Efficient Approach for Data Retrievability on Cloud

G. S. Rizwana Banu[1], G. V. Manojkumar[2], M. Ranjith Kumar[3], J. Samira[4], M. Tamilselvi[5]
[1, 2, 3, 4, 5] Department of CSE
[1, 2, 3, 4, 5] K.S.R college of Engineering, Tiruchengode

Abstract- *Cloud storage is a model of data storage in which the digital data is stored in logical pools. It allows users to store their data in a remote server to get rid of expensive local storage, management costs and to access data of interest anytime anywhere. We propose a dynamic proof of retrievability scheme for supporting efficient recovery from data corruptions. We split up the data into small data blocks and encode that data block using network coding. To eliminate the communication overhead for small data corruptions within a server, each data block is further encoded. Based on the encoded data blocks, we utilize tree structure to enforce the data sequence for dynamic operations, preventing the cloud service provider from manipulating data block. We also analyze with our encrypted Base64 method for the effectiveness of the proposed construction in defending against pollution attacks during data retrievability*

Keywords- Cloud Storage, Data Retrievability, Base64 method.

I. INTRODUCTION

A cloud refers to a distinct IT environment that is designed for the purpose of remote provisioning scalable and measured IT resources. The term originated as a metaphor for the Internet which is, in essence, a network of networks providing remote access to a set of decentralized IT resources. Prior to cloud computing becoming its own formalized IT industry segment, the symbol of a cloud was commonly used to represent the Internet in a variety of specifications and mainstream documentation of Web-based architectures. This same symbol is now used to specifically represent the boundary of a cloud environment. The cloud symbol from the Internet. As a specific environment used to remotely provision IT resources, a cloud has a finite boundary. There are to achieve within server and cross server data redundancy, tolerating data corruption. By combing range based 2-3 tree and improved version of aggregately signature based broadcast encryption, our construction can support efficient data dynamics while defending against data replay attack. many individual clouds that are accessible via the Internet.

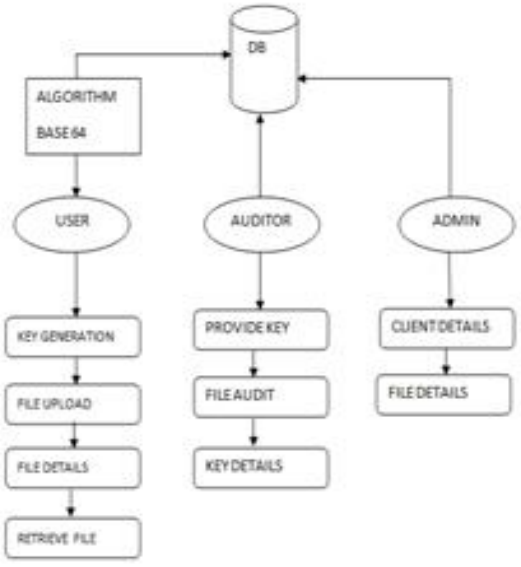
II. EXISTING SYSTEM

In existing system, while uploading, the entire data were uploaded as a single block, so we couldn't find the particular data loss. Do not support efficient data dynamics and/or suffer from security vulnerabilities when involving dynamic data operations. Here they haven't used any network code or erasure codes; hence they faced many difficulties while finding the redundancies. No file audit report and file audit delegation. Data corruption caused by server hacks or Byzantine failures.

III. PROPOSED SYSTEM

We propose an enhanced dynamic proof of retrievability scheme supporting public audit ability and communication-efficient recovery from data corruptions. To this end, we split up the data into small data blocks and encode each data block individually using network coding. Network coding and erasure codes are adopted to encode data blocks

IV. ARCHITECTURE DIAGRAM



V. HARDWARE DESCRIPTION

Developing Kit			
	Processor	RAM	Disk Space
Net Beans 8.0	Computer with a 2.6GHz processor or higher	512MB Minimum	Minimum 20GB
Database			
MySql 5.0	Intel Pentium processor at 2.6GHz or faster	Minimum 512MB physical memory; 1GB recommended	Minimum 20GB
HeidiSQL 8.3	Intel Pentium processor at 2.6GHz or faster	Minimum 512MB physical memory; 1GB recommended	Minimum 20GB

VI. SOFTWARE DESCRIPTION

6.1 FRONT END

Java Platform

Java is an object–oriented programming language developed by sun Microsystems, a company best known for its high-end UNIX/LINUX workstations. Modeled after C++, the java language was designed to be small, simple and portable across platforms and operating systems, both at source and the binary level, which means that java programs can run on any machine that has java virtual machine installed. They are two types of java programs. They are java applets and java applications.

