Authentication And Monitoring in Cloud Computing by Fingerprint and Iris Scan

Stephy Paul PS[1], Mrs. Aiswarya Lekshmi A.C[2]
[1]*Department of Computer Science and Engineering*
[2]*Assistant Professor, Department of Computer Science and Engineering Sivaji College of Engineering and Technology, Parasuvaikkal, Tamil Nadu 695504, India*

Abstract: Cloud storage can provide powerful and on-demand data storage services for users. By using the cloud service, users can outsource their data to the cloud without wasting substantial maintenance expenditure of hardware and software, which brings great benefits to users. However, once the users upload their data to the cloud, they will lose the physical control of their data since they no longer keep their data in local. Thus, the integrity of the cloud data is hard to be guaranteed, due to the inevitable hardware/software failures and human errors in the cloud. Using cloud storage services, users can store their data in the cloud to avoid the expenditure of local data storage and maintenance. To ensure the integrity of the data stored in the cloud, many data integrity auditing schemes have been proposed. In most, if not all, of the existing schemes, a user needs to employ his private key to generate the data authenticators for realizing the data integrity auditing. Thus, the user has to possess a hardware token (e.g. USB token, smart card) to store his private key and memorize a password to activate this private key. If this hardware token is lost or this password is forgotten, most of the current data integrity auditing schemes would be unable to work. In order to overcome this problem, we propose a new paradigm called data integrity auditing without private key storage and design such a scheme. In this scheme, we use biometric data (e.g. iris scan, fingerprint) as the user's fuzzy private key to avoid using the hardware token. Meanwhile, the scheme can still effectively complete the data integrity auditing. We utilize a linear sketch with coding and error correction processes to confirm the identity of the user. In addition, we design a new signature scheme which not only supports block less verifiability, but also is compatible with the linear sketch. The security proof and the performance analysis show that our proposed scheme achieves desirable security and efficiency

Index Terms- TPA- Third Party Auditor, IBE-Identity Based Encryption, ASP-Active Server Pages, CLR - Common Language Run, DLL -Dynamic Link Library, DFD -Data Flow Diagram, UML-Unified Modelling Language

I. INTRODUCTION

Cloud storage can provide powerful and on-demand data storage services for users. By using the cloud service, users can outsource their data to the cloud without wasting substantial maintenance expenditure of hardware and software, which brings great benefits to users. However, once the users upload their data to the cloud, they will lose the physical control of their data since they no longer keep their data in local. Thus, the integrity of the cloud data is hard to be guaranteed, due to the inevitable hardware/software failures and human errors in the cloud. Many data integrity auditing schemes have been proposed to allow either the data owner or the Third Party Auditor (TPA) to check whether the data stored in the cloud is intact or not. These schemes focus on different aspects of data integrity auditing, such as data dynamic operation, the privacy protection of data and user identities, key exposure resilience, the simplification of certificate management and privacy-preserving authenticators, etc. In the above data integrity auditing schemes, the user needs to generate authenticators for data blocks with his private key. It means that the user has to store and manage his private key in a secure manner. In general, the user needs a portable secure hardware token (e.g. USB token, smart card) to store his private key and memorizes a password that is used to activate this private key. The user might need to remember multiple passwords for different secure applications in practical scenarios, which is not user friendly. In addition, the hardware token that contains the private key might be lost. Once the password is forgotten or the hardware token is lost, the user would no longer be able to generate the authenticator for any new data block.

The data integrity auditing will not be functioning as usual. Therefore, it is very interesting and appealing to find a method to realize data integrity auditing without storing the private key.

