A SECURE MULTI OWNER DATA SHARING SCHEME

[1]Prof.A.R.Pimpalapure, [2]Puja K. Kale, [3]Nishigandha Deshmukh,[4]Chetna Tayde
[1,2,3,4]Computer Science and Engineering,
Anuradha Engineering Collage,
Sant Gadgebaba Amravati, Chikhli, India

Abstract: The significant points of this technique a safe multi-proprietor information sharing plan. It infers that any client in the gathering can safely impart information to others by the un put stock in cloud. This plan can bolster dynamic gatherings. Effectively, particularly, new allowed clients can specifically decode information records transferred before their investment without reaching with information proprietors. Client renouncement can be effectively accomplished through a novel denial list without refreshing the mystery Keys of the rest of the clients. The size and calculation overhead of encryption are steady and Independent with the quantity of disavowed users.We display a safe and security saving access control to clients, which ensure any part in a gathering to secretly use the cloud asset. In addition, the genuine personalities of information proprietors can be uncovered by the gathering chief when question happen.

We give thorough security examination, and perform broad reproductions to exhibit the effectiveness of our plan regarding capacity and calculation overhead. Distributed computing gives a temperate and proficient answer for sharing gathering asset among cloud clients. Shockingly, sharing information in a multi-proprietor way while protecting information and personality security from an untrusted cloud is as yet a testing issue, because of the continuous difference in the enrollment.

OBJECTIVE:

This paper presents a Secure Multi owner data sharing scheme, Named Mona, For dynamic groups in the cloud. By leveraging group Signature and dynamic broadcast Encryption techniques, any cloud user can anonymously share data with others.

KEYWORDS: Cloud Computing, java.

I. INTRODUCTION

The major aims of this method a secure multi-owner data sharing scheme. It implies that any user in the group can securely share data with others by the un trusted cloud. This scheme is able to support dynamic groups. Efficiently, specifically, new granted users can directly decrypt data files uploaded before their participation without contacting with data owners. User revocation can be easily achieved through a novel revocation list without updating the secret Keys of the remaining users. The size and computation overhead of encryption are constant and Independent with the number of revoked users. We present a secure and privacy-preserving access control to users, which guarantee any member in a group to anonymously utilize the cloud resource. Moreover, the real identities of data owners can be revealed by the group manager when disputes occur.

We provide rigorous security analysis, and perform extensive simulations to demonstrate the efficiency of our scheme in terms of storage and computation overhead. Cloud computing provides an economical and efficient solution for sharing group resource among cloud users. Unfortunately, sharing data in a multi-owner manner while preserving data and identity privacy from an untrusted cloud is still a challenging issue,due to the frequent change of the membership.

Cloud Computing:

Cloud computing is the use of computing resources (hardware and software) that are delivered as a service over a network (typically the Internet). The name comes from the use of a cloud-shaped symbol as an abstraction for the complex infrastructure it contains in system diagrams. Cloud computing entrusts remote services with a user's data, software and computation.

There are many types of public cloud computing:

- Infrastructure as a service (IaaS)
- Platform as a service (PaaS)
- Software as a service (SaaS)
- Storage as a service (STaaS)
- Security as a service (SECaaS)

- Data as a service (DaaS)
- Test environment as a service (TEaaS)
- Desktop as a service (DaaS)
- API as a service (APIaaS)

The business model, IT as a service (ITaaS), is used by in-house, enterprise IT organizations that offer any or all of the above services.

Using software as a service, users also rent application software and databases. The cloud providers manage the infrastructure and platforms on which the applications run. End users access cloud-based applications through a web browser or a light-weight desktop or mobile app while the business software and user's data are stored on servers at a remote location. Proponents claim that cloud computing allows enterprises to get their applications up and running faster, with improved manageability and less maintenance, and enables IT to more rapidly adjust resources to meet fluctuating and unpredictable business demand.

Introduction to JAVA

The JAVA language was created by James Gosling in June 1991 for use in a set top box project.The language was initially called Oak, after an oak tree that stood outside Gosling's office - and also went by the name Green - and ended up later being renamed to Java, from a list of random words.Gosling's goals were to implement a virtual machine and a language that had a familiar C/C++ style of notation.[6] The first public implementation was Java 1.0 in 1995. It promised "Write Once, Run Anywhere" (WORA), providing no-cost runtimes on popular platforms. It was fairly secure and its security was configurable, allowing network and file access to be restricted.

