A Novel Approach to Trustable Data Storage in Cloud Computing

Vegi Srinivas, V. Valli Kumari

Abstract*: With the advent of new platforms on computing techniques, the cloud is widely accepted and adoptable environment. Many cloud applications demand ease of use, speed, and fault tolerance over consistency. Though the benefits are clear, such a service is also hand over users physical control of their outsourced data, which inevitably poses new security risks toward 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 distributed ensure-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 identification block of errors, but also simultaneously achieves fast data error localization, i.e., the identification of misbehaving server.

Index Terms: Cloud computing, data integrity, distributed storage, error localization.

I. INTRODUCTION

Cloud Computing has been envisioned as the next generation architecture of IT enterprise, due to its long list of unprecedented advantages in the IT history. On-demand self-service, ubiquitous network access, location independent resource pooling, rapid resource elasticity, usage-based pricing and transference of risk [1]. 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 into the Cloud. From users perspective, including both individuals and IT enterprises, storing data remotely into the cloud in a flexible on-demand manner brings appealing benefits: relief of the burden for storage management, universal data access with independent geographical locations, and avoidance of capital expenditure on hardware, software, and personnel maintenances, etc [2]. 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.

Manuscript Received on July 2014.

Vegi Srinivas, CSE, Dadi Institute of Engineering and Technology, Anakapalle, India.

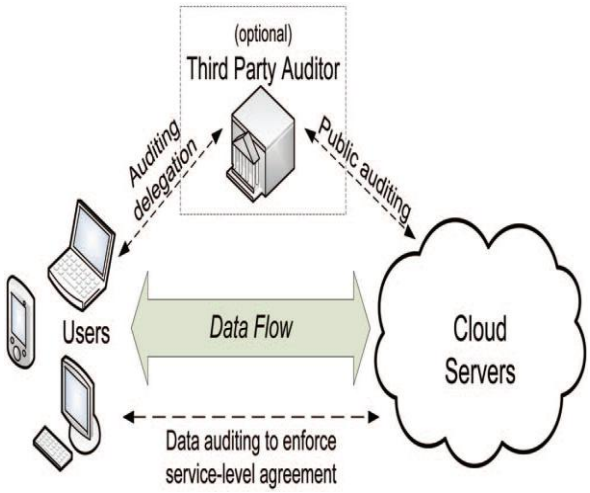
Dr. V. Valli Kumari, Professor, CS&SE, AU College of Engineering, Andhra University, Visakhapatnam, India.

Examples of outages and security breaches of noteworthy cloud services appear from time to time [3][4][5]. Secondly, for the benefits of their own, there do exist various motivations for cloud service providers to behave unfaithfully towards the cloud users regarding the status of their outsourced data. Examples include cloud service providers, for monetary reasons, reclaiming storage by discarding data that has not been or is rarely accessed, or even hiding data loss incidents so as to maintain a reputation [6][7][8]. In short, although outsourcing data into the cloud is economically attractive for the cost and complexity of long-term large-scale data storage, it does not offer any guarantee on data integrity and availability. This problem, if not properly addressed, may impede the successful deployment of the 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. Thus, how to efficiently verify the correctness of outsourced cloud data without the local copy of data files becomes a big challenge for data storage security in Cloud Computing. Note that simply downloading the data for its integrity verification is not a practical solution due to the expensiveness in I/O cost and transmitting the file across the network. Besides, it is often insufficient to detect the data corruption when accessing the data, as it might be too late for recover the data loss or damage. Considering the large size of the outsourced data and the user's constrained resource capability, the ability to audit the correctness of the data in a cloud environment can be formidable and expensive for the cloud users [8], [9]. Therefore, to fully ensure the data security and save the cloud users' computation resources, it is of critical importance to enable public audit ability for cloud data storage so that the users may resort to a third party auditor (TPA), who has expertise and capabilities that the users do not, to audit the outsourced data when needed. Based on the audit result, TPA could release an audit report, which would not only help users to evaluate the risk of their subscribed cloud data services, but also be beneficial for the cloud service provider to improve their cloud based service platform[10].

