International Conference on Advances in Materials, Computing and Communication Technologies
In Association withInternational Journal of Scientific Research in Science and Technology
Volume 9 | Issue 1 | Print ISSN: 2395-6011 | Online ISSN: 2395-602X (www.ijsrst.com)

Enhanced Data Integrity Auditing And Sharing Scheme for Secure Cloud Storage

S. B. Autharsha[1], Mrs R Megiba Jasmine[2], Mrs S Shiny[2], Mrs S Vidhya[2]
[1]PG Scholar, Department of CSE, Ponjesly College of Engineering, Tamil Nadu, India
[2]Assistant Professor, Department of CSE, Ponjesly College of Engineering, Tamil Nadu, India

ABSTRACT

With cloud storage services, users will remotely store their data to the cloud and notice the info sharing with others. Remote information integrity auditing is planned to ensure the integrity of the info hold on within the cloud. In some common cloud storage systems adore the Electronic Health Records (EHRs) system, the cloud file may contain some sensitive information. The sensitive data mustn't be exposed to others once the cloud file is shared. Encrypting the total shared file will realize the sensitive data hiding, however can create this shared file unable to be utilized by others. the way to notice information sharing with sensitive data concealing in remote information integrity auditing still has not been explored up to now. so as to deal with this problem, we tend to propose a distant information integrity auditing theme that realizes information sharing with sensitive data concealing in this paper. during this scheme, a sanitizer is employed to sanitize the data blocks comparable to the sensitive data of the file and transforms these information blocks' signatures into valid ones for the sanitised file. These signatures are accustomed verify the integrity of the sanitised get in the part of integrity auditing. As a result, our theme makes the file hold on within the cloud able to be shared and utilized by others on the condition that the sensitive information is hidden, whereas the remote information integrity auditing is still able to be expeditiously executed. Meanwhile, the planned scheme is predicated on identity-based cryptography, that simplifies the sophisticated certificate management. the safety analysis and the performance analysis show that the planned theme is secure and economical

Index Terms: Cloud computing; Data integrity auditing; Data sharing; Sensitive information hiding.

I. INTRODUCTION

To achieve data sharing with sensitive information hiding in remote data integrity auditing, and propose a new concept called identity-based shared data integrity auditing with sensitive information hiding for secure cloud storage. In such a scheme, the sensitive information can be protected and the other information can be published. It makes the file stored in the cloud able to be shared and used by others on the condition that the sensitive information is protected, while the remote data integrity auditing is still able to be efficiently executed.

To design a practical identity-based shared data integrity auditing scheme with sensitive information hiding for secure cloud storage. A sanitizer is used to sanitize the data blocks corresponding to the sensitive information of the file. In our detailed scheme, firstly, the user blinds the data blocks corresponding to the personal sensitive information of the original file and generates the corresponding signatures, and then sends them to a sanitizer. The sanitizer sanitizes these blinded data blocks into a uniform format and also sanitizes the data blocks corresponding to the organization's sensitive information. It also transforms the corresponding signatures into valid ones for the sanitized file. This method not only realizes the remote data integrity auditing, but also supports the data sharing on the condition that sensitive information is protected in cloud storage. To the best of our knowledge ,this is the first scheme with the above functions. Besides, our scheme is based on identity-based cryptography, which simplifies the complex certificate management.

A. Existing System

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 project, propose a secure cloud storage system supporting privacy-preserving public auditing. Besides, with the prevalence of cloud computing, a foreseeable increase of auditing tasks from different users may be delegated to TPA. As the individual auditing of these growing tasks can be tedious and cumbersome, a natural demand is then how to enable the TPA to efficiently perform multiple auditing tasks in a batch manner. The techniques is a time consuming process and the Auditing system is Inefficient And less secure.

B. Cloud Architecture

The concept of cloud was introduced by Amazon. Amazon was within the business of selling goods and gift items. within the high season like Christmas, many people use to shop for gift items and other goods, therefore the load on their server increases to great extent. so as to run their business smoothly, they increased their server capability. But what about off season, the servers were idle and that they need to be kept running which successively consumes many power and at an equivalent time power was consumed in cooling them. therefore the Amazon decided to hire out their servers within the off season to others, such they will make money out of it. This is how the concept of cloud computing evolved as IaaS, SaaS,PaaS.

Software as a service(SaaS)

The SaaS service model offers the services as applications to the buyer ,using standardized interfaces. The services run on top of a cloud infrastructure, with cloud infrastructure being invisible for the buyer. The responsibility of the management the appliance,operating systems and underlying infrastructure lies with in the domain of cloud provider. the buyer can only control a number of the user-specific application configuration settings.

