Grenze International Journal of Engineering and Technology, Special issue

File Syncing-and-Sharing using Efficient Traitor Tracing and Revocation

Abinesh.R[1], Lavanya.S[2] and Dhinesh.P[3] and Gnanaprakash.P[4]
[1, 3, 4]UG Scholar,B.Tech(IT), Kongunadu College Of Engineering and Technology,Trichy, Tamilnadu, India.
abinesh621315@gmail.com, lavsen89@gmail.com, mohandinesh1997@gmail.com, gnanaprakash1218@gmail.com
[2]Assistant Professor, Information technology, Kongunadu College of Engineering and Technology,Trichy, Tamilnadu, India

***Abstract*—Recently, many more enterprises have moved their data into the cloud by using file syncing and sharing (FSS) service, but bring-your-own-device (BYOD) policies and greatly increasing mobile devices have in fact raised a new challenge for preventing the player/decoder abuse in the FSS service. In this paper, we address this issue using a new system model with anomaly detection, tracing and revoking traitors. To implement this model, we present a new threshold cryptosystem, called Partially-ordered Hierarchical Encryption (PHE), which implements the partial-order key hierarchy, similar to role hierarchy in Hierarchical RBAC, in public-keyinfrastructure. This cryptosystem provides two security mechanisms, traitor tracing and revocation, to support efficient digital forensics.The security and performance analysis shows that our construction is threshold provably secure and has following features: dynamicjoining and revoking users, constant-size ciphertexts and decryption keys, lower overloads for large-scale systems.**

***Index Terms*— Security, Cloud Storage, File Syncing-and-Sharing, Partial Order Key Hierarchy, Traitor Tracing, Revocation.**

I. INTRODUCTION

In recent years, many cloud storage services, such as Box,Dropbox, MediaFire, SkyDrive, SugarSync, have been availableto small-to-medium business, and individual. These cloudbased storage could be particularly attractive for consumers by providing on demand capacity, low-cost service, and longterm archive. Furthermore, cloud services have brought great convenience to people's lives because consumers can access applications and data from the cloud anywhere in the world and via any available device, such as personal computers, tables, and mobile phones. Therefore, many more enterprises and individuals have moved their data, such as personal data and large archive system, into the cloud everyday. The cloud has become a necessity to many of us for individual, enterprise,and government use Bring-your-own-device (BYOD) policies and an increasingly mobile devices are changing the requirements for how users want (and need) to access corporate data. The cloud storage provide these capabilities in a secure manner that gives IT over- sight and control.

Online player/editor: It provides the ability to access this data from any location and any of their devices, including smart-phones and tablets, without having to go through a corporate VPN or firewall (shown in the middle module in the figure).

End users: The FSS service also provide the ability to share information with other users, both inside and outside the organization.

Grenze ID: 01.GIJET.5.1.53

A. New System Model for FSS

We present a new FSS model for player abuse prevention and enhanced protection against unauthorized access. The proposed model uses the hierarchical role-based access control (H-RBAC) model, which is recognized for its support for simplified administration and scalability of collaboration and working with teams. Moreover, the design of this model is generic enough to support other access control policies, such as discretionary access control and multilevel security.

We show such a cloud-based FSS model in the Fig. 4. This model that addresses and incorporates the afore-mentioned authorization requirements can be built using three types of components:

Anomaly detection: This is used for detecting abnormal play- ers. More exactly, it is responsible for monitoring deployed resources and might allocate or release them to ensure the compliance of enterprise-side existing access control system. The output of this module is some suspected anomaly players.

Tracing traitors: This is responsible for finding out the traitors from the suspected players recognized in the previous step. In some cases this is simple and straightforward, but such a practice procedure sometimes results in solution difficulties if we request that the secrets or keys stored in the player cannot be leaked in the tracing procedure. We call it 'black box tracing'.