II. RELATED WORK

From user's perspective, the adversary model has to capture all kinds of threats toward his cloud data integrity. Because cloud data do not reside at user's local site but at CSP's address domain, these threats can come from two different sources: internal and external attacks. For internal attacks, a CSP can be self-interested, untrusted, 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. For external attacks, data integrity threats may come from outsiders who are beyond the control domain of CSP, for example, the economically motivated attackers. They may compromise a number of cloud data storage servers in different time intervals and subsequently be able to modify or delete users' data while remaining undetected by CSP. 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. The simplest Proof of irretrievability (POR) scheme can be made using a keyed hash function hk(F). In this scheme the verifier, before archiving the data file F in the cloud storage, pre-computes the cryptographic hash of F using hk(F) and stores this hash as well as the secret key K. To check if the integrity of the file F is lost the verifier releases the secret key K to the cloud archive and asks it to compute and return the value of hk(F). By storing multiple hash values for different keys the verifier can check for the integrity of the file F for multiple times, each one being an independent proof. The traditional architecture contains basic three roles data owner, auditor and user as follows.



In this paper we are proposing an efficient mechanism i.e novel signature for authentication for error recovery and for the data integrity we implemented an efficient file segmentation method for error correctness and for providing the language interoperability we implemented our application in service oriented application

III. PROPOSED WORK

Our work proceeds with the data integrity, data correctness and with language interoperability. Our mechanism implements an efficient authentication code for individual block for error detection of the block, and thirdly our architecture runs as service oriented application.

Novel Secure Architecture:

Our architecture contains various roles 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 computation resources (we will not differentiate CS and CSP hereafter).

Third-Party Auditor: 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.

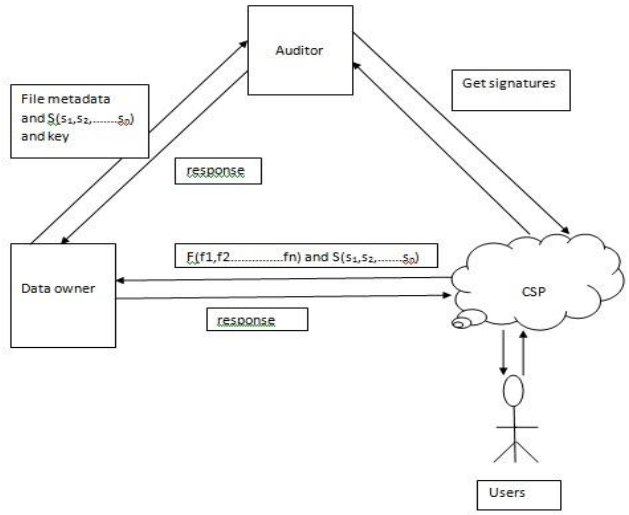


Fig. 1 Architecture for Data Correctness Over Cloud

In our approach data owner applies signature mechanism on individual blocks of the content and generates the hash code and encrypts the content with Rijandael algorithm and uploads in to the server and forward the file meta data information and key to the third party auditor ,there auditor performs same signature mechanism and generates signature on the blocks and then check the both signatures if any block code is mismatched that can be intimated to the data owner, then administrator can forward only the corrected information instead of total content. User can access the information which is provided by the cloud service.

IV. DATA CORRECTNESS AND ERROR DETECTION

In order to accomplish this task we have devised an algorithm which uses block signature method to identify the exact block error. A new block signature strategy is proposed in this paper to know the exact location of error. We call this error free transfer technique. The above algorithm generates signatures against the data in file and appends those generated signatures at the end of file.

It is very obvious from the algorithm generates signatures for every block separately and then those signatures are appended at the end of file as well. This algorithm uses 16 bytes as blocks reserved bytes. These bytes are used to send the original size of the file. Block size in this algorithm (n) is dependent upon the preference of users. The method of identifying corruption at the Technique receiving site uses the similar technique. The algorithm at receiving site first identifies the actual size of the file received. Then it separates the signatures from the received file. After doing this process file only contains the original data with appended zeros and 16 reserved .The signatures are separated the file. This algorithm then again generates signatures with received original file and compares the signatures with received signatures. If signatures exactly match, it means the file is received without errors. If match is not found, it means that the file is corrupted. One very strong point about the proposed algorithm I - Calculate Length of(F1) is that it first divides the whole file into blocks of equal count *- 1/n size. Signature for each block is separately generated for j =1 to count and stored in the file. It means that the number of S <- 0 n blocks in the file is exactly equal to the number of signatures generated. That is, each signature represents signatures of the file received after removing sending site signatures from the file. The signatures generated Fn *- F1l Sig at sending site are then matched against the signature generated against the receiving site. Matching of match is found, it means that the block is received accurately. The mask us capable of corrupted. After the identification of corrupted blocks, our receiving side asks sending side only for those blocks which are received corrupted.

