1. Three different network entities can be identified as follows:

- User: an entity, who has data to be stored in the cloud and relies on the cloud for data storage and computation, can be either enterprise or individual customers.
- Cloud Server (CS): an entity, which is managed by cloud service provider (CSP) to provide data storage service and has significant storage space and com-putation resources (we will not differentiate CS and CSP hereafter.).
- 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. It is a simple graphical formalism that can be used to represent a system in terms of the input data to the system, various processing carried out on these data, and the output data is generated by the system.

Data Flow Diagram:

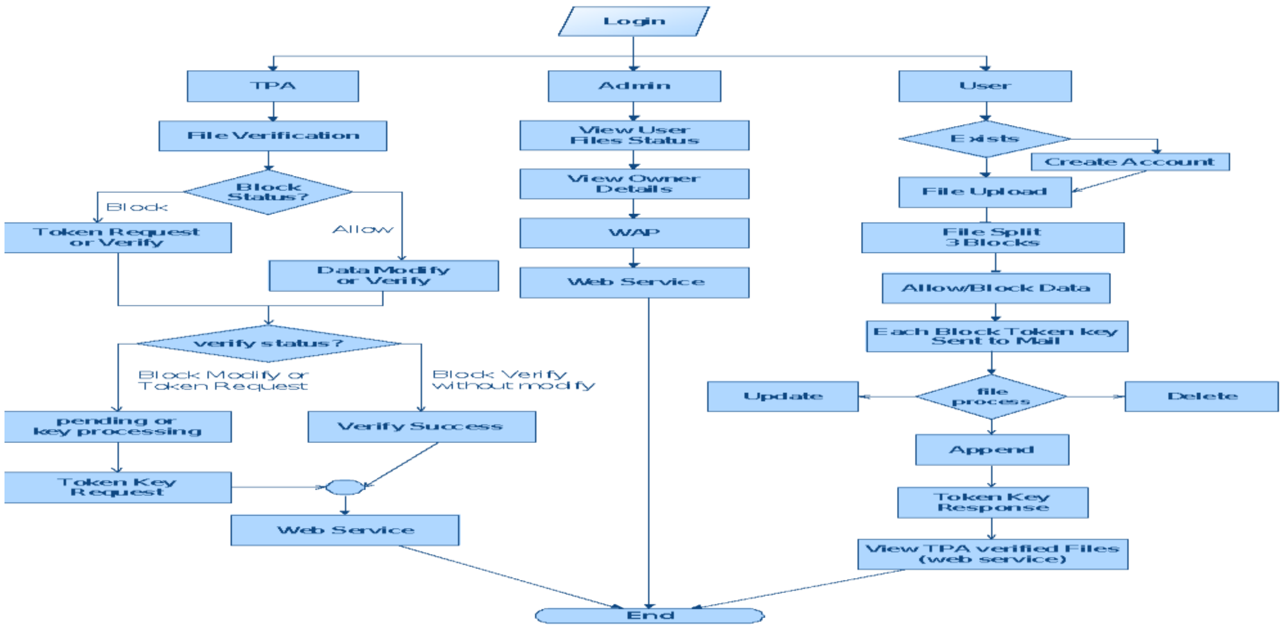


Fig:2 Data Flow Diagram:

5.Testing:

Software Testing can be defined as: Testing is an activity that helps in finding out bugs/defects/errors in a software system under development, in order to provide a bug free and reliable system/solution to the customer. In other words, let us consider an example as: suppose you are a good cook and are expecting some guests at dinner. we start making dinner; we make few very very very delicious dishes (off-course, those which you already know how to make). And finally, when you are about to finish making the dishes, you ask someone (or you yourself) to check if everything is fine and there is no extra salt/chili/anything, which if is not in balance, can ruin your evening (This is what called 'TESTING').