Major web browsers soon incorporated the ability to run secure Java applets within web pages. Java quickly became popular. With the advent of Java 2, new versions had multiple configurations built for different types of platforms. For example, J2EE was for enterprise applications and the greatly stripped down version J2ME was for mobile applications. J2SE was the designation for the Standard Edition. In 2006, for marketing purposes, new J2 versions were renamed Java EE, Java ME, and Java SE, respectively.

In 1997, Sun Microsystems approached the ISO/IEC JTC1 standards body and later the Ecma International to formalize Java, but it soon withdrew from the process. Java remains a de facto standard that is controlled through the Java Community Process. At one time, Sun made most of its Java implementations available without charge although they were proprietary software. Sun's revenue from Java was generated by the selling of licenses for specialized products such as the Java Enterprise System. Sun distinguishes between its Software Development Kit (SDK) and Runtime Environment (JRE) which is a subset of the SDK, the primary distinction being that in the JRE, the compiler, utility programs, and many necessary header files are not present.

On 13 November2006, Sun released much of Java as free software under the terms of the GNU General Public License (GPL). On 8 May2007 Sun finished the process, making all of Java's core code open source, aside from a small portion of code to which Sun did not hold the copyright.

Primary goals:

There were five primary goals in the creation of the Java language:[citation needed]

1. It should use the object-oriented programming methodology.
2. It should allow the same program to be executed on multiple operating systems.
3. It should contain built-in support for using computer networks.
4. It should be designed to execute code from remote sources securely.
5. It should be easy to use by selecting what were considered the good parts of other object-oriented languages.

II.LITRETURE SURVEY:

1) Achieving Secure, Scalable, and Fine-grained Data Access Control in Cloud Computing

Shucheng Yu_, Cong Wang†, Kui Ren†, and Wenjing Lou__Dept. of ECE, Worcester Polytechnic Institute, Email: {yscheng, wjlou}@ece.wpi.edu

This paper addresses this challenging open issue by, on one hand, defining and enforcing access policies based on data attributes, and, on the other hand, allowing the data owner to delegate most of the computation tasks involved in finegrained data access control to untrusted cloud servers without disclosing the underlying data contents. We achieve this goal by exploiting and

uniquely combining techniques of attribute-based encryption (ABE), proxy re-encryption, and lazy re-encryption. Our proposed scheme also has salient properties of user access privilege confidentiality and user secret key accountability.

2)Plutus: Scalable secure file sharing on untrusted storage
MaheshSan Francisco, CA, USAMarch 31–April 2, 2003

This paper has introduced novel uses of cryptographic primitives applied to the problem of secure storage in the presence of untrusted servers and a desire for ownermanaged key distribution. Eliminating almost all requirements for server trust (we still require servers not to destroy data – although we can detect if they do) and keeping key distribution (and therefore access control) in the hands of individual data owners provides a basis for a secure storage system that can pr otect and share data at very large scales and across trust boundaries.

3) SiRiUS: Securing Remote Untrusted Storage
Eu-Jin Goh_, Hovav Shacham†, Nagendra Modadugu, Dan Boneh‡Stanford University

This paper presents SiRiUS, a secure file system designed to be layered over insecure network and P2P file systems such as NFS, CIFS, OceanStore, and Yahoo! Briefcase. SiRiUS assumes the network storage is untrusted and provides its own read-write cryptographic access control for file level sharing. Key management and revocation is simple with minimal out-of-band communication. File system freshness guarantees are supported by SiRiUS using hash tree constructions. SiRiUS contains a novel method of performing file random access in a cryptographic file system without the use of a block server.

4) **Secure Provenance: The Essential of Bread and Butter of Data Forensics in Cloud Computing**
Rongxing Lu†, Xiaodong Lin‡, Xiaohui Liang†, and Xuemin (Sherman) Shen†

In this paper proposed scheme is characterized by providing the information confidentiality on sensitive documents stored in cloud, anonymous authentication on user access,
and provenance tracking on disputed documents. With the provable security techniques, we formally demonstrate the proposed scheme is secure in the standard model.

5) Ciphertext-Policy Attribute-Based Encryption: An Expressive, E_cient, and Provably Secure Realization
Brent Waters _University of Texas at Austin

This Paper present a new methodology for realizing Ciphertext-Policy Attribute Encryption (CP- ABE) under concrete and non interactive cryptographic assumptions in the standard model. Our solutions allow any encryptor to specify access control in terms of any access formula over the attributes in the system. In our most e_cient system, ciphertext size, encryption, and decryption time scales linearly with the complexity of the access formula. The only previous work to achieve these parameters was limited to a proof in the generic group model.