Revoking traitors: This is responsible for revoking the author- ity (or license) of traitors found in the previous step. The simple revocation method (e.g., the license is appeared in Blacklist) may be evaded in the way of license forgery and tampering. Taking into account the difficulty in comparing cryptographic key forgery and license forgery, the key- based revocation would be a more effective and secure manner.

Anomaly Detection and Key Hierarchy Obviously, the anomaly detection is an important module for protecting the outsourcing data against unauthorized access and finding out the suspected players in our system model. Some potential abnormal behaviors are listed as follows:

To recognize unauthorized access, e.g., anomaly behavior of players against RBAC policies and the license terms (such as a period of time, days, weeks, months, etc.), privilege escalation (PE) [31] attacks, etc.

To detect non-normal distribution exceeding the preset bounds, e.g., the unregistered devices away from mobile device management (MDM) [32], a user's license (or key) used into other unauthorized users or devices, etc.

To identify the player abuse outside the permitted range, e.g., the request to access a sensitive resource in the extra working time or region, etc.

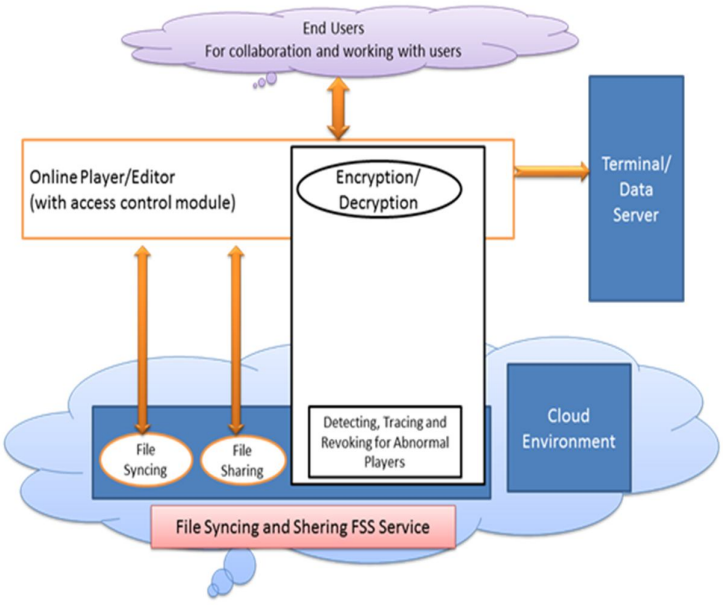


Fig2.1.1 system architecture

B. UML Diagram

The Unified Modeling Language (UML) is a general-purpose, developmental, modeling language in the field of software engineering that is intended to provide a standard way to visualize the design of a system.

C. Use Case Diagram

Use case diagrams are usually referred to as behavior diagrams used to describe a set of actions (use cases) that some system or systems (subject) should or can perform in collaboration with one or more external users of the system.

D. Activity Diagram

An activity diagram visually presents a series of actions or flow of control in a system similar to a flowchart or a data flow diagram. Activity diagrams are often used in business process modelling. They can also describe the steps in a use case diagram. Activities modelled can be sequential and concurrent.

E. Sequence Diagram

The sequence diagram is a good diagram to use to document a system's requirements and to flush out a system's design. The reason the sequence diagram is so useful is because it shows the interaction logic between the objects in the system in the time order that the interactions take place.

F. Collaboration Diagram

A collaboration diagram, also called a communication diagram or interaction diagram, is an illustration of the relationships and interactions among software objects in the Unified Modeling Language (UML).