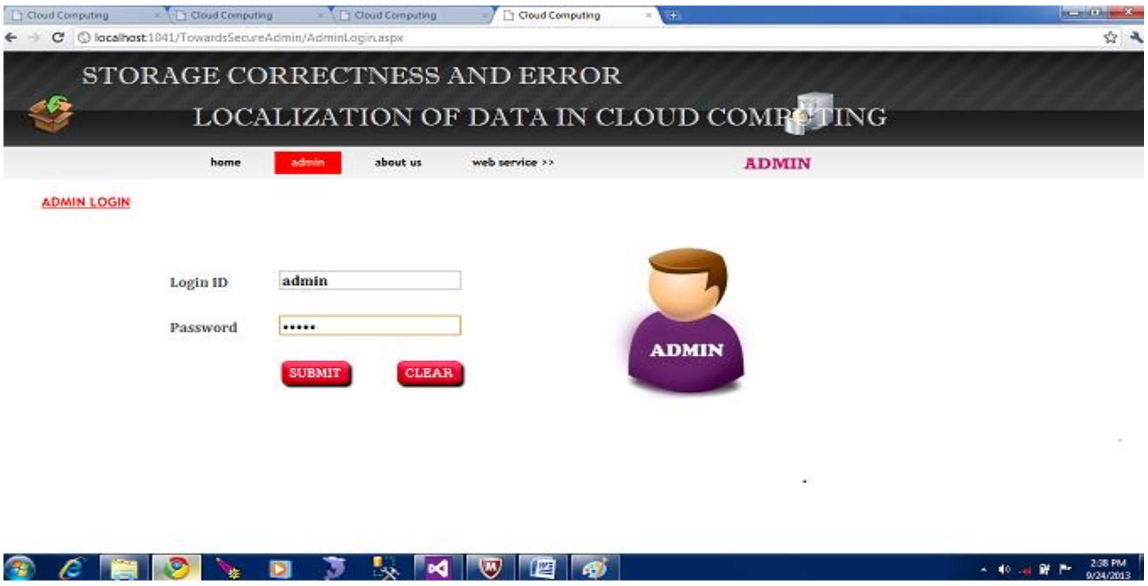
Software Testing Fundamentals:

1) Testing is a process of executing a program with the intent of finding an error.
2) A good test case is one that has a high probability of finding an as yet undiscovered error.
3) A successful test is one that uncovers an as yet undiscovered error.

Testing should systematically uncover different classes of errors in a minimum amount of time and with a minimum amount of effort. A secondary benefit of testing is that it demonstrates that the software appears to be working as stated in the specifications. The data collected through testing can also provide an indication of the software's reliability and quality. But, testing cannot show the absence of defect -- it can only show that software defects are present.

Well, while making food, its ok to have something extra, people might understand and eat the things you made and may well appreciate your work. But this isn't the case with Software Project Development. If you fail to deliver a reliable, good and problem free software solution, you fail in your project and probably you may lose your client. This can get even worse! So in order to make it sure, that you provide your client a proper software solution, you go for TESTING. You check out if there is any problem, any error in the system, which can make software unusable by the client.