We design a practical data integrity auditing scheme without private key storage for secure cloud storage. In our scheme, two fuzzy private keys (biometric data) are extracted from the user in the phase of registration and the phase of signature generation. We respectively use these two fuzzy private keys to generate two linear sketches that contain coding and error correction processes. In order to confirm the user's identity, we compare these two fuzzy private keys by removing the "noise" from two sketches. If the two biometric data are sufficiently close, we can confirm that they are extracted from the same user; otherwise, from different users. How to design a signature satisfying both the compatibility with the linear sketch and the block less verifiability is a key challenge for realizing data integrity auditing without private key storage. In order to overcome this challenge, we design a new signature scheme named as MBLSS by modifying the BLS short signature based on the idea of fuzzy signature. We give the security analysis and justify the performance via concrete implementations. The results show that the proposed scheme is secure and efficient.

II. LITERATURE REVIEW

PRIVACY-PRESERVING PUBLIC AUDITING FOR SECURECLOUD STORAGE

Using Cloud Storage, users can remotely store their data and enjoy the on-demand high quality applications and services from a shared pool of configurable computing resources, without the burden of local data storage and maintenance. However, the fact that users no longer have physical possession of the outsourced data makes the data integrity protection in Cloud Computing a formidable task, especially for users with constrained computing resources. Moreover, users should be able to just use the cloud storage as if it is local, without worrying about the need to verify its integrity. Thus, enabling public auditability for cloud storage is of critical importance so that users can resort to a third party auditor (TPA) to check the integrity of outsourced data and be worry-free. To securely introduce an effective TPA, the auditing process should bring in no new vulnerabilities towards user data privacy, and introduce no additional online burden to user. In this paper, we propose a secure cloud storage system supporting privacy-preserving public auditing. We further extend our result to enable the TPA to perform audits for multiple users simultaneously and efficiently. Extensive security and performance analysis show the proposed schemes are provably secure and highly efficient.

III. SYSTEM ANALYSIS OVERVIEW

System analysis is the starting point for system design. System Analysis is a detailed study of various operations performed by a system and its relationships within and outside the system. The basic aim of system analysis is to obtain a clear understanding of the needs of all users, what exactly is decided from the software and what the constraints on the solutions. System Analysis refers to the process of examining a situation with the intention of improving it through better process and methods. System analysis is therefore, the process of gathering and interpreting facts, diagnosing problem and using the information to recommend information in system or in other words, it means a detailed explanation or description. Before computerizing a system under consideration, it has to be analyzed. We need to study how it functions currently, what are problems and what are requirements that proposed software should meet.

IV. EXISTING SYSTEM

In existing system they implement a cryptographic key to encrypt the data this key is generated using some random key or any algorithm .so theses key can be easily miss used by the attacker's. If someone use our username or password they can easily access our file without any authentication. To protect the confidentiality of the shared data, the cryptographic schemes are usually applied. The security of cryptographic schemes stem from the security of underlying cryptographic key. Currently, the cryptographic key is simply stored in the computer in most of existing cryptographic schemes. While it has been reported that the stored keys can be revealed by some viruses. To deal with the key exposure problem, many techniques have been proposed, such as key-insulated public key technique [9], [10], and parallel key insulated public key technique. Focus on different aspects of data integrity auditing, such as data dynamic

operation, the privacy protection and user identities, key exposure resilience, the simplification of certificate management and privacy-preserving authenticators etc. The user needs to generate authenticators for data blocks with his private key. The user has to store and manage his private key in a secure manner

V. PROPOSED SYSTEM

In proposed system we implemented a secure way to protect our data from attackers. We implemented a 3 lever security system to secure our data. For thatwe first develop a device security only with the use of this device a user can login to their particular account. Then we implemented a secure way to generate the key with the help of device we generated key for encryption and decryption. Auditor will monitor the accessibility of the file and users

- An efficient data integrity auditing mechanism is introduced.
- Use biometric data as the user's fuzzy private key to avoid using the hardware token.
- Utilize a linear sketch with coding and error correction processes to confirm the identity of the user.

VI. SYSTEM DESIGN

- INTRODUCTION

System Design is the process of planning of new system or to replace or complement an existing system .It must be thoroughly understood about the old system and determine how computers can be used to make its operations more effective. System design sits at technical the kernel of system development. Once system requirements have been analyzed and specified system design is the first of the technical activities-design, code generation and test- that required build and verifying the software. System design is the most creative and challenging phases of the system life cycle. The term design describes the final system and the process by which it is to be developed.