6) Attribute-Based Encryption for Fine-Grained Access Control of Encrypted Data
Vipul Goyal¤ Omkant Pandey*y* Amit Sahaiz Brent Waters *x*

This Paper presents more sensitive data is shared and stored by third-party sites on the Internet, there will be a need to encrypt data stored at these sites. One drawback of encrypting data, is that it can be selectively shared only at a coarse-grained level (i.e., giving another party your private key). We develop a new cryptosystem for ¯ne-grained sharing of encrypted data that we call Key-Policy Attribute-Based Encryption (KP-ABE). In our cryptosystem, ciphertexts are labeled with sets of attributes and private keys are associated with access structures that control which ciphertexts a user is able to decrypt. We demonstrate the applicability of our construction to sharing of audit-log information and broadcast encryption.

7) Revocation and Tracing Schemes for Stateless Receivers ?
Dalit Naor1, Moni Naor2??, and Je® Lotspiech1

This paper provide a general traitor tracing mechanism that can be integrated with any Subset-Cover revocation scheme that satis¯es a \bifurcation property". This mechanism does not need an a priori bound on the number of traitors and does not expand the message length by much compared to the revocation of the same set of traitors.

EXISTING SYSTEM:

Several security schemes for data sharing on untrusted servers have been proposed. In these approaches, data owners store the encrypted data files in untrusted storage and distribute the corresponding decryption keys only to authorized users. Thus, unauthorized users as well as storage servers cannot learn the content of the data files because they have no knowledge of the decryption keys However, the complexities of user participation and revocation in these schemes are linearly increasing with the number of data owners and the number of revoked users, respectively. By setting a group with a single attribute, Lu et al. proposed a secure provenance scheme based on the cipher text-policy attribute-based encryption technique, which allows any member in a group to share data with others. However, the issue of user revocation is not addressed in their scheme. presented a scalable and fine-grained data access control scheme in cloud computing based on the key policy attribute-based encryption (KP-ABE) technique.

Unfortunately, the single owner manner hinders the adoption of their scheme into the case where any user is granted to store and share data.

III. ARCHITECTURE

3.1 PROPOSED SYSTEM:

This paper, we propose a secure multi owner data sharing scheme, named Mona, for dynamic groups in the cloud. By leveraging group signature and dynamic broadcast encryption techniques, any cloud user can anonymously share data with others. Meanwhile, th e storage overhead and encryption computation cost of our scheme are independent with the number of revoked users. In addition, we analyze the security of our scheme with rigorous proofs, and demonstrate the efficiency of our scheme in experiments.

SYSTEM ARCHITECTURE:

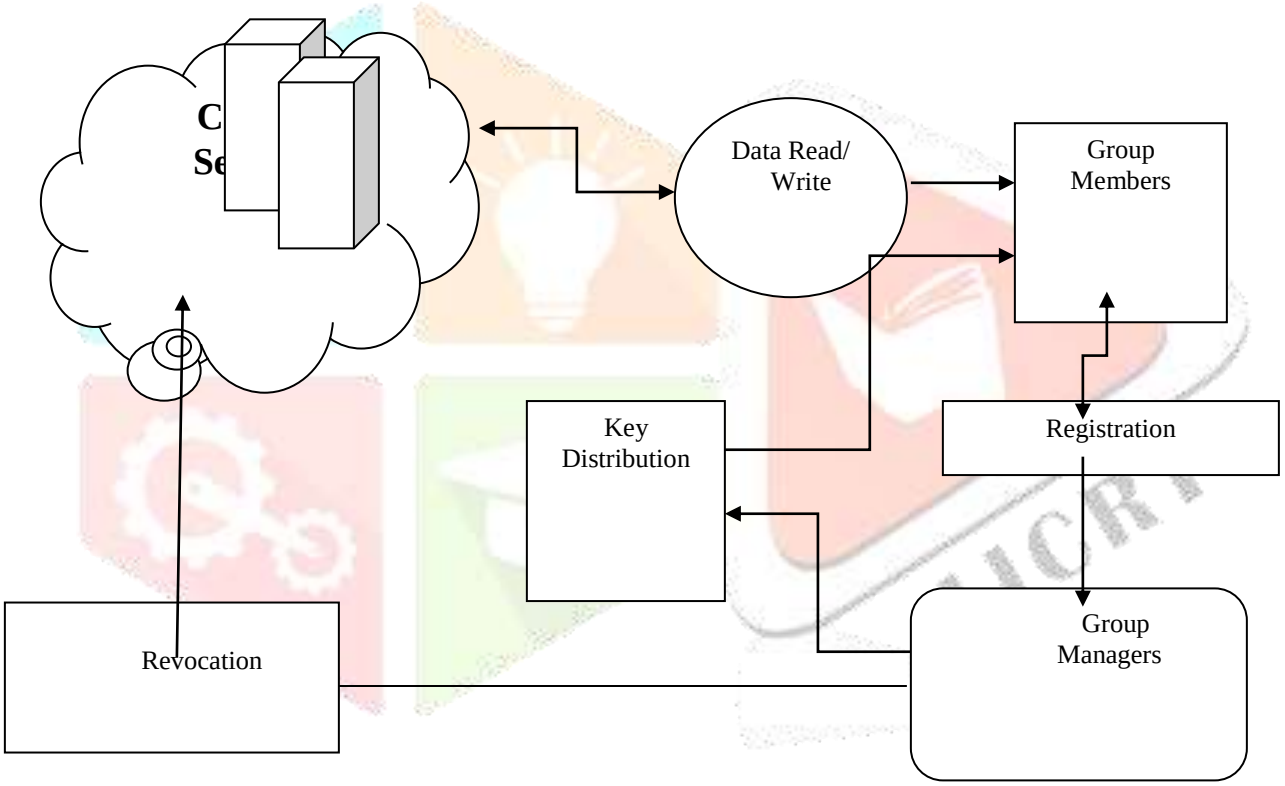


Fig. Block diagram of proposed system

3.2 HARDWARE SPECIFICATION

- Main Processor : 2GHz
- Ram : 512 MB (min)
- Hard Disk : 80 GB

3.3 SOFTWARE SPECIFICATION

- Language : Java

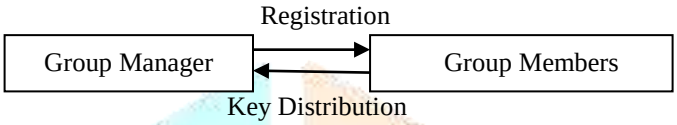
- Web Server : Tomcat 6
- Operating System : Windows 7 32 Bit
- CloudSim

3.4 MODULES:

1. User Registration
2. User Revocation
3. File Uploading and Deletion
4. File Access and Traceability

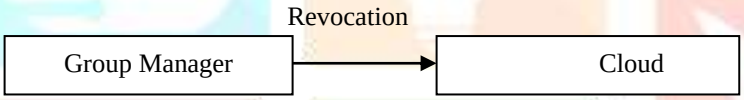
1. User Registration:

For the registration of user with identity ID the group manager randomly selects a number. Then the group manager adds into the group user list which will be used in the traceability phase. After the registration, user obtains a private key which will be used for group signature generation and file decryption.



2. User Revocation:

User revocation is performed by the group manager via a public available. Revocation list, based on which group members can encrypt their data files and ensure the confidentiality against the revoked users. Group manger update the revocation list each day even no user has being revoked in the day. In other words, the others can verify the freshness of the revocation list from the contained current date.



3. File Generation and Deletions:

To store and share a data file in the cloud, a group member performs to getting the revocation list from the cloud. In this step, the member sends the group identity IDgroup as a request to the cloud. Verifying the validity of the received revocation list. File stored in the cloud can be deleted by either the group manager or the data owner.

