Improving Performance of Efficient Data Dynamics in STaaS using FMT

Geetha.K
M.Tech, IVth Semester
Dept of CSE, RGIT Bangalore
geethak382@gmail.com

Narendra Babu C.R
Assistant Professor,
Dept of CSE, RGIT Bangalore
narendrababu.cr@gmail.com

George Fernandez. I
Associate Professor,
Dept of CSE, RGIT Bangalore
george.contact@gmail.com

ABSTRACT
Storage as a Service (STaaS) is the service offered by cloud computing over the Internet where user data is stored and maintained remotely without the burden of local data storage and maintenance. However, the users will no longer have physical possession of the outsourced data brings new challenge towards the integrity of the data. The introduction of TPA eliminates the involvement of the client through the auditing of whether his data stored in the cloud are indeed intact. Moreover, user frequently wants to update the data they store. The support for data dynamics via the most general forms of data operation, such as data insertion, deletion and modification is also very important, since services in Cloud Computing are not just store static files. Experimental analysis shows how to construct an elegant verification scheme which preserves privacy. To achieve efficient data dynamics, the existing storage models is manipulated by using the Fractal tree traversal technique. The Fractal tree construction and traversal greatly improves the performance of data dynamics where user's time taken to insert, update and delete the content of the file is reduced significantly.

Key words*: Cloud storage, Data dynamics, Fractal Merkle tree traversal, Merkle hash tree, Third party auditor.*

1. INTRODUCTION

CLOUD Computing is a promising technology which is changing the way the computing and storage resources will be accessed in the near future. Cloud computing delivers software, platform, and infrastructure that are provided as subscription-based services in a pay-as-you-go model to consumers. These services are: Infrastructure as a Service (IaaS) is providing general on-demand computing resources such as various forms of storage or virtualized servers as metered resources. This can be viewed as a direct evolution of shared hosting with added on-demand resource virtualization where billing is primarily based on usage. Platform as a Service (PaaS) is providing an existent managed higher-level software infrastructure for building particular classes of applications and services. The platform consists of underlying computing resources and are typically billed similar to IaaS products. Here the infrastructure is abstracted away below the platform. Software as a Service (SaaS) is providing specific applications as fully or partially remote services. Sometimes it will be in the form of web-based applications and other times it might consist of standard non-remote applications with Internet-based storage.

Storage as a Service (STaaS) is the service offered by cloud computing over the Internet where user data is stored and maintained remotely. Storing data remotely to the cloud in a flexible on–demand manner brings appealing benefits such as cost reduction, risk transfer, location independent access, rapid resource elasticity, ease of maintenance. It also brings new and challenging security threats towards user's outsourced data. Since data is stored remotely, the user ultimately loses control over the data and consistency of the data is under threat. To fully ensure the data integrity and save the cloud user's computation resources as well as online burden, it is very importance to enable public auditing service for cloud data storage, so that users may delegate to an independent third-party auditor (TPA) to audit the outsourced data whenever needed. Here we support privacy-preserving public auditing in cloud computing, with a focus on efficient dynamic data storage.

The main goal is to support efficient data dynamics for privacy-preserving public auditing using Merkle Signature Scheme [4]. Although hash functions are very inefficient, too many secret leaf values would need to be authenticated for each digital signature. The time or space cost, we found that for medium - size trees the computational cost can be made insufficiently efficient for practical use. This construction roughly speeds up the signing operation inherent in Merkle's algorithm at a cost of requiring more space.

How to modify Merkle's scheduling algorithm to achieve various tradeoffs between storage and computation. The improvement is achieved by means of a careful choice of what nodes to compute, retain, and discard at each stage. By reducing the time or space cost, we found that for medium - size trees the computational cost can be made sufficiently efficient for practical use by manipulating the Fractal tree construction and traversal [5].Existing system is among the first few ones to support privacy-preserving public auditing in cloud computing, with a focus on dynamic data storage. Besides, with the widespread use of cloud computing, more number of auditing tasks from different users may be delegated to TPA. For the TPA being alone it will be difficult to audit these many growing tasks. So there is a natural demand how to enable the TPA to efficiently perform multiple auditing tasks in a batch manner, i.e., simultaneously. Our scheme enables an external auditor to audit user's cloud data without learning the data content. Existing system uses Merkle hash tree and its traversal technique for implementing data dynamics which has performance issues.