- INPUT DESIGN

Input design is the method by which valid data are accepted from the user. This part of the designing requires very careful attention. If the data going into the system is incorrect then the processing and output will magnify these errors. Inaccurate input data are the most common cause of errors in data processing.

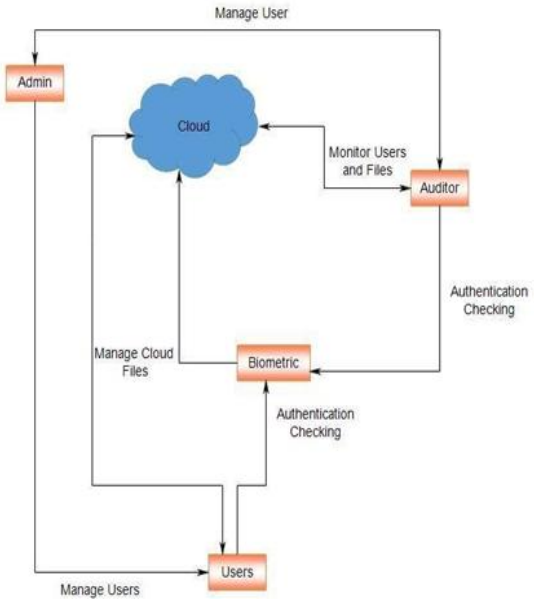
Controlling Amount of Input: Wherever user input is required, giving possible input values as default in that area reduces the amount of user keystrokes. Thus the user can pass on to next data without much typing. This makes the data entry much fast and error free. When the user has the format of input to be given, it will be very easy for the user to give input in the same format.

Avoiding Errors in Data: The rate at which errors occur depends on the quantity of the data. As told in the above objective these errors are reduced by making the number of data to be entered in each form is reduced.

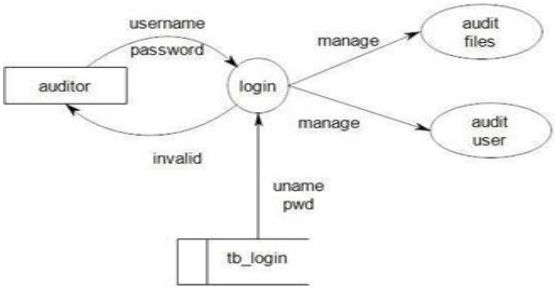
- OUTPUT DESIGN

Output design is one of the most important features of the information system. When the output is not of good quality, the users will be averse to use the newly designed system and may not use the system. There are many types of outputs, all of which can be either highly useful or can be critical to the users, depending on the manner and degree to which they are used. Outputs from computer system are required primarily to communicate the results of processing to users. They are also used to provide a permanent hard copy of the results for later consultation.

