IJCSEC- International Journal of Computer Science and Engineering Communications.
Vol.3, Issue 2, 2015, Page.885-888, ISSN: 2347–8586

Fine Grained updates with Consignment Auditing of Dynamic Bigdata Storage on Cloud

G.Navaneetham[1] ,K.Gandhimathi[2]

[1]PG Student [2]Asst. Prof Dept. of CSE,
[1&2]Dept. of Computer Science and Engineering
Idhaya Engineering College for Women
navanee2208@gmail.com, pk.gmathi@gmail.com

ABSTRACT: The users may need to split the large-scale datasets into smaller chunks before uploading to the cloud for privacy-preserving. In this regard, efficiency in processing small updates is always essential in big data applications. And the data integrity verification is done by a trusted third party where the process is called as Third-Party Auditing (TPA). It investigates the problem of integrity verification for big data storage in cloud and do better support for small dynamic updates. To provide Fine-grained data updates that can fully support authorized auditing and Fine-grained update requests. It utilizes a flexible data segmentation strategy and a Ranked Merkle Hash Tree (RMHT). An additional authorization process is added among the three participating parties of client, Cloud Storage Server (CSS) and a third-party auditor (TPA). The auditing can be performed for a consignment of files, where the authorized auditor performs the auditing for a set of files at a time.

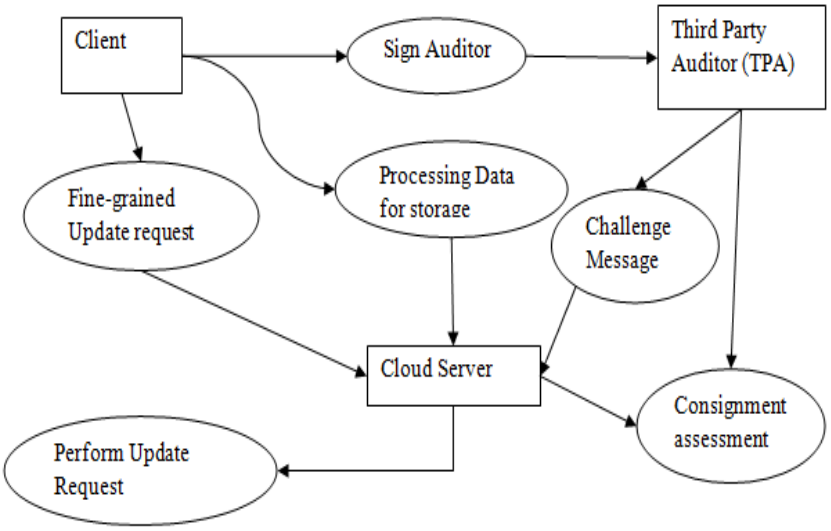
Index Terms: Third Party Auditing, Fine Grained updates, Consignment auditing.

1 INTRODUCTION

As internet users are increasing every year all the work gets digitalized today, as a result the data in the internet gets increased. Big Data is a term refers a large dataset. It consists of both structured and unstructured data. Unstructured data refers information that is not able to fit in row-column database. Social network like Facebook, Twitter, linkdln, Wikipedia and YouTube are some of the source for production of large amount of unstructured data. Big Data cannot be treated by existing database application. Some of the NoSQL database like Mongodb, MarkLogic, Apache Hadoop, Apache Cassandra, and IBMpureXML are used to analyze an unstructured data.Public auditing schemes can support full data dynamics. In their models, only insertions, deletions and modifications on fixed-sized blocks are discussed. Particularly, in BLS-signature-based schemes with 80-bit security, size of each data block is either restricted by the 160-bit prime group order, as each block is segmented into a fixed number of 160-bit sectors. This design is inherently unsuitable to support variable-sized blocks, despite their remarkable advantage of shorter integrity proofs. To provide insertion, deletion or modification of one or multiple fixed-sized blocks, which we call 'coarse-grained' updates. Although support for coarse-grained updates can provide an integrity verification scheme with basic scalability, data updating operations in practice can always be more complicated.A modification that can dramatically reduce communication overheads for verifications of small updates. Theoretical analysis and experimental results have demonstrated that our scheme can offer not only enhanced security and flexibility, but also significantly lower overheads for big data applications with a large number of frequent small updates such as applications in social media and business transactions.