2. PROPOSED WORK

We use the algorithm of privacy-preserving public auditing system for data storage in cloud computing. We are using the homomorphic linear authenticator and random masking [1] to guarantee that the TPA would not learn any knowledge about the data content stored on the cloud server during the efficient auditing process. This not only eliminates the burden of cloud user from the tedious and possibly expensive auditing task, but also eliminates the users' fear of their outsourced data leakage. The TPA may concurrently handle multiple audit sessions from different users for their outsourced data files, and privacy-preserving public auditing protocol into a multiuser setting. In this setting the TPA can perform multiple auditing tasks in a batch manner for better efficiency.

We extend our work to achieve efficient data dynamics using the Fractal tree representation and traversal [5], since user wants to constantly modify the data they store. The user should feel as if data is stored locally and they do not tolerate the delay while updating the data. The construction of fractal tree and its traversal greatly improves the time taken to support data dynamics. The main idea of the fractal tree traversal is, to split up the merkle tree in to subtrees and to save and compute these subtrees, instead of single nodes which increase the efficiency of dynamic data.

3. SYSTEM DESIGN

In this experimental analysis the cloud data storage service involving three different entities[1]: the cloud user, who wants to store lot of data files into the cloud; the cloud server, which provide data storage service and is managed by the cloud service provider and has sufficient storage space and resources; the third-party auditor, who has extensive capabilities that cloud users do not have and is allowed to assess the cloud storage service reliability on behalf of the user upon request. Users depend on the CS for cloud data storage and maintenance. They may also frequently interact with the CS to access and update their stored data for various application purposes. As users no longer have their data locally, it is very important for users to ensure that their data are being stored correctly and maintained properly. To save the computation resource as well as the online burden potentially brought by the periodic storage correctness verification. The cloud users may depend on TPA for ensuring the storage integrity of their outsourced data, with a hope of keeping their data secret from TPA. A public auditing scheme consists of four algorithms (KeyGen, SigGen, GenProof, VerifyProof)[7]. KeyGen is a key generation algorithm that is run by the user to setup the scheme. SigGen algorithm is used by the user to generate verification metadata. GenProof algorithm is run by the cloud server to generate a proof of data storage correctness, while VerifyProof algorithm is run by the TPA to audit the proof.

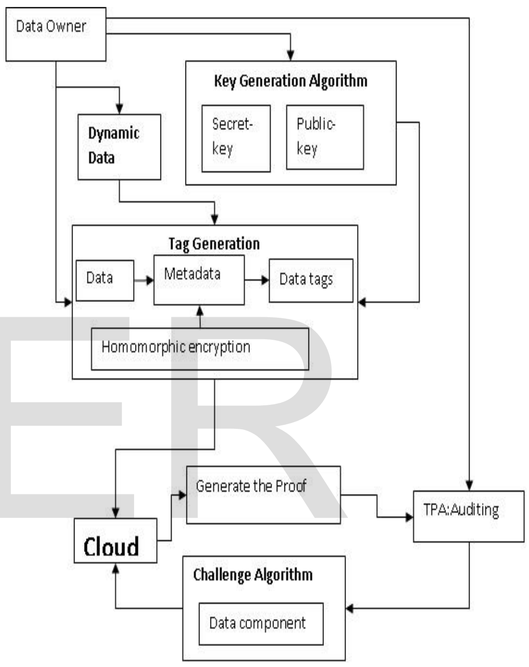

Fig 1: Architecture Diagram

The support for data dynamics via the most general forms of data operation, such as block insertion, deletion and modification is also very significant step, since services in Cloud Computing are not limited to static data only. While prior works on ensuring remote data integrity often lacks the support of either public auditability or dynamic data operations, this paper achieves both. Here we mainly focus on time taken to perform update operations on remotely stored data. We can make user to feel as if data stored locally by providing efficient data dynamics.