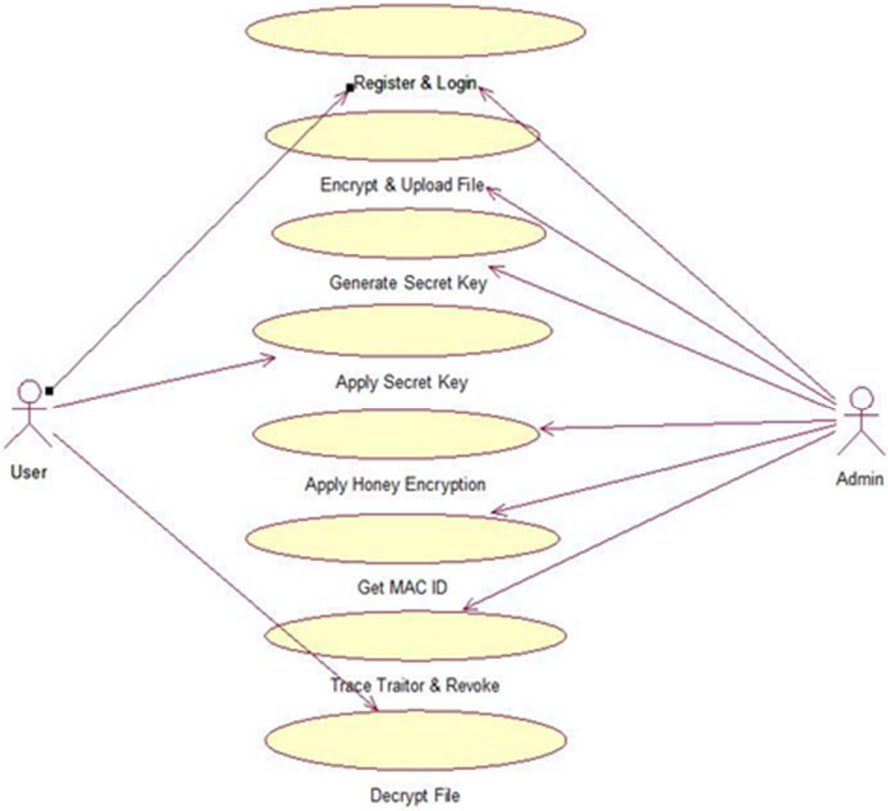
II. MODULE DESCRIPTION

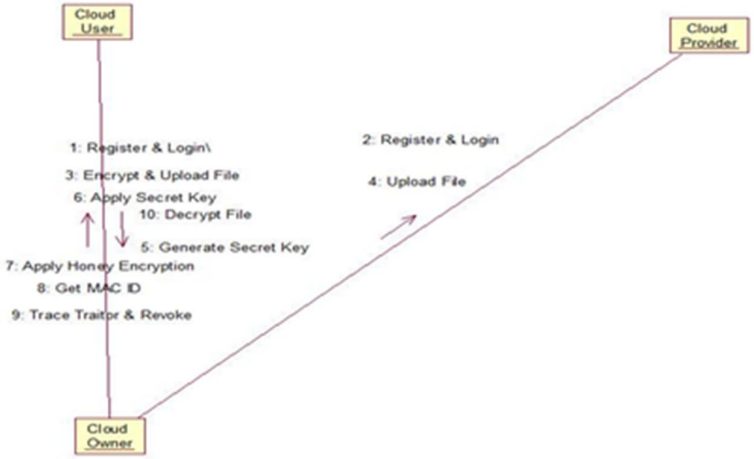
A. Cloud Framework

In this module, cloud data storage service three different entities such as the cloud user, who has large amount of data files to be stored in the cloud; the third-party auditor, who has expertise and capabilities
that cloud users do not have and is trusted to assess the cloud storage service reliability on behalf of the user upon request.

B. Key Generation

Merkle hash tree contains three algorithms: KeyGen, Sign and Verify. In KeyGen, each user in the group generates his/her public key and private key. In Sign, a user in the group is able to generate a signature on a block and its block identifier with his/her private key and all the group members' public keys.





C. Detect Anomaly and Pattern

This is used for detecting abnormal players. More exactly, it is responsible for monitoring deployed resources and might allocate or release them to ensure the compliance of enterprise-side existing access control system. The output of this module is some suspected anomaly players.

D. Traitor Tracing and Revoke Traitors

This is responsible for finding out the traitors from the suspected players recognized in the previous step. In some cases this is simple and straightforward, but such a practice procedure sometimes results in solution difficulties if we request that the secrets or keys stored in the player cannot be leaked in the tracing procedure.