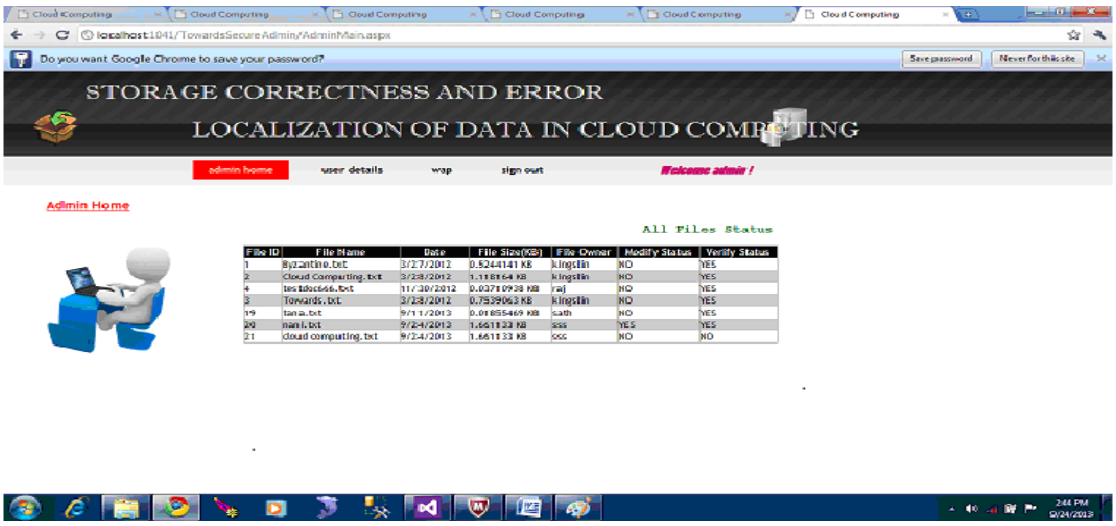
6.Output Screens


Cloud Computing
localhost:1041/TowardsSecureAdmin/AdminLogin.aspx
STORAGE CORRECTNESS AND ERROR
LOCALIZATION OF DATA IN CLOUD COMPUTING
home
admin
about us
web service >>
ADMIN
ADMIN LOGIN
SUBMIT
CLEAR
ADMIN
2:38 PM
9/24/2013

Login ID: admin

Password: •••••

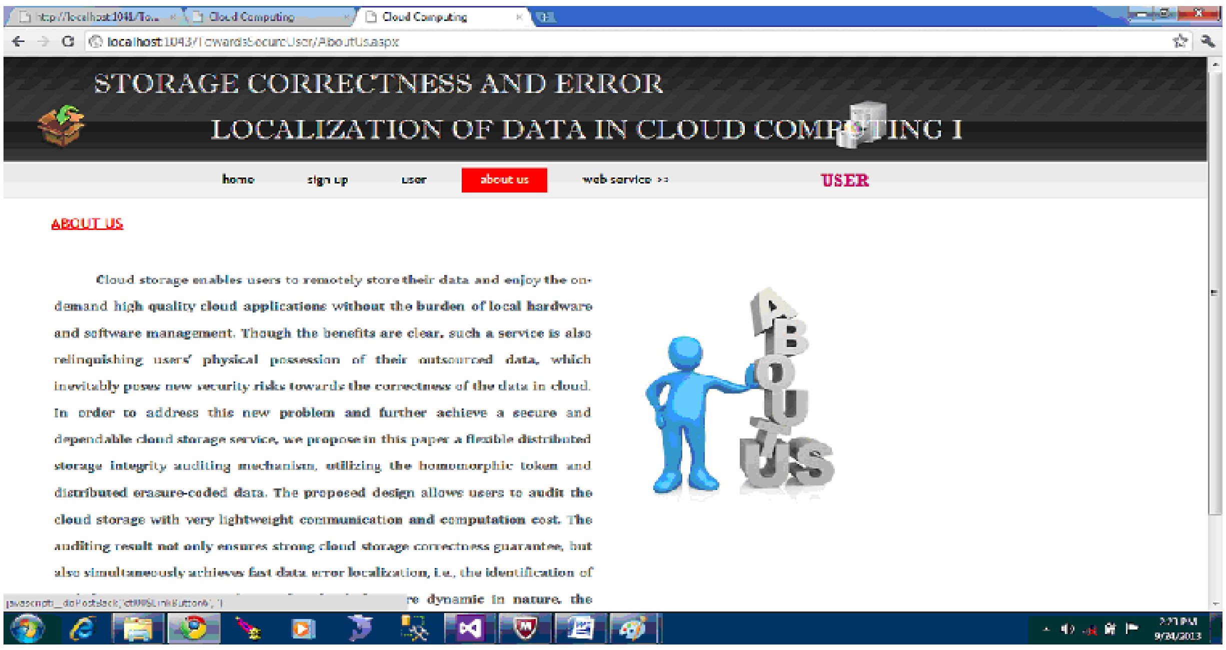
Fig 3: Admin Login Page.