In particular, to achieve efficient data dynamics, we are trying to improve the existing proof of storage models by manipulating the Fractal tree construction for block tag authentication. The basic idea of the fractal algorithm is that it uses subtrees of T instead of single nodes (like the authentication nodes in the Merkle's algorithm). Instead of storing all tree nodes, we store a smaller set – those within the stacked subtrees. In the process of the algorithm it will always have a stacked series of L subtrees, called i Exist. These existing subtrees are always precomputed, which means that the values of all of their nodes are already calculated and stored (except of the roots) in the beginning of the algorithm. Their role in the traversal is very important – they simply contain the authentication path of every leaf of the lowest subtree Exist (the idea of the algorithm is that Exist is constructed and after that modified in order to contain every leaf and its authentication path that have to be output). Like Merkle traversal technique the Fractal traversal technique also has setup phase and output phase. The procedure for Fractal tree traversal is as follows.

Procedure: Fractal merkle tree Traversal.

Setup phase

```
1 while i<{1,2,...,L}://where L is the level
        { At each level i
        Calculate all non – root nodes in
        existing subtree .
        Create new empty desired subtree
        except root.}
 2 Calculate and output tree root ie public key.
```

Output phase

```
1 Set leaf =0.
2 while (i<2H-1)//where i is leaf,H is height
  Display the Authentication path for leaf i .
3 For each i for which i Exist is
   no longer needed
• Remove node in i Exist .
• Rename tree i Desire as tree i Exist .
• Create new, empty tree i Desire where
4 Grow Subtrees: while i<{1,2,...,L−1}:
  Grow tree i Desire to calculate its child node
5 Increment leaf and goto step 2
```

The main idea of the fractal traversal is, to split up the merkle tree in subtrees [5] and to save and compute these subtrees, instead of single nodes. Each of this subtree t has the same height h. Each root of one subtree is signed with a signature of the superior subtree. The signature generation phase consists again of two phases. In the output phase, the signature and the authentication path is outputted. In the update phase, the subtrees with the next authentication node Exist_i are set and the sub-trees Desire_i are computed. In order to generate the subtree Desire_i, we use a slightly modified treehash algorithm. Fractal algorithm requires a maximum of 2log(N)/ log(log(N)) hash function evaluations for every output (in its worst case) and maximum storage of $1.5\log^2(N)/\log(\log(N))$ hash values[5]. Whereas Merkle tree traversal technique requires a maximum of $2\log_2(N)$ invocations of the hash function f per round and also a maximum storage of $\log_2{}^2(N)/2$ outputs of f [5] . This tradeoff of time and space greatly improves the efficiency of data dynamics in our experimental analysis.

4. PERFORMANCE EVALUATION

Here we deal with performance evaluation results of the proposed work. The proposed work is implemented using java language. For evaluation we are considering the time taken to insert, delete and modify the content of the remotely stored files with number of files.

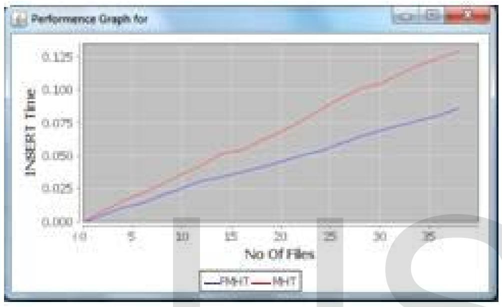

Fig 2: Comparison of insert time with No of files

It has been proved experimentally that time taken to insert data to file is less with Fractal Merkle hash tree traversal as shown in Fig 2.

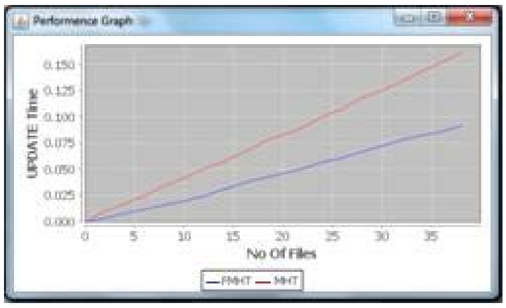

Fig 3: Comparison of update time with No of files

Above graph clearly shows that the update time with Fractal tree traversal is less when compared with merkle hash tree traversal.

