IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661, p- ISSN: 2278-8727Volume 16, Issue 3, Ver. I (May-Jun. 2014), PP 16-18
www.iosrjournals.org

Mutual Trust to Provide Data Security in Cloud Environment

[1]Srinivas Kolli , [2]Ch.Sravan Kumar , [3]Dr P.Chenna Reddy,
[1]*Asst Professor, Vardhaman College Of Engineering(Cse Dept),*
[2]*Asst Professor,Vardhaman College Of Engineering(Cse Dept),*
[3]*Professor, Jntua College Of Engineering, Cse Dept,*

Abstract: *We introduce a model for provable data possession (PDP) that allows a client that has stored data at an untrusted server to verify that the server possesses the original data without retrieving it. The model generates probabilistic proofs of possession by sampling random sets of blocks from the server, which drastically reduces I/O costs. The client maintains a constant amount of metadata to verify the proof. The challenge/response protocol transmits a small, constant amount of data, which minimizes network communication. Thus, the PDP model for remote data checking supports large data sets in widely-distributed storage systems. It is also superior in performance by minimizing the use of expensive public-key cryptography in metadata management. We present the architecture and implementation of various SHAROES com- ponents and our experiments demonstrate performance superior to other proposals by over 40% on a number of benchmarks.*

Index Terms: *Cloud Computing, Trust ability.*

I. Introduction

96% of common people used to think that cloud is the best place to store and retrieve the values virtually, and 62% of business entrepreneurs used to think that cloud is the best place to store the content but the case about security from hackers.To make use of these resources we need search mechanisms that distill the information relevant to each user. Nor- mally, such mechanisms require the user to provide a server with a query such as a textual keyword that the server will compare against the documents in some large data set. This model becomes problematic for applications in which the user would like to hide the search criteria. A user might want to protect the pri- vacy of his search queries for a variety of reasons, in- cluding protection of commercial interests and personal privacy. Such privacy issues were brought into the spotlight in 2005 when the U.S. Department of Jus- tice subpoenaed records of search terms from popular web search engines.

In the current era of digital world, different organizations produce a large amount of sensitive data including personal information, electronic health records, and financial data. The amount of digital data increases at a staggering rate; doubling almost every year and a half [1].

This data needs to be widely distributed and stored for a long time due to operational purposes and regulatory compliance. The local management of such huge amount of data is problematic and costly. While there is an observable drop in the cost of storage hardware, the management of storage has become more complex and represents approximately 75% of the total ownership cost [1]. SaaS offered by CSPs is an emerging solution to mitigate the burden of large local data storage and reduce the maintenance cost via the concept of outsourcing data storage.

Commonly, traditional access control techniques assume the existence of the data owner and the storage servers in the same trust domain. This assumption, however, no longer holds when the data is outsourced to a remote CSP, which takes the full charge of the outsourced data management, and resides outside the trust domain of the data owner. A feasible solution can be presented to enable the owner to enforce access control of the data stored on a remote untrusted CSP. Through this solution, the data is encrypted under a certain key, which is shared only with the authorized users.

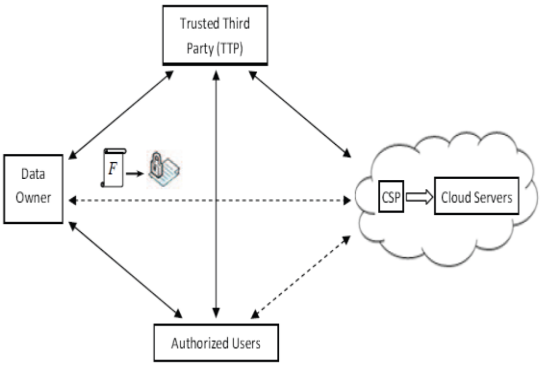
II. Related Works

Existing research close to our work can be found in the areas of integrity verification of outsourced data, cryptographic file systems in distributed networks, and access control of outsourced data. Different variations of PDP protocols have been presented for static or warehoused data; for example,see [3]–[10].

Some other PDP schemes consider the case of dynamic data that are usually more prevailing in practical applications. Examples of PDP schemes that deal with dynamic data are [11]–[15]. While the schemes [3]–[15] are for a single copy of a data file, PDP schemes have been presented for multiple copies of static data, e.g., [16]–[18]. Reference [19] addresses a PDP construction for multiple copies of dynamic data.

Proof of retrievability (POR) is a complementary approach to PDP, and is stronger than PDP in the sense that the entire data file can be reconstructed from portions of the data that are reliably stored on the servers. This is due to encoding of the data file, for example using erasure codes, before outsourcing to remote servers. References [20]–[25] are examples of POR schemes that can be found in the literature. Kallahalla et al. [26] designed a cryptography-based file system called Plutus for secure sharing of data on untrusted servers. Some authorized users of the data have the privilege to read and write, while others can only read the data.

