International Journal of Ethics in Engineering & Management Education
Website: www.ijeee.in (ISSN: 2348-4748, Volume 1, Issue 5, May2014)

Identity-Based Secure Distributed Data Storage Schemes

Dr.M.V.Siva Prasad
Principal & Professor
Anurag Engg College, Kodad, Nalgonda

J.Nagaraju
Assistant Professor, CSE
Anurag Engg College, Kodad, Nalgonda

N.Sahithi
M. Tech Student, CSE
Anurag Engg College, Kodad, Nalgonda
sahithi.n64@gmail.com

Abstract: **Secure distributed data storage can shift the burden of maintaining a large number of files from the owner to proxy servers. Proxy servers can convert encrypted files for the owner to encrypted files for the receiver without the necessity of knowing the content of the original files. In practice, the original files will be removed by the owner for the sake of space efficiency. Hence, the issues on confidentiality and integrity of the outsourced data must be addressed carefully. In this paper, we propose two identity-based secure distributed data storage (IBSDDS) schemes. Our schemes can capture the following properties: The file owner can decide the access permission independently without the help of the private key generator (PKG); For one query, a receiver can only access one file, instead of all files of the owner; Our schemes are secure against the collusion attacks, namely even if the receiver can compromise the proxy servers, he cannot obtain the owner's secret key. Although the first scheme is only secure against the chosen plaintext attacks (CPA), the second scheme is secure against the chosen cipher text attacks (CCA). To the best of our knowledge, it is the first IBSDDS schemes where access permissions is made by the owner for an exact file and collusion attacks can be protected in the standard model.**

Keywords: **PKG, CCA, DAS, CFS, NFS, SBIDS, MAC, UML**

1. INTRODUCTION

Cloud computing provides users with a convenient mechanism to manage their personal files with the notion called database-as-a-service (DAS). In DAS schemes, a user can outsource his encrypted files to un trusted proxy servers. Proxy servers can perform some functions on the outsourced cipher texts without knowing anything about the original files. Unfortunately, this technique has not been employed extensively. The main reason lies in that users are especially concerned on the confidentiality, integrity and query of the outsourced files as cloud computing is a lot more complicated than the local data storage systems, as the cloud is managed by an un trusted third party. After out sourcing the files to proxy servers, the user will remove them from his local machine. Therefore, how to guarantee the out soured files are not accessed by the unauthorized users and not modified by proxy servers is an important problem that has been considered in the data storage research community. Furthermore, how to guarantee that an authorized user can query the outsourced files from proxy servers is another concern as the proxy server only maintains the outsourced cipher texts. Consequently, research around these topics grows significantly.

2. PROPOSED SYSTEM:

In this paper, we propose two identity-based secure distributed data storage (IBSDDS) schemes in standard model where, for one query, the receiver can only access one of the owner's files, instead of all files. In other words, access permission (re-encryption key) is bound not only to the identity of the receiver but also the file. The access permission can be decided by the owner, instead of the trusted party (PKG). Furthermore, our schemes are secure against the collusion attacks. ADVANTAGES OF PROPOSED SYSTEM are It has two schemes of security, the first scheme is CPA secure, the second scheme achieves CCA security, To the best of our knowledge, it is the *first* IBSDDS schemes where access permission is made by the owner for an exact file and collusion attacks can be protected in the standard model, To achieve a stronger security and implement file based access control, the owner must be online to authenticate requesters and also to generate access permissions for them. Therefore, the owner in our schemes needs do more computations than that in PRE schemes. Although PRE schemes can provide the similar functionalities of our schemes when the owner only has one file, these are not flexible and practical.

3. MODULES:

- Data Storage Systems
- File Systems.
- Storage-based Intrusion Detection Systems.
- Cryptographic File System.