Platform as a service(PaaS)

The PaaS service model offers the services as operation and development platforms to the buyer . the buyer can use the platform to develop and run his own applications, supported by a underlying cloud based infrastructure."The consumer doesn't manage or control the underlying cloud- based infrastructure including network, servers, operating systems, or storage, but has control over the deployed applications and possibly application hosting environment configurations".

Infrastructure as a service(IaaS)

The IaaS service model is that the lowest service model within the technology stack, offering infrastructure resources as a service, like data storage, with processing power and increased network capacity. the buyer can the utilization IaaS based service offerings to deploy his own operating systems and applications, offering a wider sort of deployment possibilities for a consumer than the PaaS and SaaS models. It opens a replacement horizon for user for deployment of resources with greater flexibility.

a) Public cloud

Public clouds are owned and operated by third parties; they deliver superior economies of scale to customers, because the infrastructure costs are spread among a mixture of users, giving each individual client a beautiful low-cost, "Pay-as-you- go" model.

b) Private cloud

Private clouds are built exclusively for one enterprise. They aim to deal with concerns on data security and offer greater control, which is usually lacking during a public cloud.

c) Hybrid cloud

Hybrid Clouds combine both public and personal cloud models. With service providers can utilize 3rd party Cloud Providers during a full or partial manner thus increasing the pliability of computing. The Hybrid cloud environment is capable of providing on-demand, externally provisioned scale. the power to reinforce a personal cloud with the resources of a public cloud are often wont to manage any unexpected surges in workload.

II. LITERATURE SURVEY

Cong Wang et.al [1] presented a survey on cloud secure storage: 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.

Kamalam G K et.al [2] proposed a survey on cloud storage: the privacy preserving public auditing mechanism for shared data in the cloud. We utilize ring signatures to construct homomorphic authenticators, so the TPA is able to audit the integrity of shared data, yet cannot distinguish who is the signer on each block, which can achieve identity privacy. An interesting problem in future work is how to efficiently audit the integrity of shared data with dynamic groups while still preserving the identity of the signer on each block from the third party auditor. public auditing for such shared data while preserving identity privacy remains to be an open challenge. Here secure and effective methods

are needed to secure integrity and privacy data stored in cloud.

Lei Zhang et.al [3] proposed a survey on shared data on secure data storage. In this paper,an efficient public auditing solution that can preserve the identity privacy and the identity traceability for group members simultaneously. Specifically, we first design a new framework for data sharing in cloud, and formalize the definition of the public auditing scheme for shared cloud data supporting identity privacy and traceability. And then we construct such a scheme, in which a group manager is introduced to help members generate authenticators to protect the identity privacy and two lists are employed to record the members who perform the latest modification on each block to achieve the identity traceability. Besides, the scheme also achieves data privacy during authenticator generation by utilizing blind signature technique. Based on the proposed scheme, we further design an auditing system for practical scenarios.

Boyang et.al [4] projected the first privacy preserving public auditing mechanism for shared data in the cloud. We utilize ring signatures to construct homomorphic authenticators, so the TPA is able to audit the integrity of shared data, yet cannot distinguish who is the signer on each block, which can achieve identity privacy. To improve the efficiency of verification for multiple auditing tasks, we further extend our mechanism to support batch auditing. An interesting problem in our future work is how to efficiently audit the integrity of shared data with dynamic groups while still preserving the identity of the signer on each block from the third party auditor.

Min G et.al [5] proposed remote data integrity checking (RDIC) enables a data storage server, say a cloud server, to prove to a verifier that it is actually storing a data owner data honestly. To date, a number of RDIC protocols have been proposed in the literature, but most of the constructions suffer from the issue of a complex key management, that is, they rely on the expensive public key infrastructure (PKI), which might hinder the deployment of RDIC in practice. A new construction of identity-based (ID-based) RDIC protocol by making use of key-homomorphic cryptographic primitive to reduce the system complexity and the cost for establishing and managing the public key authentication framework in PKI-based RDIC schemes. Here formalize ID-based RDIC and its security model, including security against amalicious cloud server and zero knowledge privacy against a third party verifier.

Ravali et.al [6] proposed a key expore resistant in cloud computing. Most of the auditing protocols are based on the assumption that the client's secret key for auditing is secure. The security is not fully achieved, because of the low security parameters of the client. If the auditing protocol is not secured means the data of the client will exposed inevitably. In this paper a new mechanism of cloud auditing is implemented. And investigate to reduce the damage of the client key exposure in cloud storage auditing. Here the designing is built upon to overcome the week key auditing process. The auditing protocol is designed with the help of key exposure resilience.