4. File Access and Traceability:

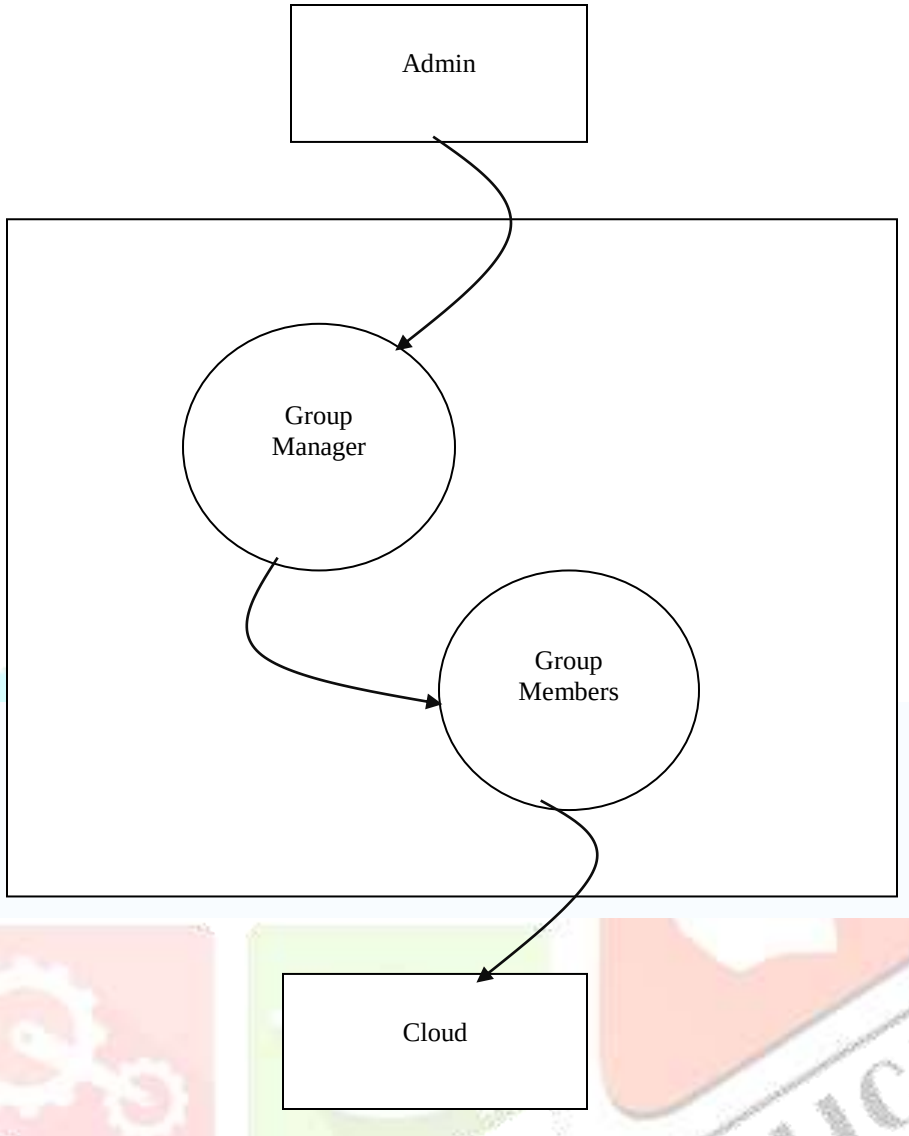
To access the cloud, a user needs to compute a group signature for his/her authentication. The employed group signature scheme can be regarded as a variant of the short group signature which inherits the inherent unforgeability property, anonymous authentication, and tracking capability. When a data dispute occurs, the tracing operation is performed by the group manager to identify the real identity of the data owner.