Data Storage Systems: Data storage systems enable users to store their data to external proxy servers to enhance the access and availability, and reduce the maintenance cost. Samaratiand Vimercati. Addressed the privacy issues in data utility, and pointed out the main research directions in the protection of the externally stored data. Kherand Kim surveyed the data storage systems comprehensively and classified them into three kinds based on their security services: networked file systems (NFS), storage-based intrusion detection systems (SBIDS) and cryptographic file systems (CFS).
File Systems: In these systems, proxy servers are assumed to be trusted. They authenticate receivers and validate access permissions. The interactions between the proxy servers and receivers are executed in a secure channel. Therefore, these

International Journal of Ethics in Engineering & Management Education
Website: www.ijeee.in (ISSN: 2348-4748, Volume 1, Issue 5, May2014)

systems cannot provide an end-to-end data security, namely they cannot ensure the confidentiality of the data stored at the proxy server in these schemes, a receiver authenticates himself to the proxy server using his password. Then, the proxy sever passes the authentication result to the file owner. The owner will make access permission according to the received information.

Storage-based Intrusion Detection Systems: In these systems, an intrusion detection scheme is embedded in proxy servers or the file owner to detect the intruder's behaviors, such as adding backdoors, inserting Trojan horses and tampering with audit logs. These schemes can be classified into two types: host-based system and network-based system. In the host-based systems, an intrusion detection scheme is embedded in the host to detect the local intrusion actions. On the contrary, in network-based systems, an intrusion detection scheme is embedded in the proxy servers to detect the external intruder's actions. The main advantage of these systems is that proxy servers can still detect the intrusion action seven if the host is compromised as the proxy server is independent from the host.

Cryptographic File System: In these systems, an end to-end security is provided by cryptographic protocols which are executed by the file owner to prevent proxy servers and unauthorized users from modifying and accessing the sensitive files. These systems can be divided into two types: shared file system and non-shared system. In shared file systems the owner can share his files with a group of users. Cryptographic techniques deployed in these systems are key sharing, key agreement and key revocation. In non-shared file systems in order to share a file with another user, the owner can compute an access key for the user using his secret key. In these two systems, the integrity of the sensitive files is provided by digital signature schemes and message authentication codes (MAC).

4. SYSTEM ARCHITECTURE:

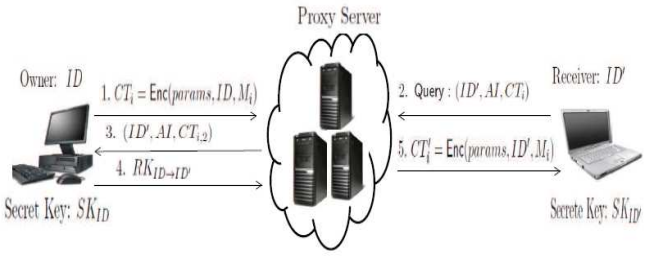


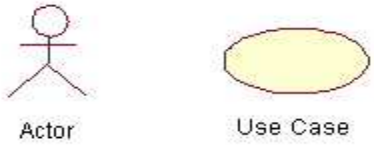
Fig. 1: The Model of Identity-Based Secure Distributed Data Storage Scheme

UML Diagrams: A diagram is the graphical presentation of a set of elements, most often rendered as a connected graph of vertices (things) and arcs (relationships).There are two types of diagrams, they are: Structural and Behavioral Diagrams

Structural Diagrams: The UML's four structural diagrams exist to visualize, specify, construct and document the static aspects of a system. ican View the static parts of a system using one of the following diagrams. Structural diagrams consist of Class Diagram, Object Diagram, Component Diagram, and Deployment Diagram.