Qian Wang et.al [7] suggested a survery on storage security in cloud computing. Third party auditor eliminates the involvement of the client through the auditing of whether his data stored in the cloud is indeed intact, which can be important in achieving economies of scale for Cloud Computing. The support for data dynamics via the most general forms of data operation, such as block modification, insertion and deletion, is also a significant step toward practicality, since services in Cloud Computing are not limited to archive or backup data only. While prior works on ensuring remote data integrity often lacks the support of either public auditability or dynamic data operations, this paper achieves both. We first identify

the difficulties and potential security problems of direct extensions with fully dynamic data updates from prior works and then show how to construct an elegant verification scheme for the seamless integration of these two salient features in our protocol design. In particular, to achieve efficient data dynamics, we improve the existing proof of storage models by manipulating the classic Merkle Hash Tree construction for block tag authentication.

Jinyuan Sun et.al [8] suggested across domain data sharing in distributed electronic health record system. Cautious design of delegation mechanism must be in place as a building block of cross-domain cooperation, since the cooperation inevitably involves exchanging and sharing relevant patient data that are considered highly private and confidential. Patients are unwilling to accept the EHR system unless their health data are guaranteed proper use and disclosure, which cannot be easily achieved without cross-domain authentication and fine-grained access control. In addition,revocation of the delegated rights should be possible at any time during the cooperation. Our EHR system further incorporates advanced mechanisms for fine-grained access control, and on-demand revocation, as enhancements to the basic access control offered by the delegation mechanism, and the basic revocation mechanism, respectively.

Jia Yu et.al [9] projected storng key exposure resilent auditing for secure storage. the malicious cloud might still forge valid authenticators later than the key-exposure time period if it obtains the current secret key of data owner. In this paper, we innovatively propose a paradigm named strong key-exposure resilient auditing for secure cloud storage, in which the security of cloud storage auditing not only earlier than but also later than the key exposure can be preserved. We formalize the definition and the security model of this new kind of cloud storage auditing and design a concrete scheme. In our proposed scheme, the key exposure in one time period doesn't affect the security of cloud storage auditing in other time periods.

Chaowen Guan et.al [10] proposed symmetric-key based proofs of retrievability supporting public verification. To explore indistinguishability obfuscation for building a Proof-of-Retrievability scheme that provides public verification while the encryption is based on symmetric key primitives. The resulting scheme offers light-weight storing and proving at the expense of longer verification. This could be useful in apations where outsourcing files is usually done by low-power client and verifications can be done by well equipped machines (e.g., a third party server). We also show that the proposed scheme can support dynamic updates. At last, for better assessing our proposed scheme, we give a performance analysis of our scheme and a comparison with several other existing schemes which demonstrates that our scheme achieves better performance on the data owner side and the server side.

III. PROPOSED DESIGN

In order to achieve data sharing with sensitive information hiding, consider making use of the idea in the sanitizable signature to sanitize the sensitive information of the file by introducing an authorized sanitizer. None the less, it is infeasible if this sanitizable signature is directly used in remote data integrity auditing. Firstly, this signature is constructed based on chameleon hashes. However, a lot of chameleon hashes exhibit the key exposure problem. To avoid this security problem, the signature used requires strongly unforgeable chameleon hashes, which will inevitable incur huge computation overhead . Secondly, the signature used does not support blockless verifiability. It means that the verifier has to download the entire data from the cloud to verify the integrity of data, which will incur huge communication overhead and excessive verification time in big data storage scenario. Thirdly,

the signature used is based on the PKI, which suffers from the complicated certificate management.

In order to address above problems, a new efficient signature algorithm in the phase of signature generation. The designed signature scheme supports blockless verifiability, which allows the verifier to check the integrity of data without downloading the entire data from the cloud. In addition, it is based on identity-based cryptography, which simplifies the complicated certificate management.