2 SYSTEM MODEL

It utilizes a flexible data segmentation strategy and a ranked merkle hah tree, which supports variable-sized data blocks, authorized third-party auditing and fine-grained dynamic data updates. The client will generate keying materials then upload the data to CSS. The files segmented are of equal size and large as possible. Different from previous schemes, the client will store a RMHT instead of a MHT as metadata. The client will authorize the Third-party auditor (TPA). The CSS performs the client's fine-grained update requests and then the client checks whether CSS has performed the updates on both the data blocks and their corresponding authenticators honestly. The data integrity of the data stored on CSS is verified by TPA.



3 SECURITY ANALYSIS

Challenge and Verification: within the challenge/verification method of our theme, we have a tendency to try and secure the theme against a malicious CSS United Nations agency tries to cheat the friend TPA concerning the integrity standing of the client's knowledge, that is that the same as previous work on each PDP and POR. during this step, other than the new authorization method (which are mentioned intimately later during this section), the sole distinction compared to is that the RMHT and variable-sectored blocks. Therefore, the protection of this section is proved through a method extremely similar with, exploitation a similar framework, adversarial model and interactive games outlined in . an in depth security proof for this section is so omitted here.

3.1 TPA Authorization

Security of the new authorization strategy in our theme relies on the existential unforgeability of the chosen signature theme. we have a tendency to 1st outline the behavior of a malicious third-party auditor. A malicious TPA could be a third party United Nations agency aims at difficult a user's information keep on CSS for integrity proof while not this user's permission. The malicious TPA has access to the whole network. in keeping with this definition, none of the previous information auditing schemes is resilient against a malicious TPA.

3.2 Verifiable Data Updating

In the verifiable change method, the most somebody is that the shady CSS who failed to do the information update with success, however still manages to come back a satisfactory response to the shopper thenceforth.

4 TECHNIQUES

4.1 THIRD PARTY AUDITOR (TPA) AUTHORIZATION

Input: Request from client

Output: Encrypted ID of TPA

Client enters detail to fill registration form of cloud. Cloud provides ID and Password to the client. Client enters username and password while logging in, that has been verified, if the information is correct client is ready to store the data on cloud. Client generates an asymmetric key, Public key is used for encryption and secret key is used for decryption of the data block. TPA has its own ID, which is encrypted using the Cloud's public key. When client asks for TPA's ID, it sends the encrypted ID for security purpose. The client receives the Secure ID and provide signature, ie. Hash value is generated and sends that signature to the TPA along with the auditing delegate request for it to compose challenge later.

4.2 DATA PROCESSING FOR STORAGE

Input: Client data

Output: RMHT

The Rank Merkle Hash Tree (RMHT) is created for the segmented blocks. Similar to a binary tree, each node will have a maximum of 2 child nodes. The information contained in each node is the hash value for the data block and its rank.

4.3 FINE-GRAINED UPDATE

Input: Block level update request

Output: Verify update

Fine-grained update process between client and cloud service provider (CSS) has five types of updates such that insertion, deletion, modification, blocks splitting in each data block.

4.4 PROOF GENERATION AND VERIFICATION

Input: TPA ID and Data request

Output: Verification by TPA for the Data Updating

Third Party Auditor sends ID, signature given by the client and the data block verification request to CSS. Cloud server decrypts the TPA ID by its own secret key; hence TPA ID is encrypted using the CSS's public key. CSS verifies the signature on the TPA whether the TPA is an authorized person or not. If TPA is an authorized auditor then the CSS sends the requested updated data blocks to the TPA.

4.5 CONSIGNMENT ASSESSMENT

Input: TPA Challenge Message

Output: Assessment of a batch of blocks

This extends the scheme to support scalable and efficient public auditing in Cloud Computing. In particular, scheme achieves batch auditing where multiple delegated auditing tasks from the client can be performed simultaneously by the TPA. The challenge verification is done for a collection of blocks by the auditor which is referred as the batch auditing.