We call it ‘black box tracing’.This is responsible for revoking the authority (or license) of traitors found in the previous step.

E. File Syncing and Sharing

File sharing: it allows the users to not only access files anywhere, anytime and from a variety of endpoint devices, but also collaboratively edit file together;
File syncing: it is a new online backup mechanism for syncing data across multiple devices, such as a home computer, tablet or smart phone, as well as collaboration and working with teams.

F. Performance Evaluation

We evaluate the performances of the proposed schemes based on the following factors: bandwidth, user’s storage, computation costs and the number of keys. For secure key hierarchy $\Psi = \_C,E,K\_$, let n is the number of classes, m is the average number of users in each class, $\overline{}$n is the number of partial order relations, l is the number of roots, and t is the value of threshold.

III. System Testing

A. Unit Testing

It is the testing of an individual unit or group of related units. It is done by programmer to test that the implementation is producing expected output against given input and it falls under white box testing. Unit testing is done in order to check registration whether the user properly registered into the cloud. It is done in order to check whether a file is properly uploaded into the cloud.

B. Integration Testing

All the modules should be integrated into a single module and it should be checked that it is still working still by integration testing.

C. System Testing

It is done to ensure that by putting the software in different environments and check that it still works. System Testing is done by uploading same file in this cloud checking whether any duplicate file exists.

D. Software Testing

It is the process of evaluating a software item to detect differences between given input and expected output. Also to assess the feature of a software item. Testing assess the quality of the product. It is a process that should be done during the development process. In other words software testing is a verification and validation process.
There are two types of software testing.
1. Black box testing
2. White box testing

E. Verification

Verification is the process to make sure the product satisfies the conditions imposed at the start of the development phase. In other words, to make sure the product behaves the way we want it to.

F. Validation

Validation is the process to make sure the product satisfies the specified requirements at the end of the development phase. In other words, to make sure the product is built as per customer requirements.

G. Black Box Testing

Black box testing is a testing which ignores internal mechanism of system and focuses on output generated against any input and execution of system. It is done for validation. It is done to check encryption and decryption after uploading a file into the cloud.

H. White Box Testing

It is done for verification and it is a testing that takes into account the internal mechanism of the system. It is done by checking content verification. It will verify that whether same content exists in the cloud.

IV. Conclusion & Future Enhancement

In this paper, we focus on protection the privacy of outsourcing data and preventing player abuse in file syncing and sharing services in the cloud. We highlight the development of a group-oriented cryptosystem with digital forensics, especially for tracing and revoking methods that can ensure the security of player/editor. Based on this cryptosystem, we present a new secure service model to provide a forensic analysis framework to guide investigations. In our future work, we are planning to introduce a comprehensive anomaly detection, using audit, pattern matching, and risk assessment, for identifying the suspected players.

References

[1] Z. Liu, Z. Cao, and D. S. Wong, "Traceable Cp-abe: A How To Trace Decryption Devices Found In The Wild" Vol.10,No.1,Pg.338-353.

[2] D. Boneh and M. K. Franklin, "Identity-based Encryption From The Weil Pairing," in CRYPTO,2001,Pg.213-229.

[3] Sahai and B. Waters, "Fuzzy Identity-based Encryption" in EUROCRYPT,2005,Pg.457-473.

[4] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data" in ACM Conference On CCS,2006,Pg.89-98.

[5] R. Ostrovsky, A. Sahai, and B.Waters, "Attribute-based encryption with non-monotonic access structures" in ACM Conference On computer and communications security, 2007, Pg.193-203.

[6] D.Quick and K.R.Choo,Google Drive:Forensic Analysis Of Data Remnants," J.Network And Computer Applications, Vol. 40, Pg.179-193,2014.

[7] D.Boneh and M.K.Franklin,"An Efficient Public Key Traitor Tracing Scheme," in CRYPTO, 1999,Pg.338-353.