Behavioral Diagrams: The UML's five behavioral diagrams are used to visualize, specify, construct, and document the dynamic aspects of a system. The UML's behavioral diagrams are roughly organized around the major ways which can model the dynamics of a system. Behavioral diagrams consists of Use case Diagram, Sequence Diagram, Collaboration Diagram, State chart Diagram, Activity Diagram

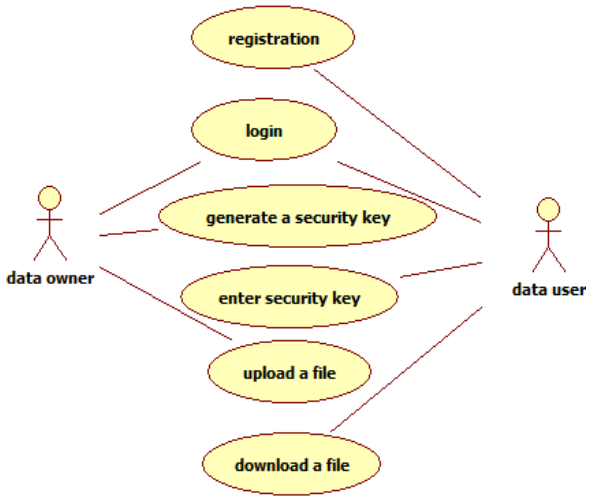
Use-Case diagram: A use case is a set of scenarios that describing an interaction between a user and a system. A use case diagram displays the relationship among actors and use cases. The two main components of a use case diagram are use cases and actors.



An actor is represents a user or another system that will interact with the system you are modeling. A use case is an external view of the system that represents some action the user might perform in order to complete a task.

Contents:

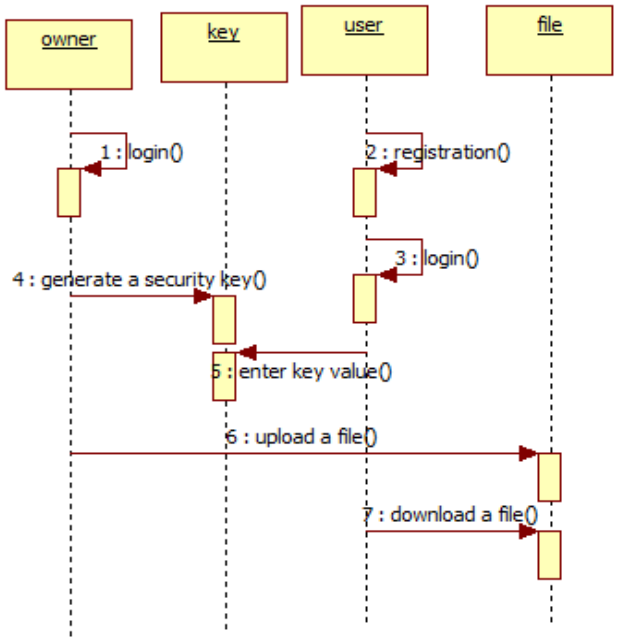
- Use cases
- Actors
- Dependency, Generalization, and association relationships
- System boundary



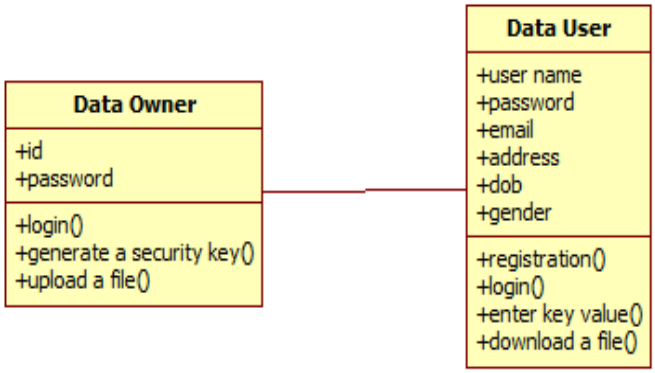
Link for more class diagram examples. UML Class Diagram with Relationships

Sequence Diagram: Sequence diagrams in UML shows how object interact with each other and the order those interactions occur. It's important to note that they show the interactions for a particular