Cloud Computing
localhost:1041/TowardsSecureAdmin/AdminMain.aspx
Do you want Google Chrome to save your password?
Save password
Never for this site
STORAGE CORRECTNESS AND ERROR
LOCALIZATION OF DATA IN CLOUD COMPUTING
admin home
user details
wap
sign out
Welcome admin !
Admin Home
2:44 PM
9/24/2013

All Files Status

File ID	File Name	Date	File Size(KB)	File Owner	Modify Status	Verify Status
1	Byzantine.txt	3/27/2012	0.5244141 KB	kingslin	NO	YES
2	Cloud Computing.txt	3/28/2012	1.118164 KB	kingslin	NO	YES
4	testdoc666.txt	11/30/2012	0.03710938 KB	raj	NO	YES
3	Towards.txt	3/28/2012	0.7539063 KB	kingslin	NO	YES
19	tana.txt	9/11/2013	0.01855469 KB	sath	NO	YES
20	nani.txt	9/24/2013	1.661133 KB	sss	YES	YES
21	cloud computing.txt	9/24/2013	1.661133 KB	sss	NO	NO

Fig 2: The admin home page with the file status and file uploaded owner name.


STORAGE CORRECTNESS AND ERROR
LOCALIZATION OF DATA IN CLOUD COMPUTING I
home
sign up
user
about us
web service >>
USER
ABOUT US

Cloud storage enables users to remotely store their data and enjoy the on-demand high quality cloud applications without the burden of local hardware and software management. Though the benefits are clear, such a service is also relinquishing users' physical possession of their outsourced data, which inevitably poses new security risks towards the correctness of the data in cloud. In order to address this new problem and further achieve a secure and dependable cloud storage service, we propose in this paper a flexible distributed storage integrity auditing mechanism, utilizing the homomorphic token and distributed erasure-coded data. The proposed design allows users to audit the cloud storage with very lightweight communication and computation cost. The auditing result not only ensures strong cloud storage correctness guarantee, but also simultaneously achieves fast data error localization, i.e., the identification of ... re dynamic in nature, the

Fig 3: About US Page.