In proposed scheme, the PKG generates the private key for user according to his identity ID. The user can check the correctness of the received private key. When there is a desire for the user to upload data to the cloud, in order to preserve the personal sensitive information of the original file from the sanitizer, this user needs to use a blinding factor to blind the data blocks corresponding to the personal sensitive information of the original file. When necessary, the user can recover the original file from the blinded one by using this blinding factor. And then this user employs the designed signature algorithm to generate signatures for the blinded file. These signatures will be used to verify the integrity of this blinded file. In addition, the user generates a file tag, which is used to ensure the correctness of the file identifier name and some verification values. The user also computes a transformation value that is used to transform signatures for sanitizer. Finally, the user sends the blinded file, its corresponding signatures, and the file tag along with the transformation value to the sanitizer. When the above messages from user are valid, the sanitizer firstly sanitizes the blinded data blocks in to a uniform format and also sanitizes the data blocks corresponding to the organization's sensitive information to protect the privacy of organization, and then transforms their corresponding signatures into valid ones for sanitized file using transformation value. Finally, the sanitizer uploads the sanitized file and the corresponding signatures to the cloud. When the data integrity auditing task is performed, the cloud generates an auditing proof according to the challenge from the TPA. The TPA can verify the integrity of the sanitized file stored the cloud by checking whether this auditing proof is correct or not. The details will be described in the following subsection.

A. Advantages of proposed system

- Remote data integrity auditing is efficient
- Reduce verification time
- It achieves desirable security and efficiency
- Good performance.

IV. SYSTEM ARCHITECTURE

The system model involves five kinds of different entities: the cloud, the user, the sanitizer, the Private Key Generator (PKG) and the Third Party Auditor

(1) Cloud:

The cloud provides enormous data storage space to the user. Through the cloud storage service, users can upload their data to the cloud and share their data with others.

(2) User:

The user is a member of an organization, which has a large number of files to be stored in the cloud.

(3) Sanitizer:

The sanitizer is in charge of sanitizing the data blocks corresponding to the sensitive information (personal sensitive information and the organization's sensitive information) in the file, transforming these data blocks' signatures into valid ones for the sanitized file, and uploading the sanitized file and its corresponding signatures to the cloud.

(4) PKG:

The PKG is trusted by other entities. It is responsible for generating system public parameters and the private key for the user according to his identity ID.

(5) TPA:

The TPA is a public verifier. It is in charge of verifying the integrity of the data stored in the cloud on behalf of users

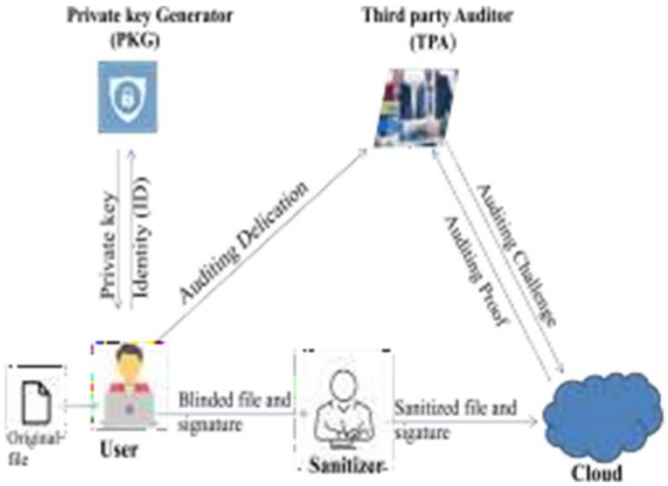

Fig 1.1 System Architecture

The user firstly blinds the data blocks corresponding to the personal sensitive information of the file, and generates the corresponding signatures. These signatures are used to guarantee the authenticity of the file and verify the integrity of the file. Then the user sends this blinded file and its corresponding signatures to the sanitizer. After receiving the message from the user, the sanitizer sanitizes these blinded data blocks and the data blocks corresponding to the organization's sensitive information, and then transforms the signatures of sanitized data blocks into valid ones for the sanitized file. Finally, the sanitizer sends this sanitized file and its corresponding signatures to the cloud. These signatures are used to verify the integrity of the sanitized file in the phase of integrity auditing

When the TPA wants to verify the integrity of the sanitized file stored in the cloud, he sends an auditing challenge to the cloud. And then, the cloud responds to the TPA with an auditing proof of data possession. Finally, the TPA verifies the integrity of the sanitized file by checking whether this auditing proof is correct or not.