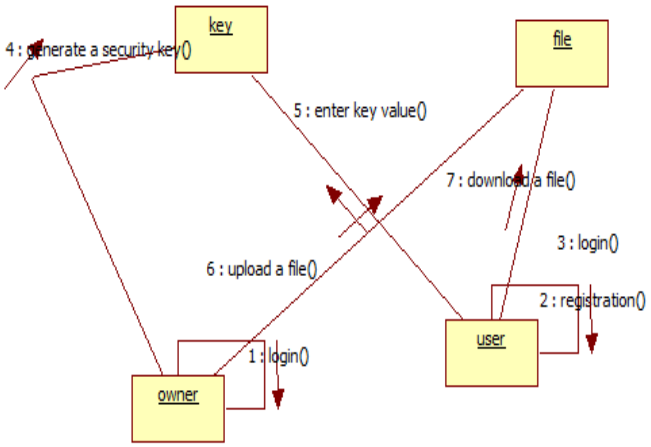
International Journal of Ethics in Engineering & Management Education
Website: www.ijeee.in (ISSN: 2348-4748, Volume 1, Issue 5, May2014)



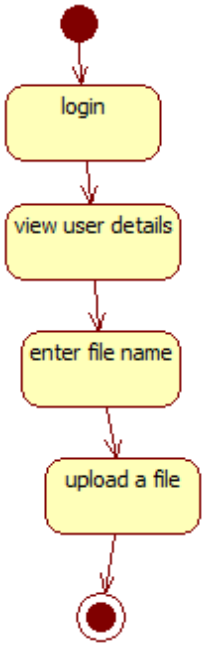
Class Diagram: Class diagrams are widely used to describe the types of objects in a system and their relationships. Class diagrams model class structure and contents using design elements such as classes, packages and objects. Class diagrams describe three different perspectives when designing a system, conceptual, specification, and implementation. These perspectives become evident as the diagram is created and help solidify the design. Class diagrams are arguably the most used UML diagram type. It is the main building block of any object oriented solution. It shows the classes in a system, attributes and operations of each class and the relationship between each class. In most modeling tools a class has three parts, name at the top, attributes in the middle and operations or methods at the bottom. In large systems with many classes related classes are grouped together to to create class diagrams. Different relationships between diagrams are show by different types of Arrows. Below is a image of a class diagram. Follow the scenario. The processes are represented vertically and interactions are show as arrows. This article explains the purpose and the basics of Sequence diagrams.



Collaboration diagram: Communication diagram was called collaboration diagram in UML 1. It is similar to sequence diagrams but the focus is on messages passed between objects. The same information can be represented using a sequence diagram and different objects. Click here to understand the differences using an example.



State machine diagrams: State machine diagrams are similar to activity diagrams although notations and usage changes a bit. They are sometime known as state diagrams or start chart diagrams as well. These are very useful to describe the behavior of objects that act different according to the state they are at the moment. Below State machine diagram show the basic states and actions. State Machine diagram in UML, sometime referred to as State or State chart diagram

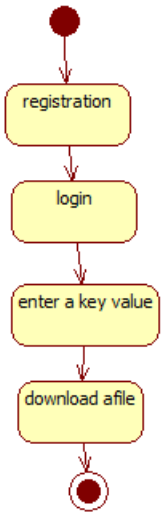


Activity Diagram: Activity diagrams describe the workflow behavior of a system. Activity diagrams are similar to state diagrams because activities are the state of doing something. The diagrams describe the state of activities by showing the sequence of activities performed. Activity diagrams can show activities that are conditional or parallel.

International Journal of Ethics in Engineering & Management Education
Website: www.ijeee.in (ISSN: 2348-4748, Volume 1, Issue 5, May2014)

How to Draw: Activity Diagrams: Activity diagrams show the flow of activities through the system. Diagrams are read from top to bottom and have branches and forks to describe conditions and parallel activities. A fork is used when multiple activities are occurring at the same time. The diagram below shows a fork after activity1. This indicates that both activity2 and activity3 are occurring at the same time. After activity2 there is a branch. The branch describes what activities will take place based on a set of conditions. All branches at some point are followed by a merge to indicate the end of the conditional behavior started by that branch. After the merge all of the parallel activities must be combined by a join before transitioning into the final activity state.