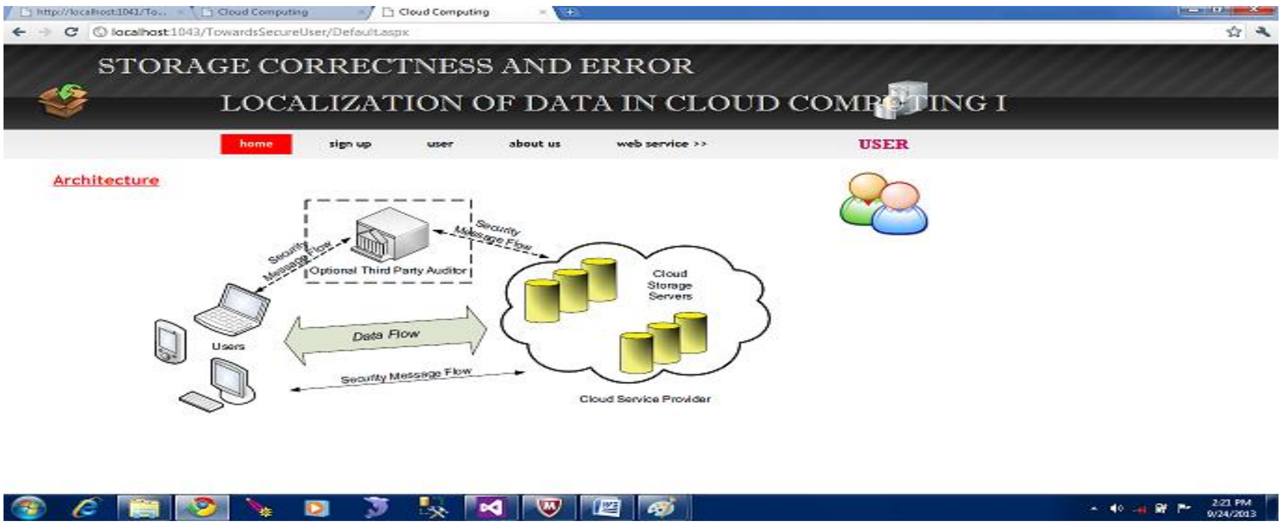
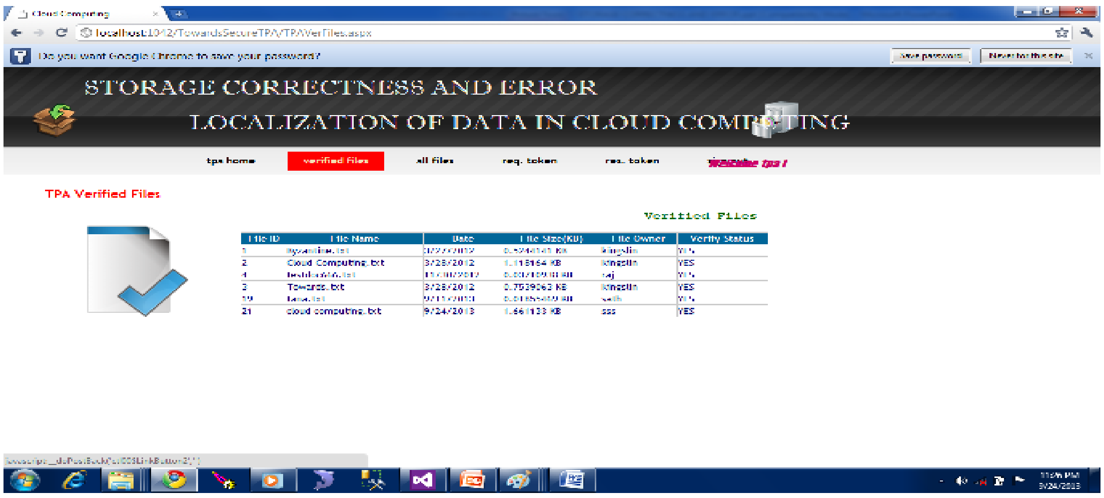


Fig 4: TPA Home Page : Views all the files which are be verified.