3.5 DATAFLOW DIAGRAM

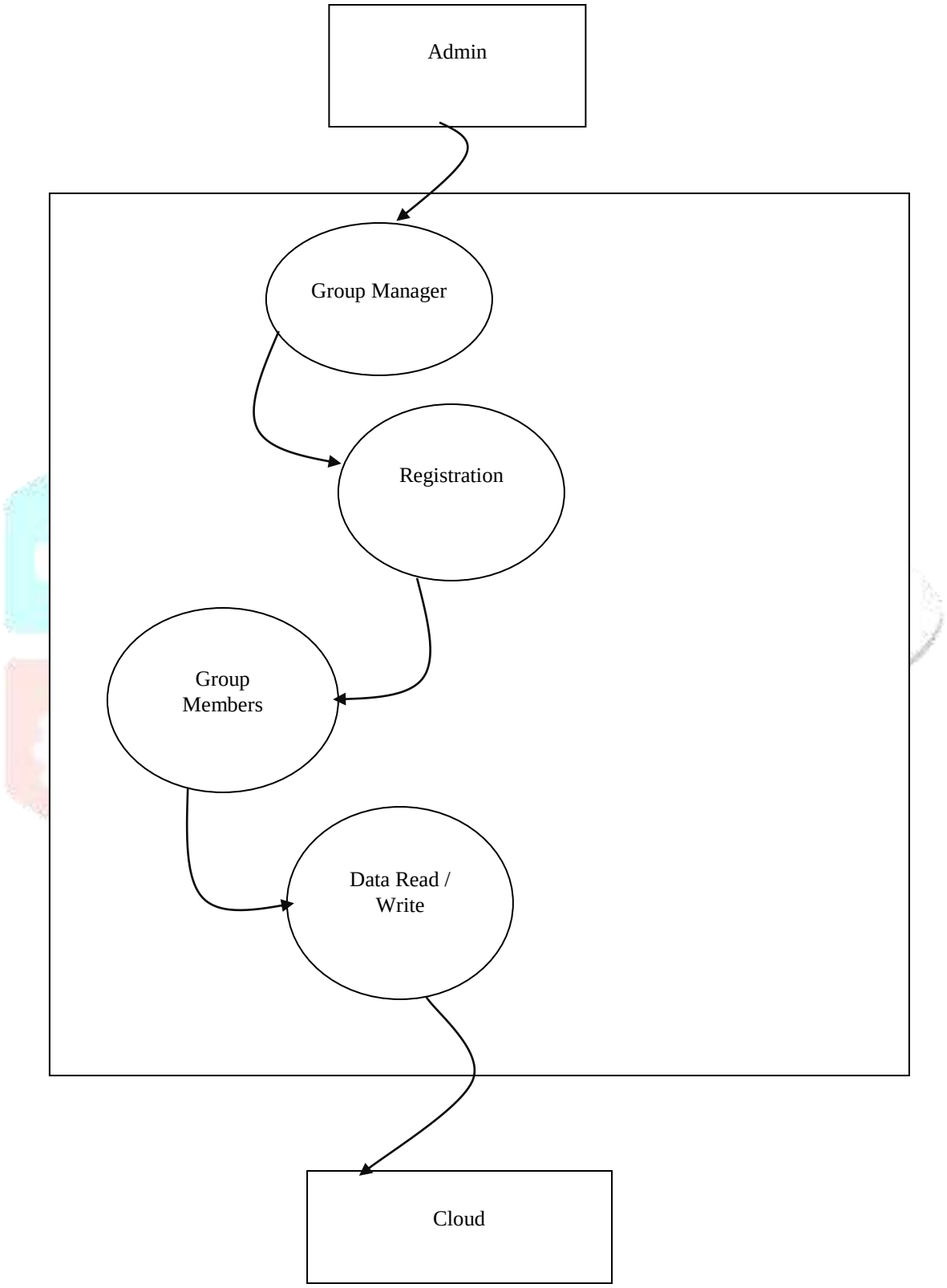
Data flow diagrams illustrate how data is processed by a system in terms of inputs and outputs. Data flow diagrams can be used to provide a clear representation of any business function. The technique starts with an overall picture of the business and continues by analyzing each of the functional areas of interest. This analysis can be carried out to precisely the level of detail required. The technique exploits a method called top-down expansion to conduct the analysis in a targeted way.

As the name suggests, Data Flow Diagram (DFD) is an illustration that explicates the passage of information in a process. A DFD can be easily drawn using simple symbols. Additionally, complicated processes can be easily automated by creating DFDs using easy-to-use, free downloadable diagramming tools. A DFD is a model for constructing and analyzing information processes. DFD illustrates the flow of information in a process depending upon the inputs and outputs. A DFD can also be referred to as a Process Model. A DFD demonstrates business or technical process with the support of the outside data saved, plus the data flowing from the process to another and the end results Level 0 DFD.