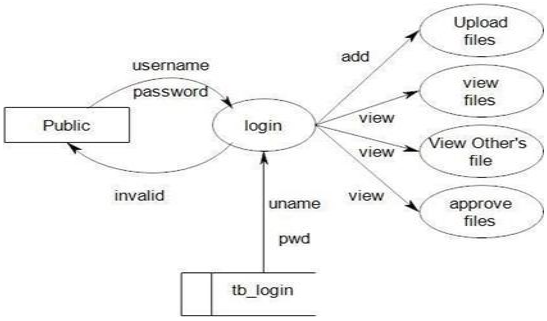
VII. SYSTEM ARCHITECTURE



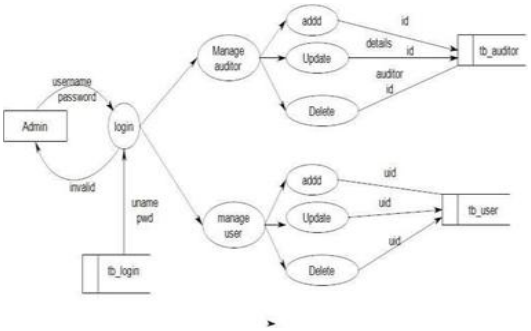
- DFD FOR BIOMETRIC AUTHENTICATION



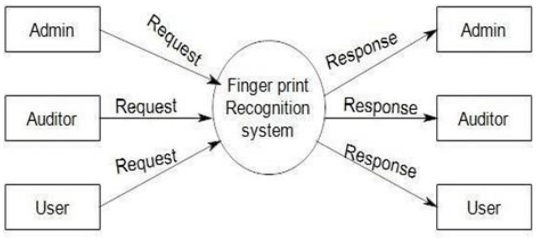
- DFD FOR ADMIN



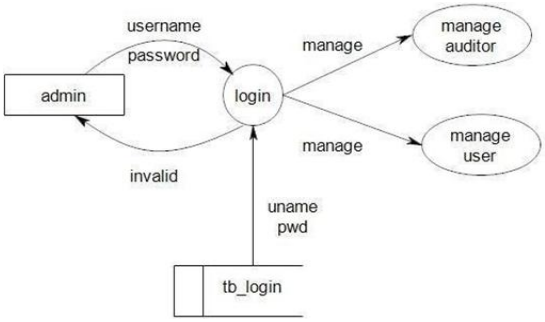
- DFD FOR AUDITOR



DFD FOR PUBLIC USER

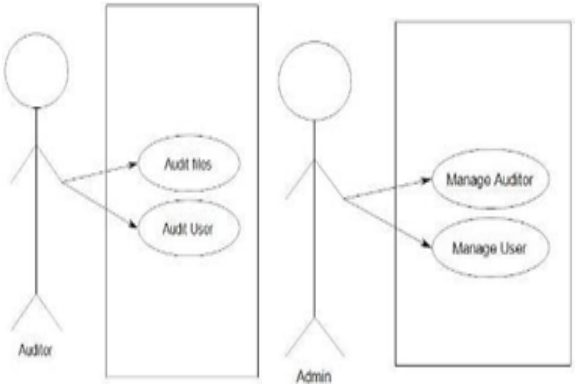


- DFD FOR ADMIN OPERATION



- USE CASE DIAGRAM

In software engineering, a use case diagram in the Unified Modelling Language (UML) is a type of behavioural diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals, and any dependencies between those use cases.



VIII. SYSTEM IMPLEMENTATION MODULES:

ADMIN

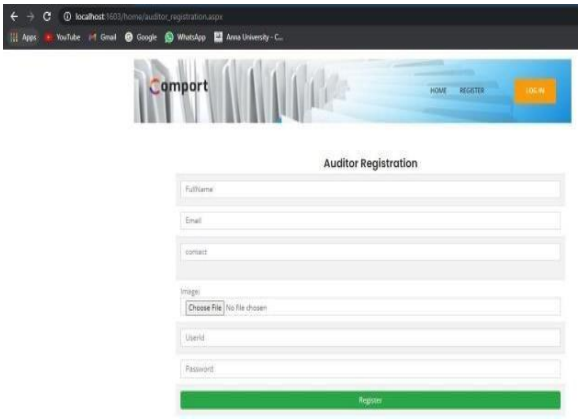
The Admin will have the full power to access the system. The admin will manage users. they have the rights to accept and reject each users in our system. They have also had the power of monitoring the auditor that who verify individual users and files. They can also view the integrity checking results from the auditor.