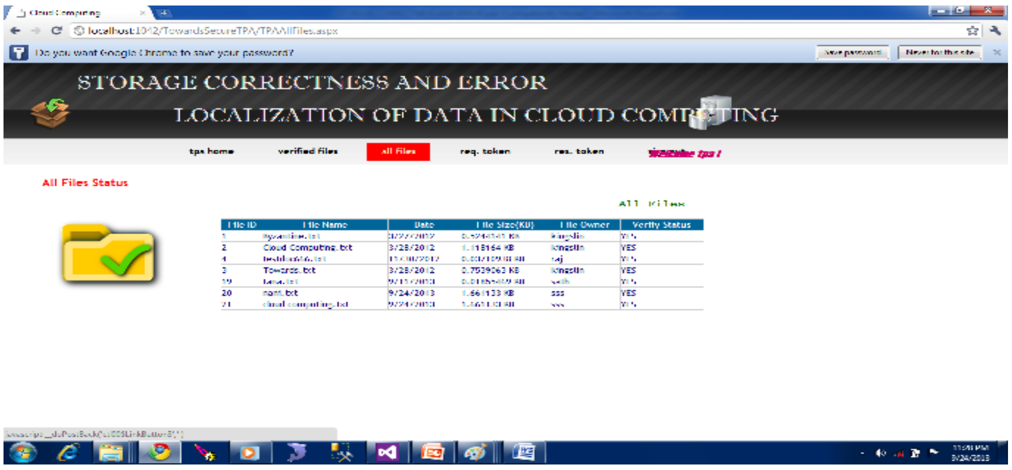

Cloud Computing
localhost:1042/TowardsSecureTPA/TPA/VerFiles.aspx
Do you want Google Chrome to save your password?
Save password
Never for this site
STORAGE CORRECTNESS AND ERROR
LOCALIZATION OF DATA IN CLOUD COMPUTING
tpa home
verified files
all files
req. token
res. token
Welcome tpa !

TPA Verified Files

Verified Files

File ID	File Name	Date	File Size(KB)	File Owner	Verify Status
1	Byzantine.txt	[illegible]	0.5244141 KB	kingslin	YES
2	Cloud Computing.txt	3/28/2012	1.118164 KB	kingslin	YES
4	[illegible]	[illegible]	[illegible]	raj	YES
3	Towards.txt	3/28/2012	0.7539063 KB	kingslin	YES
19	[illegible]	[illegible]	[illegible]	[illegible]	YES
21	cloud computing.txt	9/24/2013	1.661133 KB	sss	YES

Fig 5: Verified files tab : Used to view the files which are verified by TPA.

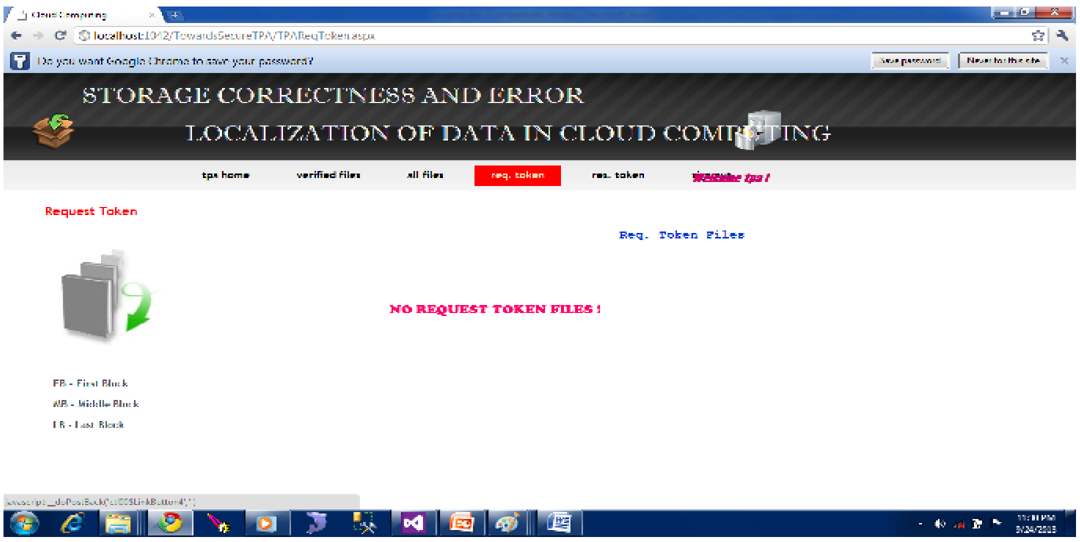

Cloud Computing
localhost:1042/TowardsSecureTPA/TPA/AllFiles.aspx
Do you want Google Chrome to save your password?
Save password
Never for this site
STORAGE CORRECTNESS AND ERROR
LOCALIZATION OF DATA IN CLOUD COMPUTING
tpa home
verified files
all files
req. token
res. token
Welcome tpa !