Java is a platform independent at both the source level and the binary level; platform independence means that a program can run on any computer system. Java programs can run or any system for which a Java virtual machine has been installed. Unlike other programming languages when Java programs are compiled byte codes are generated which is a special set of machine instructions that are not specific to any one-processor or computer system.

About Java:

Initially the language was called as “oak” but it was renamed as “Java” in 1995.The primary motivation of this language was the need for a platform-independent (i.e. Architecture neutral) language that could be used to create software to be embedded in various consumer electronic devices.

- Java is a programmer’s language
- Java is cohesive and consistent
- Except for those constraints imposed by the Internet environment. Java gives the programmer, full control

Object oriented

Java was not designed to be source-code compatible with any other language. This allowed the Java team the freedom to design with a blank state. One outcome of this was a clean, usable, pragmatic approach to objects. The object model in Java is simple and easy to extend, while simple types, such as integers, are kept as high-performance non-objects.

Robust

The multi-platform environment of the web places extraordinary demands on a program, because the program must execute reliably in a variety of systems. The ability to create robust programs. Was given a high priority in the design of Java. Java is strictly typed language; it checks your code at compile time and runtime.

6.2 BACK END

JDBC

Any relational database. One can write a single program using the JDBC API, and the JDBC is a Java Api for executing SQL, Statements(As a point of interest JDBC are trademarks names and is not an acronym; nevertheless, Jdbc is often thought of as standing for Java Database Connectivity. It consists of a set of classes and interfaces written in the Java Programming language. JDBC provides a standard API for tool/database developers and makes it possible to write database applications using a pure Java API

What Does JDBC Do?

Simply put, JDBC makes it possible to do three things

- Establish a connection with a database
- Send SQL statements
- Process the results
- JDBC Driver Types
- The JDBC drivers that we are aware of this time fit into one of four categories

- JDBC-ODBC Bridge plus ODBC driver
- Native-API party-Java driver
- JDBC-Net pure Java driver
- Native-protocol pure Java driver

An individual database system is accessed via a specific JDBC driver that implements the java.sql.Driver interface. Drivers exist for nearly all-popular RDBMS systems, through few are available for free. Sun bundles a free JDBC-ODBC bridge driver with the JDK to allow access to a standard ODBC, data sources, such as a Microsoft Access database, Sun advises against using the bridge driver for anything other than development and very limited development.

There are four driver categories

Type 01-JDBC-ODBC Bridge Driver

Type 01 drivers use a bridge technology to connect a Java client to an ODBC database service. Sun's JDBC-ODBC bridge is the most common type 01 driver. These drivers implemented using native code.

Type 02-Native-API party-Java Driver

Type 02 drivers wrap a thin layer of Java around a database-specific native code libraries for Oracle databases, the native code libraries.

Importance of Java to the Internet

Java has had a profound effect on the Internet. This is because; Java expands the Universe of objects that can move about freely in Cyberspace. In a network, two categories of objects are transmitted between the server and the personal computer. They are passive information and Dynamic active programs. In the areas of Security and probability. But Java addresses these concerns and by doing so, have opened the door to an exciting new form of program called the Applet.

Applications and applets. An application is a program that runs on our Computer under the operating system on that computer. It is more or less like one creating, using C or C++.Java's ability to create Applets makes it important. An Applet I son application, designed to be transmitted over the Internet and executed by a Java-compatible web browser. An applet I actually a tiny Java program, dynamically downloaded across the network, just like an image. But the difference is, it is an intelligent program, not just a media file. It can be reacted to the user input and dynamically change.

Java Architecture

Java architecture provides a portable, robust, high performing environment for development. Java provides portability by compiling the byte codes for the Java Virtual Machine, which is then interpreted on each platform by the run-time environment. Java is a dynamic system, able to load code when needed for a machine in the same room or across the planet.

When you compile the code, the Java compiler creates machine code (called byte code) for a hypothetical machine called a Java Virtual Machine (JVM). The JVM is supposed to be executed the byte code. The JVM is created for the overcoming the issue of probability. The code is written and compiled for one machine and interpreted on all machines. This machine is called a Java Virtual Machine.