USERS

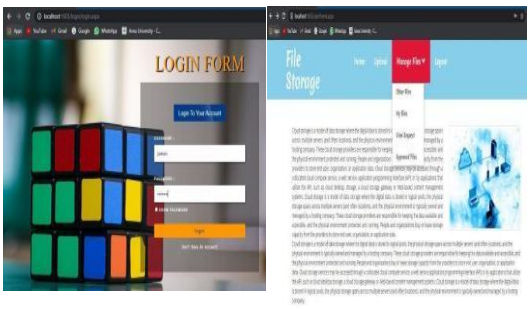
In the phase of user registration, the biometric data (e.g. fingerprint) is extracted from the user who wants to use the cloud storage service. When a data owner would like to upload data to the cloud, he firstly extracts biometric data as his fuzzy private key and randomly generates a signing key. Users also login to our system using there biometrics when they login they can view the integrity checking of their files. They upload the files to the cloud for provide security employee will encrypt the report and outsource to the cloud storage and the employee can set the privacy for the uploaded data. They can also provide different privacy for different users for accessing ones file.

IX. SNAPSHOTS: HOME PAGE

USER REGISTRATION FORM



USER LOGIN FORM & HOMEPAGE



X. CONCLUSION

In this paper, we explore how to employ fuzzy private key to realize data integrity auditing without storing private key. We propose the first practical data integrity auditing scheme without private key storage for secure cloud storage. In the proposed scheme, we utilize biometric data (e.g. fingerprint, iris scan) as user's fuzzy private key to achieve data integrity auditing without private key storage. In addition, we design a signature scheme supporting block less verifiability and the compatibility with the linear sketch. The formal security proof and the performance analysis show that our proposed scheme is provably secure and efficient.

REFERENCE

[1] A. F. Barsoum and M. A. Hasan, "Provable multicopy dynamic data possession in cloud computing systems," IEEE Transactions on Information Forensics and Security, vol. 10, no. 3,pp. 485–497, March 2015.

[2] B. Wang, B. Li, and H. Li, "Knox: privacy-preserving auditing for shared data with large groups in the cloud," in International Conference on Applied Cryptography and Network Security, 2012, pp. 507– 525.

[3] B. Wang, H. Li, and M. Li, "Privacy-preserving public auditing for shared cloud data supporting group dynamics," in 2013 IEEE International Conference on Communications (ICC), June 2013, pp. 1946–1950.

[4] C. Ellison and B. Schneier, "Ten risks of pki: What you're not being toldabout public key infrastructure," vol. 16, no. 1, 12 2000.

[5] H. Dewan and R. C. Hansdah, "A survey of cloud storage facilities," in 2011 IEEE World Congress on Services, July 2011, pp. 224–231.

[6] H. Jin, H. Jiang, and K. Zhou, "Dynamic and public auditing with fair arbitration for cloud data," IEEE Transactions on Cloud Computing, vol.13, no. 9, pp. 1–14, 2014.

[7] H. Wang, Q. Wu, B. Qin, and J. Domingo-Ferrer, "Identity-based remote data possession checking in public clouds," IET Information Security, vol. 8, no. 2, pp. 114–121, March 2014.

[8] H. Wang, D. He, and S. Tang, "Identity-based proxy oriented data uploading and remote data integrity checking in public cloud," IEEE Transactions on Information Forensics and Security, vol. 11, no. 6, pp. 1165–1176, June 2016.

[9] Yu, K. Ren, C. Wang, and V. Varadharajan, "Enabling cloud storage auditing with key-exposure resistance," IEEE Transactions on Information Forensics and Security, vol. 10, no. 6, pp. 1167–1179, 2015.

[10] Yu, K. Ren, and C. Wang, "Enabling cloud storage auditing with verifiable outsourcing of key updates," IEEE Transactions on Information

Forensics and Security, vol. 11, no. 6, pp. 1362–1375, June 2016.

[11] Yu and H. Wang, "Strong key-exposure resilient auditing for secure cloud storage," IEEE Transactions on Information Forensics and Security, vol. 12, no. 8, pp. 1931–1940, Aug 2017.

[12] K. Ren, C. Wang, and Q. Wang, "Security challenges for the public cloud," IEEE Internet Computing, vol. 16, no. 1, pp. 69–73, Jan 2012.