All Files Status

All Files

File ID	File Name	Date	File Size(KB)	File Owner	Verify Status
1	Byzantine.txt	[illegible]	0.5244141 KB	kingslin	YES
2	Cloud Computing.txt	3/28/2012	1.118164 KB	kingslin	YES
4	[illegible]	[illegible]	[illegible]	raj	YES
3	Towards.txt	3/28/2012	0.7539063 KB	kingslin	YES
19	[illegible]	[illegible]	[illegible]	[illegible]	YES
20	nant.txt	9/24/2013	1.661133 KB	sss	YES
21	cloud computing.txt	9/24/2013	[illegible]	sss	YES

Fig 6: all files tab: displays all the files both verified and not verified.


Cloud Computing
localhost:1042/TowardsSecureTPA/TPA/ReqToken.aspx
Do you want Google Chrome to save your password?
Save password
Never for this site
STORAGE CORRECTNESS AND ERROR
LOCALIZATION OF DATA IN CLOUD COMPUTING
tpa home
verified files
all files
req. token
res. token
Welcome tpa !

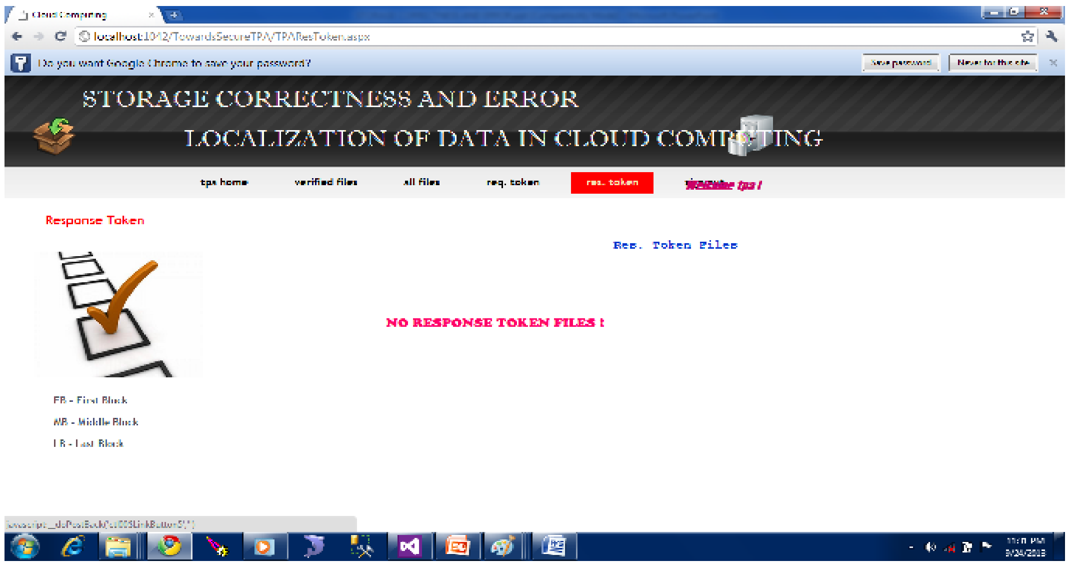
Request Token

Req. Token Files

NO REQUEST TOKEN FILES !