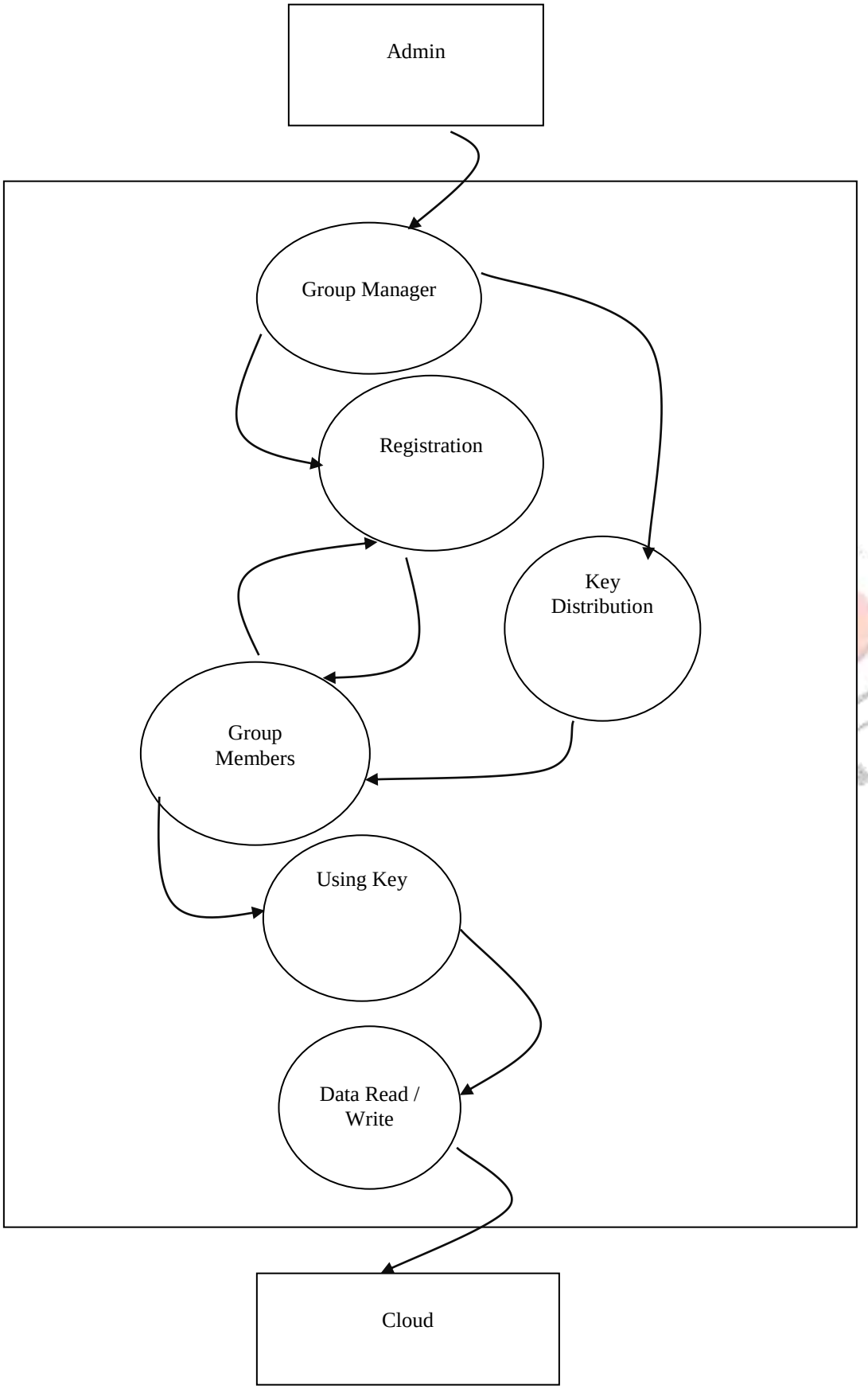
Level 0:



Level 2 DFD


Admin
Group Manager
Registration
Group
Members
Data Read /
Write
Cloud

Level 3 DFD


Admin
Group Manager
Registration
Key Distribution
Group Members
Using Key
Data Read / Write
Cloud

3.6 INTRODUCTION TO JDBC

JDBC stands for **J**ava **D**ata**b**ase **C**onnectivity, which is a standard Java API for database-independent connectivity between the Java programming language and a wide range of databases.

The JDBC library includes APIs for each of the tasks commonly associated with database usage:

- Making a connection to a database
- Creating SQL or MySQL statements
- Executing that SQL or MySQL queries in the database
- Viewing & Modifying the resulting records

Fundamentally, JDBC is a specification that provides a complete set of interfaces that allows for portable access to an underlying database. Java can be used to write different types of executables, such as:

- Java Applications
- Java Applets
- Java Servlets
- Java ServerPages (JSPs)
- Enterprise JavaBeans (EJBs)

All of these different executables are able to use a JDBC driver to access a database and take advantage of the stored data.

JDBC provides the same capabilities as ODBC, allowing Java programs to contain database-independent code.

3.7 JDBC ARCHITECTURE:

The JDBC API supports both two-tier and three-tier processing models for database access but in general JDBC Architecture consists of two layers:

- **JDBC API:** This provides the application-to-JDBC Manager connection.
- **JDBC Driver API:** This supports the JDBC Manager-to-Driver Connection.

The JDBC API uses a driver manager and database-specific drivers to provide transparent connectivity to heterogeneous databases.

The JDBC driver manager ensures that the correct driver is used to access each data source. The driver manager is capable of supporting multiple concurrent drivers connected to multiple heterogeneous databases.