In Plutus, a file-group represents a set of files with similar attributes, and each file-group is associated with a symmetric key called file-lockbox key. A data file is fragmented into blocks, where each block is encrypted with a unique symmetric key called a file-block key. The file-block key is further encrypted with the file-lockbox key of the file-group to which the data file belongs. If the data owner wants to share a file-group with a set of users, the file-lockbox key is just distributed to them. Plutus supports two operations on the file blocks:



III. Proposed System

We propose a very efficient cheating detection mechanism to effectively verify in one batch of all the computation results by the cloud server from previous algorithm iterations with high probability. We formulate the problem in the computation outsourcing model for securely solving large-scale systems of LE via iterative methods, and provide the secure mechanism design which fulfills input/output privacy, cheating resilience, and efficiency. Our mechanism brings computational savings as it only incurs O(n) local computation burden for the customer within each algorithm iteration and demands no unrealistic IO cost, while solving large scale LE locally usually demands more than O(n2) computation cost in terms of both time and memory requirements. We explore the algebraic property of matrix-vector multiplication to design a batch result verification mechanism, which allows customers to verify all answers computed by cloud from previous iterations in one batch, and further ensures both the efficiency advantage and the robustness of the design. The experiment on Amazon EC2 shows our mechanism can help customers achieve up to 2.22× savings when the sizes of the LE problems are relatively small (n ≤ 50, 000).

- ❖ The problem of securely outsourcing large-scale systems of LE via iterative methods, and provide mechanism designs fulfilling input/output privacy, cheating resilience, and efficiency.
- ❖ Our mechanism brings computational savings
- ❖ We explore the algebraic property of matrix-vector operations to design a batch verification mechanism, which allows customers to verify all results of previous iterations from cloud in one batch. It ensures both the efficiency advantage and robustness of the design.

IV. Implemention:

Implementation is the stage of the project when the theoretical design is turned out into a working system. Thus it can be considered to be the most critical stage in achieving a successful new system and in giving the user, confidence that the new system will work and be effective.

The implementation stage involves careful planning, investigation of the existing system and it's constraints on implementation, designing of methods to achieve changeover and evaluation of changeover methods.

- ❖ Cloud Computing
- ❖ Homomorphic Encryption
- ❖ General Techniques

```
try {
        Statement st=new DBConn().getConn().createStatement();
        int k=st.executeUpdate("update uquery set token= '"+buf+"' where date='"+date+"'");
```

```
          int k1=st.executeUpdate("update title set token='"+buf+"'  where "+bpVal+">min and "+bpVal+"<max 
and title='"+spt[0]+"' and mm='"+mm+"'and pa='"+pa+"' and nd='"+nd+"'");
         //  PreparedStatement psmt=new DBConn().getConn().prepareStatement("insert into title 
(token)values(?)");
          // psmt.setString(1, buf);
         //  psmt.executeUpdate();

          RequestDispatcher rd=request.getRequestDispatcher("stasuc.jsp");
          rd.forward(request, response);

        }catch(Exception e){
        e.printStackTrace();
        RequestDispatcher rd=request.getRequestDispatcher("staqn.jsp");

        rd.forward(request, response);
        }
        finally {
          out.close();
```

V. Conclusions

We Outsourcing data to remote servers has become a growing trend for many organizations to alleviate the burden of local data storage and maintenance. In this work we have studied different aspects of outsourcing data storage: block-level data dynamic, newness, mutual trust, and access control. We have proposed a cloud-based storage scheme which supports outsourcing of dynamic data, wherethe owner is capable of not only archiving and accessing the data stored by the CSP, but also updating and scaling this data on the remote servers. The proposed scheme enables the authorized users to ensure that they are receiving the most recent version of the outsourced data. Moreover, in case of dispute regarding data integrity/newness, a TTP is able to determine the dishonest party. The data owner enforces access

References

[1]. A. Singh and L. Liu, "Sharoes: A data sharing platform for outsourced enterprise storage environments," in Proceedings of the 24th International Conference on Data Engineering, ICDE. IEEE, 2008, pp. 993–1002.

[2]. 104th United States Congress, "Health Insurance Portability and Accountability Act of 1996 (HIPAA)," Online at http://aspe. hhs.gov/admnsimp/pl104191.htm, 1996.

[3]. R. ıdane, "Remote integrity checking," in 6th Working Conference on Integrity and Internal Control in Information Systems (IICIS), S. J. L. Strous, Ed., 2003, pp. 1–11..

[4]. B. e, J. Domingo-Ferrer, A. Martinez-Balleste, Y. Deswarte, and J.-J. Quisquater, "Efficient remote data possession checking in critical information infrastructures," IEEE Trans. on Knowl. and Data Eng., vol. 20, no. 8, 2008..

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