Simple:

Java was designed to be easy for the Professional programmer to learn and to use effectively. If you are an experienced C++ Programmer. Learning Java will oriented features of C++. Most of the confusing concepts from C++ are either left out of Java or implemented in a cleaner, more approachable manner. In Java there are a small number of clearly defined ways to accomplish a given task.

Type 03-Net-Protocol All-Java Driver

Type 03 drivers communicate via a generic network protocol to a piece of custom middleware. The middleware component might use any type of driver to provide the actual database access. These drivers are all Java, which makes them useful for applet deployment and safe for servlet deployment

Type-04-native-protocol All-Java Driver

Type o4 drivers are the most direct of the lot. Written entirely in Java, Type 04 drivers understand database-specific networking. Protocols and can access the database directly without any additional software.

VII. RESULT & CONCLUSION

We have introduced a dynamic proof of retrievability scheme for cloud storage systems. The erasure codes are adopted to encode the data blocks to achieve in server and supporting the efficient data recovery. By using the improved base function for encryption, our construction can support efficient data dynamic while defending against data reply attack and pollution attack.

We improve security by sending the mail states the user login time and device IP. In this work, we are investing in diversity in a given cloud storage system, where for every abstract service in the architecture, there exist numbers of concrete cloud services.

REFERENCES

[1] Cash A. Kupcu D and Wichs D(2013), "Dynamic Proofs of Retrievability via Oblivious",proc. 32nd Annual Int'l Conf. Theory and Application of Cryptographic Techniques (EUROCRYPT'13),pp.279-295.

[2] Chen B. and Curtmola R. (2013), "Towards Self-Repairing Replication Based Storage Systems Using Untrusted Clouds", Proc. third ACM Conf. Data and Application Security and Privacy (CODASPY'13), pp. 377-388.

[3] Mell P. and Grance P.(2013),"Draft Nist Working Definition of Cloud Computing", National Institute of Standards and Technology, Tech. Rep. http://csrc.nist.gov/groups/SNS/cloudcomputing/index.html.

[4] Y. Hu, H.C.H. Chen, P.P.C. Lee, and Y. Tang, "NCCloud: Applying Network Coding for the Storage Repair in a Cloud-of-Clouds," Proc. 10th USENIX Conf. File and Storage Technologies (FAST'12), pp. 265-273, 2012.

[5] Chen H. C. H and Lee P. C. P.(2012), "Enabling Data Integrity Protection in Regenerating-Coding-Based Cloud Storage", Proc.31st Int'lsymp. Realible Distributed System (SRDS'12),pp.51-60.

[6] Wang S. Chen D. Wang Z. and Chang S.(2012),"Public Auditing for Ensuring Cloud Data Storage Security with Zero Knowledge Privacy," Report 2012/365, Cryptology ePrint Archive.

[7] Stefanov E. Dijik M. K. Oprea A. and Jules A. (2011), "Iris: A Scalable Cloud File System with Efficient Integrity Checks", Cryptology eprint Archive.

[8] Zheng Q. and Xu.S (2011), "Fair and Dynamic Proofs of Retrievability" ,Proc. First ACM Conf. Data and Application Security and Privacy (CODASPY'11), pp. 237-248.

[9] Armbrust M. Fox A.Griffith R. Joseph A.D.Katz R.H. Konwinski A.H. Lee G.Patterson D.A. Rabkin A. Stoica I. and Zaharia M.(2010), "Above the Clouds: A Berkeley View of Cloud Computing", Commun. ACM, vol. 53, no. 4, pp. 50-58.

[10] Chen B. Curtmola R. Ateniese G. and Burns R. (2010), "Remote Data Checking for Network Coding-Based Distributed Storage Systems," Proc. 2010 ACM Workshop on Cloud Computing Security Workshop (CCSW'10), pp. 31-42.

[11] Wu Q. Mu Y. Susilo W. Qin B. and Ferrer J.D.(2009), "Asymmetric Group Key Agreement," Proc. 28th Annual Int'l Conf. Theory and Applications of Cryptography Techniques (EUROCRYPT'09), pp. 153170.

[12] H.P. Anvin, "The Mathematics of RAID-6," http://kernel.org/pub/linux/kernel/people/hpa/raid6.pdf, 2007.