Following is the architectural diagram, which shows the location of the driver manager with respect to the JDBC drivers and the Java application:

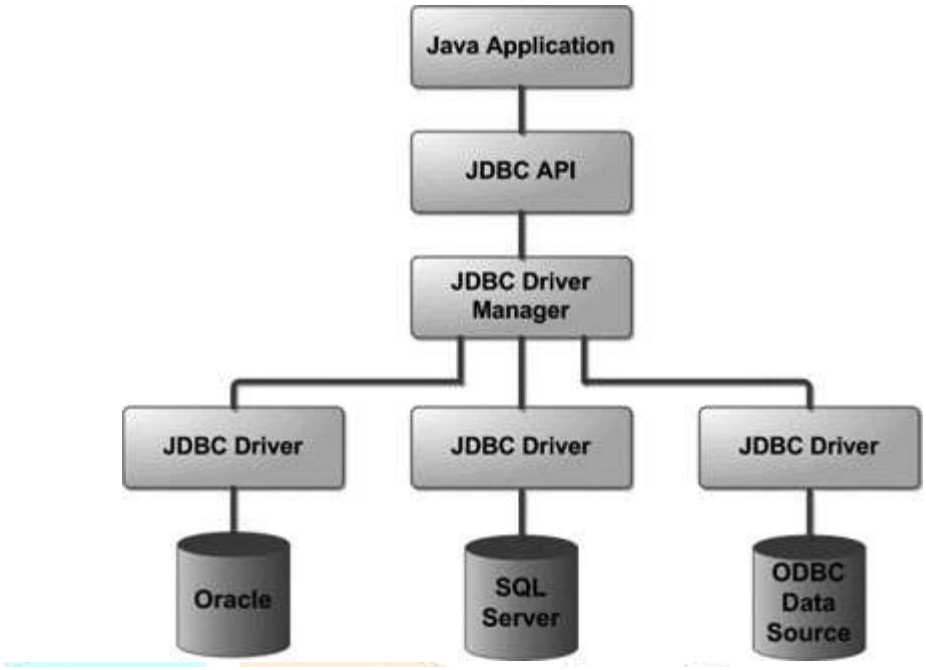


Fig.Common JDBC Components

IV. ADVANTAGES:

- We propose a secure multi-owner data sharing scheme. It implies that any user in the group can securely share data with others by the untrusted cloud.
- We provide secure and privacy-preserving access control to users, which guarantees an member in a group to anonymously utilize the cloud resource.
- Data sharing can enhanced understanding of results of an individual clinical trial and enable the pooling of data from multiple trial to extend scientific discoveries beyond those derivable from any single study.
- The the practical and scientific argument for data sharing include improving the accuracy of research, informing risk or benefits analysis of treatment options, strengthening collaborations, accelerating biomedical research, and restoring trust in clinical research enterprise.
- The participant level data are particularly useful when shared, but care nust be taken to avoid drawing inaccurate conclusions from reanalysis of such data.

V. CONCLUSION:

In this paper, we design a Distributed Accountability for Data Sharing in the Cloud, Mona, for dynamic groups in an untrusted cloud. In Mona, a user is able to share data with others in the group without revealing identity privacy to the cloud. Additionally, Mona supports efficient user revocation and new user joining. More specially, efficient user revocation can be achieved through a public revocation list without updating the private keys of the remaining users, and new users can directly decrypt files stored in the cloud before their participation. Moreover, the storage overhead and the encryption computation cost are constant. Extensive analyses show that our proposed scheme satisfies the desired security requirements and guarantees efficiency as well. proposed a cryptographic storage system that enables secure file sharing on untrusted servers, named Plutus. By dividing files into file groups and encrypting each file group with a unique file-block key, the data owner can share the file groups with others through delivering the corresponding lockbox key, where the lockbox key is used to encrypt the file-block keys. However, it brings about a heavy key distribution overhead for large-scale file sharing. Additionally, the file-block key needs to be updated and distributed again for a user revocation.

REFERENCES

[1] K. Kent and M. Souppaya. (1992). Guide to Computer Security Log Management, NIST SpecialPublication80092[Online].Available:http://csrc.nist.gov/publications/nistpubs/800-92/SP800-92.pdf

[2] U.S. Department of Health and Human Services. (2011, Sep.). HIPAA—General Information [Online]. Available: https://www.cms.gov/ hipaageninfo

[3] PCI Security Standards Council. (2006, Sep.) Payment Card Industry (PCI) Data Security Standard—Security Audit Procedures Version1.1 [Online].Available: https://www.pcisecuritystandards.org/pdfs/pci−audit−procedures−v1-1.pdf

[4] Sarbanes-Oxley Act 2002. (2002, Sep.). A Guide to the Sarbanes-Oxley Act [Online]. Available: http://www.soxlaw.com/

[5] C. Lonvick, The BSD Syslog Protocol, Request for Comment RFC 3164, Internet Engineering Task Force, Network Working Group, Aug. 2001.

[6] D. New and M. Rose, Reliable Delivery for Syslog, Request for Comment RFC 3195, Internet Engineering Task Force, Network Working Group, Nov. 2001.