V. NOVEL AUTHENTICATION BASED SIGNATURE

Algorithm: Generate File with Signatures

Input: User File in ASCII (Fo)
Output: User File with Signature appended at end of (Fn)
Method: In order to apply hash function on each n byte block of file which is corrupted? If we consider it with thefile we perform the following steps to make (m mod n)= 0 of Fo

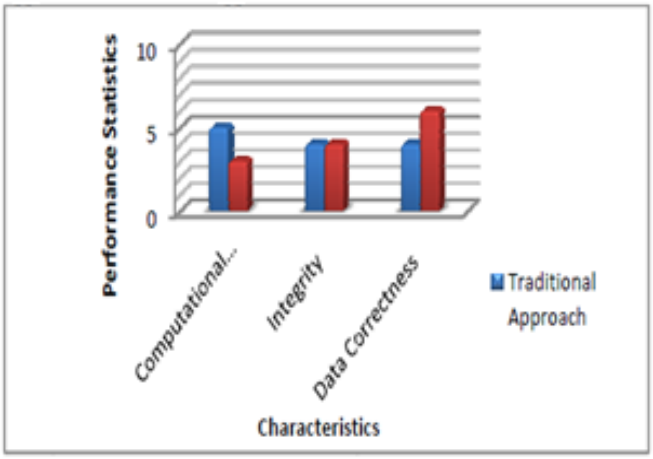
M← Calculate Length of (F0)
n ← Length of Block (any one of128/ 256 /512/ 1024 /204/4096/ 8192) bytes
res← reserved 16 bytes
P ←m mod n
Q ← n- (P + res)
if(Q > 0)
F←Append Q zeros at the end of F_0
Else if(Q < 0)
R ←n+ Q
F1 ←Append R zeros at the end of F_0
F1← Append res at the end of F_0
In order to generate Signatures of Fl, perform the following steps
I← Calculate_ Length of (F_l)
count ←l/n
For j←1 to count
S ← 0
S ←reverse[$\sum^{n}_{A=1}$((A XOR B) v (A ∩ B))]
Where B <- to_Integer (to_Char (A))
Sig ← Sig+ to-Binary (S)
Fn ←F1 + Sig

Comparative Analysis of Traditional Approach and Our Novel Approach:

Traditional token pre-computation method is complex.

- In traditional approach we are completely rely on third party auditor
- Previous approach follows the binary result approach, not with exact block identification
- Proposed approach is scalable and secure
- Remote data integrity achieves successfully in the proposed approach with optimal performance
- A novel signature based mechanism for error detection and data correctness
- Simple and secure signature generation and verification
- Our novel approach successfully provides the data correctness, Fast localization of data.

Performance evaluation as follows:



VI. CONCLUSION

Our approach is efficient during the segmentation and integration even it does not relives to the third party or auditor and error detection mechanism inform to the data owner whenever the correctness is failed with efficient signature authentication mechanism. The trust establishment mechanism we proposed here is more reliable to minimize the vulnerabilities in cloud storage data. The simulation of the process can be shown in an efficient way.

REFERENCES

1. C. Wang, Q. Wang, K. Ren, and W. Lou, “Ensuring Data StorageSecurity 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 TransparentSecurity, ”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.
6. J. Kincaid, "MediaMax/TheLinkup Closes Its Doors," http://www.techcrunch.com/2008/07/10/mediamaxthelinkup-closesits-doors, July 2008.
7. Amazon.com, "AmazonS3 AvailabilityEvent:July20,2008, "http://status.aws.amazon.com/s3-20080720.html, July 2008.
8. S. Wilson, "Appengine Outage," http://www.cio-weblog.com/ 50226711/appengine_outage.php, June 2008.
9. B. Krebs, "Payment Processor Breach May BeLargest Ever," http://voices.washingtonpost.com/securityfix/2009/01/payment_processor_breach_may_b.html, Jan. 2009.
10. A. Juels and B.S. Kaliski Jr., "PORs: Proofs of Retrievability forLarge Files," Proc. 14th ACM Conf. Computer and Comm. Security (CCS '07), pp. 584-597, Oct. 2007.

AUTHORS PROFILE

Vegi Srinivas, Received his M.Sc. and M.Tech., from Andhra University in Computer Science and Technology. He is Currently Working as Associate Professor in Dadi Institute of Engineering and Technology, Anakapalli, Visakhapatnam, India. His main areas of interests are Security, Privacy and Trusted Computing. He is a Member of IEEE, ACM, and Life Member of ISTE, CSI.

V. Valli Kumari, Received her B.E. in Electronics and Communication Engineering and M.Tech. and PhD in Computer Science and Engineering all from Andhra University, India and is Currently Working as Professor in the Same Department. Her research interests include Security and Privacy Issues in Data Engineering, Network Security and E-Commerce. She is a member of IEEE and ACM and is a fellow of IETE.