5 RELATED WORK

Ee-Chien Chang[3] Computation time and it is not clear how to aggregate multiple responses to reduce communication bits. This scheme can be deployed as a POR system and it also serves as an example of an effective POR system whose extraction is not verifiable. **Jiangtao Li[6]** Complexity of algorithm. . A policy language is introduced that enables negotiators to specify authorization requirements that must be met by an opponent to receive various amounts of information about certified attributes and the credentials that contain it. The language also supports the use of uncertified attributes, allowing them to be required as part of policy satisfaction, and to place their (automatic) disclosure under policy control. **Qian Wang[10]** Large number of data's cannot be updated. While the prior works on ensuring remote data integrity often lacks the support of either public verifiability or dynamic data operations. To provide Merkle Hash Tree (MHT) construction for block tag authentication. Extensive security and performance analysis show that the proposed scheme is highly efficient and provably secure.**Liwen Sun[8]** Store the features once and then one bit vector for each block. . To find an approximate solution efficiently,they adopt the bottom-up clustering framework. they prototyped our blocking techniques on Shark, an open-source data warehouse system. The experiments on TPC-H and a real-world workload show that our blocking technique leads to 2-5x improvement in query response time over traditional range-based blocking techniques.

6 CONCLUSION

Cloud computing is a construct that allows user to access applications that actually reside at a location other than the user's system or other internet connected devices. The widespread adoption of cloud storage services, the public cloud storage model should solve the critical issue of data confidentiality. That is, shared sensitive data must be strongly secured from unauthorized accesses. Cloud users may need to split the large-scale datasets into smaller chunks before uploading to the cloud for privacy-preserving. In this regard, efficiency in processing small updates is always essential in big data applications. And the data integrity verification is done by a trusted third party where the process is called as Third-Party Auditing (TPA).

However in such process a necessary authorization/authentication process is missing between the auditor and cloud service provider. To provide Fine-grained data updates that can fully support authorized auditing and Fine-grained update requests. An additional authorization process is added among the three participating parties of client, Cloud Storage Server (CSS) and a third-party auditor (TPA).

7 FUTURE WORK

The Auditing can be performed for a consignment of files, where the authorized auditor performs the auditing for a set of files at a time.

REFERENCES

[1] Ateniese.G et al (2009), "Proofs of Storage from Homomorphic Identification Protocols" ,in proceedings of the 15th international conference on the theory and application of Cryptology and Information Security(ASIACRYPT '09),pp.319-378.

[2] Boneh.D et al (2004), "Short Signatures from the Weil Pairing", Journal of Cryptology,vol.17,no.4,pp.297-319.

[3] Chang E.C and Xu.J, (2012), "Remote Integrity Check with Dishonest Storage server, "inProc.of ESORICS'08.Berlin, Heidelberg:Springer-Verlag.

[4] Chang Liu et al (2013), "Authorized Public Auditing of Dynamic Big Data storage on Cloud with efficient Fine-grained Updates",in IEEE Transactions on Parallel and Distributed Systems.

[5] He.Yet al (2011),"Preventing Equivalence Attacks in Updated,Anonymized Data",in proceedings of the 27th IEEE International Conference on Data Engineering(ICDE '11),pp.529-540.

[6] Jiangtao Li and Ninghui Li (2007), "Automated Trust Negotiation Using Cryptographic Credentials".In Proceedings of the 9th International Conference on FinancialCryptography and Data Security.

[7] Juels.A and Kaliski B.S, (2011), "Pors: proofs of Retrievability for large files," in Proc. of CCS'07. New York, NY, USA: ACM.

[8] LiTheyn et al (2006),"Fine-grained Partitioning for Aggressive Data Skipping". In R. Guerraoui, editor, *DISC '04*, pages 405–419. Springer, 2004. LNCS vol. 3274.

[9] Naor.M and Rothblum.G.N, (2010), "The complexity of online memory checking," inProc. of FOCS'05.

[10] Shacham.H and Waters.B (2008), " Compact proofs of Retrievability". Cryptology ePrintArchive, Report 2008/073.