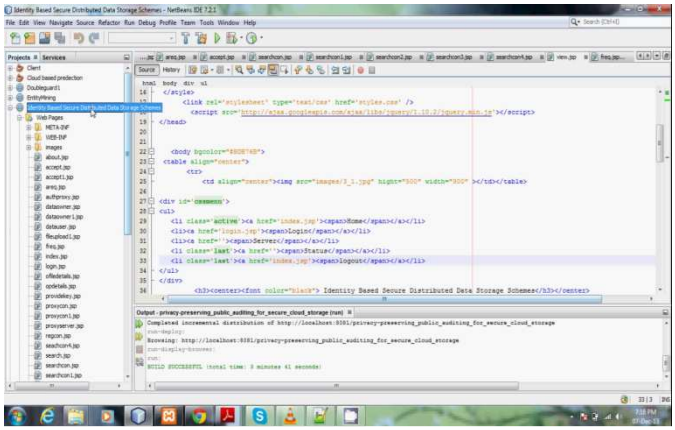
When to Use: Activity Diagrams: Activity diagrams should be used in conjunction with other modeling techniques such as interaction diagrams and state diagrams. The main reason to use activity diagrams is to model the workflow behind the system being designed. Activity Diagrams are also useful for: analyzing a use case by describing what actions need to take place and when they should occur; describing a complicated sequential algorithm; and modeling applications with parallel processes.

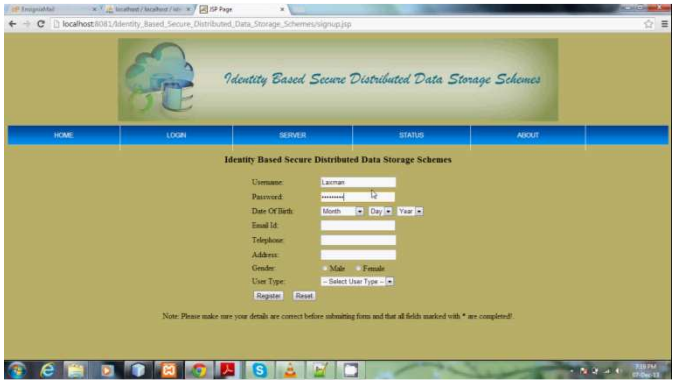


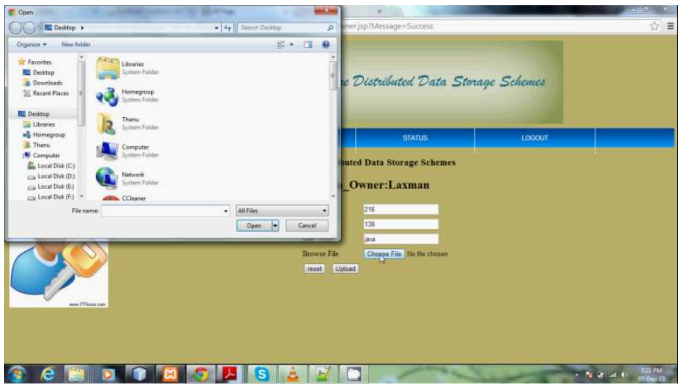
Component diagram: A component diagram displays the structural relationship of components of a software system. These are mostly used when working with complex systems that have many components. Components communicate with each other using interfaces. The interfaces are linked using connectors. Below images shows a component diagram.

Deployment Diagram: A deployment diagrams shows the hardware of your system and the software in those hardware. Deployment diagrams are useful when your software solution is deployed across multiple machines with each having a unique configuration. Below is an example deployment diagram.