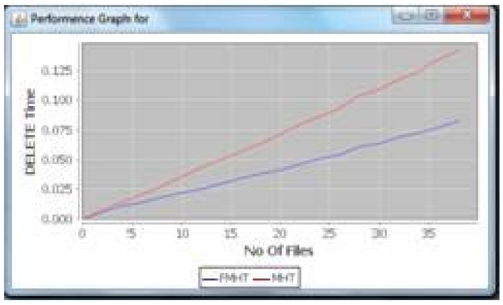

Fig 4: Comparison of Delete time with No of files

The experimental analysis has shown that Fractal tree traversal greatly improves the performance of delete the contents of file when compared with no of files.

4.1. COMPARISION TABLE

The following table shows the overall improvement in performance when compared with existing and proposed system.

	MHT	FMHT	Improvement in proposed system(%)
	Performance in existing system (%)	Performance in proposed system(%)	
Insertion	36	18	18
Updation	41	20	21
Deletion	40	17	23

Table1: Comparison of performance

It is clearly proved that with Fractal merkle tree traversal technique the time taken to insert, update and delete operations is performing better when compared to classic Merkle hash tree traversal technique.

5. CONCLUSION

Our proposed work is concept of privacy-preserving public auditing system for data storage security in cloud computing. We have used the homomorphic linear authenticator and random masking to guarantee that the TPA would not learn any knowledge about the data content stored on the cloud server during the efficient auditing process. This not only eliminates the burden of cloud user from the tedious and possibly expensive auditing task, but also eliminates the users' fear of their outsourced data leakage. Extensive analysis shows that our schemes are highly efficient and secure. We extend our work to achieve efficient data dynamics using the Fractal tree representation. This traversal technique has better space and time complexity. So the fractal tree traversal has the advantage to allow tradeoffs between storage and computation. It reducing the time or space cost, we found that for medium - size trees the computational cost can be made sufficiently efficient for practical use. Experimental analysis has shown that there is approximately 20% improvement in overall performance. This improvement makes the user to feel remotely stored data as stored locally.

REFERENCES

[1]. Cong Wang, Member, IEEE, Sherman S.M. Chow, Qian Wang, Member, IEEE, Kui Ren, Senior Member, IEEE, and Wenjing Lou, Senior Member, IEEE "Privacy-Preserving Public Auditing for Secure Cloud Storage", IEEE TRANSACTIONSON COMPUTERS, VOL. 62, NO. 2, FEBRUARY 2013.

[2]. M. A. Shah, R. Swaminathan, and M. Baker, "Privacy-preserving audit and extraction of digital contents," Cryptology ePrint Archive, Report 2008/186, 2008.

[3]. Juels and B.S. Kaliski Jr., "Pors: Proofs of Retrievability for Large Files," Proc. 14th ACM Conf. Computer and Comm. Security (CCS '07), pp. 584-597, 2007.

[4]. Ralph Merkle, "Secrecy, Authentication and Public Key Systems/ Acertified digital signature", Ph.D. dissertation, Dept. of Electrical Engineering, Stanford University, 1979.

[5]. Markus Jakobsson, Tom Leighton, Silvio Micali and Michael Szydlo,"Fractal Merkle Tree Representation and Traversal", RSA-CT '03.

[6]. H. Shacham and B. Waters, "Compact Proofs of Retrievability," Proc. 14th Int'l Conf. Theory and Application of Cryptology and Information Security: Advances in Cryptology ASIACRYPT '08), pp. 90-107, 2008.

[7]. G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song, "Provable Data Possession at Untrusted Stores," Proc. 14th ACM Conf. Computer and Comm. Security (CCS'07), pp. 598-609, 2007.

[8]. G. Ateniese, R.D. Pietro, L.V. Mancini, and G. Tsudik, "Scalable and Efficient Provable Data Possession," Proc. Fourth Int'l Conf. Security and Privacy in Comm. Networks (SecureComm '08), pp. 1-10, 2008.

[9]. C.Erway, A.Kupcu, C.Papamanthou, and R.Tamassia, "Dynamic Provable Data Passion",Proc.16th ACM conf. computer and comm.. security (2009).

[10]. C. wang, Q.Wang, K.Ren, and W.Lou, "Towards Secure and Dependable Storage Services in Cloud Computing", IEEE Transactions on Service computing,2011.