FB - First Block
MB - Middle Block
LB - Last Block

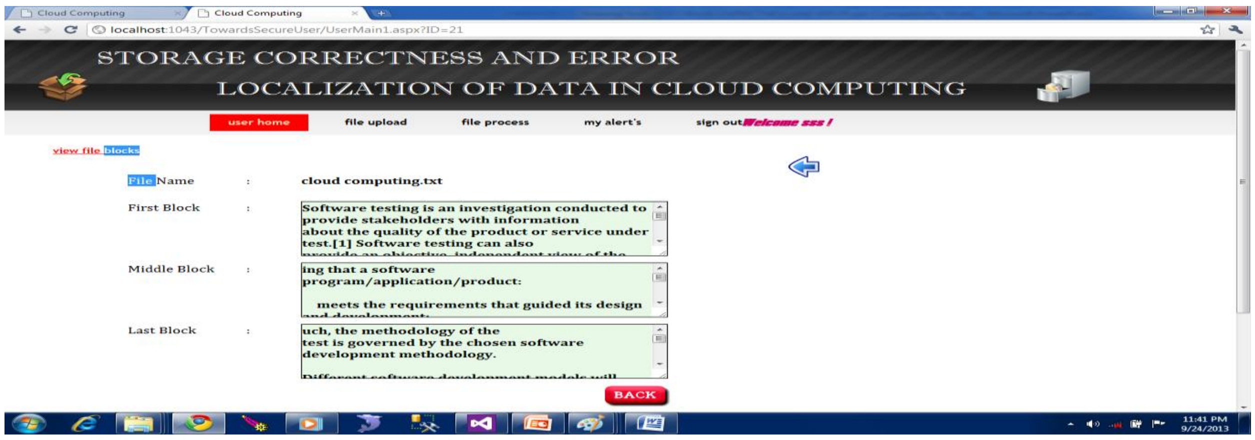
Fig 6 : Request Token tab: Shows all the request token for the files.



Res. Token Files

NO RESPONSE TOKEN FILES !

Fig 7: Response Token: shows the entire response token.



File Name : cloud computing.txt

First Block : Software testing is an investigation conducted to provide stakeholders with information about the quality of the product or service under test.[1] Software testing can also

Middle Block : ing that a software program/application/product:

meets the requirements that guided its design

Last Block : uch, the methodology of the test is governed by the chosen software development methodology.

Fig 5 View file block: After the Click on the view link at the user home page shows the file with blocks division.

7.Conclusion :

In this paper, we studied the problem of data security in data storage in cloud servers. To guarantee the correctness of users‟ data in cloud data storage, we proposed an effectual and flexible scheme with explicit dynamic data support, including block revise, erase, and affix. We use erasure-correcting code in the file distribution preparation to provide redundancy parity vectors and guarantee the data dependability. Our scheme accomplishes the integration of storage correctness insurance and data corruption has been detected during the storage correctness verification across the distributed servers. Our scheme is highly efficient and resilient to Byzantine failure, malicious data modification attack, and even server colluding attacks. We believe that data storage security in Cloud Computing is an area full of challenges and of dominance. We also focused on theoretical analysis of parallel data processing in cloud computing for scheduling the cloud server time.

Growing number of companies have to process huge amounts of data in a cost-efficient manner. Classic representatives for these companies are operators of Internet search engines. The vast amount have to deal with every day has made traditional database solutions prohibitively expensive .Instead, these companies have popularized an architectural paradigm based on a large number of commodity Consequently, the allocated compute resources may be inadequate for big parts of the submitted job and unnecessarily increase processing time and cost. The opportunities and challenges for efficient parallel data processing in clouds and present our research project. It is the first data processing framework to explicitly exploit the dynamic resource allocation offered by today‟s IaaS clouds for both, task scheduling and execution. Particular tasks of a processing job can be assigned to different types of virtual machines which are automatically instantiated and terminated during the job execution. In recent years a variety of systems to facilitate MTC has been developed. Although these systems typically share common goals (e.g. to hide issues of parallelism or fault tolerance), they aim at different fields of application. Map Reduce is designed to run data analysis jobs on a large amount of data, which is expected to be stored across a large set of share-nothing commodity servers. Once a user has fit his.

References:

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

[2] N.Gohring, “Amazon‟s S3 down for several hours,” Online at http://www.pcworld.com/businesscenter/article/142549/amazons s3down for several hours.html, 2008

[3] A. Juels and J. Burton S. Kaliski, “PORs: Proofs of Retrievability forLarge 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. ofCCS ’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] “A Cooperative Internet Backup Scheme,” *Proc. of the 2003 USENIXAnnual 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 Erasure coded 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.