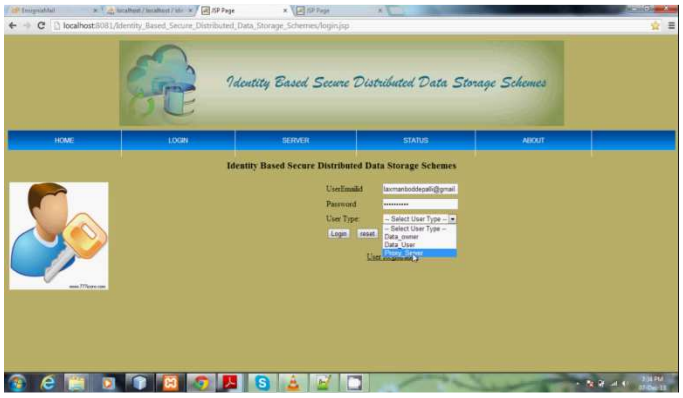
5. SCREEN SHOTS







International Journal of Ethics in Engineering & Management Education
Website: www.ijeee.in (ISSN: 2348-4748, Volume 1, Issue 5, May2014)



CONCLUSION

Distributed data storage schemes provide the users with convenience to outsource their files to un trusted proxy servers. Identity-based secure distributed data storage (IBSDDS) schemes are a special kind of distributed data storage schemes where users are identified by their identities and can communicate without the need of verifying the public key certificates. In this paper, we proposed two new IBSDDS schemes in standard model where, for one query, the receiver can only access one file, instead of all files. Furthermore, the access permission can be made by the owner, instead of the trusted party. Notably, our schemes are secure against the collusion attacks. The first scheme is CPA secure, while the second one is CCA secure.

REFERENCE

[1]. H. Hacigu¨mu¨s, B. R. Iyer, C. Li, and S. Mehrotra, “Executing SQL over encrypted data in the database-service-provider model,” inProceedings: SIGMOD Conference - SIGMOD’02 (M. J. Franklin,B. Moon, and A. Ailamaki, eds.), vol. 2002, (Madison, Wisconsin,USA), pp. 216–227, ACM, Jun. 2002.

[2]. L. Bouganim and P. Pucheral, “Chip-secured data access: Confi-dential data on untrusted servers,” in Proc. International Conferenceon Very Large Data Bases - VLDB’02, (Hong Kong, China), pp. 131–142, Morgan Kaufmann, Aug. 2002.

[3]. U. Maheshwari, R. Vingralek, and W. Shapiro, “How to build a trusted database system on untrusted storage,” in Proc. Symposium on Operating System Design and Implementation - OSDI’00, (San Diego, California, USA), pp. 135–150, USENIX, Oct. 2000.

[4]. A. Ivan and Y. Dodis, “Proxy cryptography revisited,” in Proc.Network and Distributed System Security Symposium - NDSS’03, (San Diego, California, USA), pp. 1–20, The Internet Society, Feb. 2003.H. Shacham and B. Waters, “Compact proofs of retrievability,” in Proc. Advances in Cryptology -ASIACRYPT’08 (J. Pieprzyk, ed.), vol. 5350 of Lecture Notes in Computer Science, (Melbourne, Australia), pp. 90–107, Springer, Dec. 2008

About the Authors:

Dr. M.V. Siva Prasad, Principal of Anurag Engineering College .He received B.E. [CSE] from Gulbarga University, M.Tech. [SE] from VTU, Belgaum and He was awarded Ph.D from Nagarjuna University, Guntur. He published number of papers in International & National journals. He is a Life member of ISTE M. No. : LM 53293 / 2007. His research interests are Information Security, Web Services, Mobile Computing, Data mining and Knowledge.

J.Nagaraju received Master of Technology [CSE] from JNTU- K. His research interests are Information Security, Web Services, Mobile Computing, Data mining and Knowledge.

N.Sahithi pursuing Master of Technology [Computer Science] from Anurag college of engineering kodad,She received B-tech[CSE] from Nagarjuna Institute of Technology and sciences(NITS) Miryalguda, Nalgonda District. Her research interests are Network Security,Web Services and Information Security.