The system model involves five kinds of different entities: the cloud, the user, the sanitizer, the Private Key Generator (PKG) and the Third Party Auditor (TPA)

- Cloud Storage
- Setup Phase
- Extract Phase
- Sanitizer Phase
- ProofGen Phase
- ProofVerify Phase

A. Cloud storage

The cloud provides enormous data storage space to the user. Through the cloud storage service, users can upload their data to the cloud and share their data with others

B. Setup

The PKG chooses two multiplicative cyclic groups G1 and G2 of prime order p, a generator g of G1, a bilinear map e : G1 × G1 → G2 and a pseudorandom function f : Z ∗ p × Z ∗ p → Z ∗ p . The PKG randomly chooses an element x ∈ Z ∗ p , elements μ 0 , μ1, μ2, ..., μl , u, g2 ∈ G1 and a cryptographic hash function H : {0, 1} ∗ → G1. The PKG computes the public value g1=g x and the master secret key msk = g2 x . The PKG publishes system parameters pp = (G1, G2, p, e, g, μ0 , μ1, μ2,
..., μl , u, g1, g2, H, f). The PKG computes the public value g1=g x and the master secret key msk = g2 x .

C. Extract

Extract is an extraction algorithm run by the PKG. It takes as input the system public parameters pp, the master secret key msk, and a user's identity ID. It outputs the user's private key skID. The user can verify the correctness of skID and accept it as his private key only if it passes the verification. After receiving the user's identity ID = (ID1, ID2, ..., IDl) ∈ {0, 1}, the PKG randomly picks a value rID ∈ Z∗p and computes

$$(sk'_{ID}, sk''_{ID}) = (g_2{}^{x} \cdot (\mu' \prod_{j=1}^{l} \mu_j{}^{ID_j})^{r_{ID}}, g^{r_{ID}})$$

as the private key of the user ID. The PKG sends it to the user ID. The user ID verifies the correctness of the received private key skID by checking whether the following equation holds or not.

$$e(sk'_{ID}, g) = (g_1, g_2) \cdot e(\mu' \prod_{j=1}^{l} \mu_j{}^{ID_j}, sk''_{ID})$$

D. Sanitizer

The sanitizer is in charge of sanitizing the data blocks corresponding to the sensitive information (personal sensitive information and the organization's sensitive information) in the file, transforming these data blocks' signatures into valid ones for the sanitized file, and uploading the sanitized file and its corresponding signatures to the cloud. Sanitizer is a sensitive information sanitization algorithm run by the sanitizer. It takes as input the blinded file and its signature set. It outputs the sanitized file and its corresponding signature set. The sanitizer checks the validity of the file tag τ by verifying whether SSigssk(τ0) is a valid signature

E. ProofGen

The TPA verifies the validity of the file tag τ . The TPA will not execute auditing task if the file tag τ is invalid; otherwise, the TPA parses τ0 to obtain file identifier name name and verification values g rID and g r , and then generates an auditing challenge chal as follows: Randomly picks a set I with c elements, where I [1, n]. Generates a random value vi Z p for each i I. Sends the auditing challenge chal = {i, vi}iI to the cloud. After receiving an auditing challenge from the TPA, the cloud generates a proof of data possession as follows: Computes a linear combination of data blocks λ = P iI m0 i vi . Calculates an aggregated signature Q σ = iI σ 0 i vi . Outputs an auditing proof P = {λ, σ} to the TPA.

F. ProofVerify

The TPA verifies the correctness of auditing proof as follows:

$$e(\sigma, g) = e(g_1, g_2)^{\sum_{i \in I} v_i} \cdot e(\mu' \prod_{j=1}^{l} \mu_j{}^{ID_j}, g^{r_{ID}})^{\sum_{i \in I} v_i} \cdot e(\prod_{i \in I} H(name||i)^{v_i} \cdot u^{\lambda}, g^{r}).$$

V. RESULT AND DISCUSSION

In this section, evaluate the efficiency of our protocols. Since the protocol is the only privacy preserving auditing protocol which enables data dynamics. In particular, perform several simulations to evaluate the efficiency of our protocols. The cryptographic algorithms are implemented using the pairing-based cryptography (PBC) library.

A. Performance Graph

In this section, evaluate the efficiency of our protocols. Since the protocol is the only privacy preserving auditing protocol which enables data dynamics, compare our protocols with it. In particular, perform several simulations to evaluate the efficiency of our protocols. The cryptographic algorithms are

implemented using the pairing based cryptography (PBC) library.

The efficiency of the whole protocol is also dominated by the Audit phase. In this section, we evaluate the performance of our Audit phase. As mentioned before, the TPA only needs to select c file blocks to be checked rather than all the file blocks. In order to achieve the high assurance, the value of c is usually selected to be 300 and 460 for the probability of 95% and 99% respectively.

The above analysis indicates that our protocols are also efficient in the Audit phase. Here, do not increase the computational cost for the CSP to respond the challenge from the TPA. Meanwhile, the TPA needs much less time to verify the response.

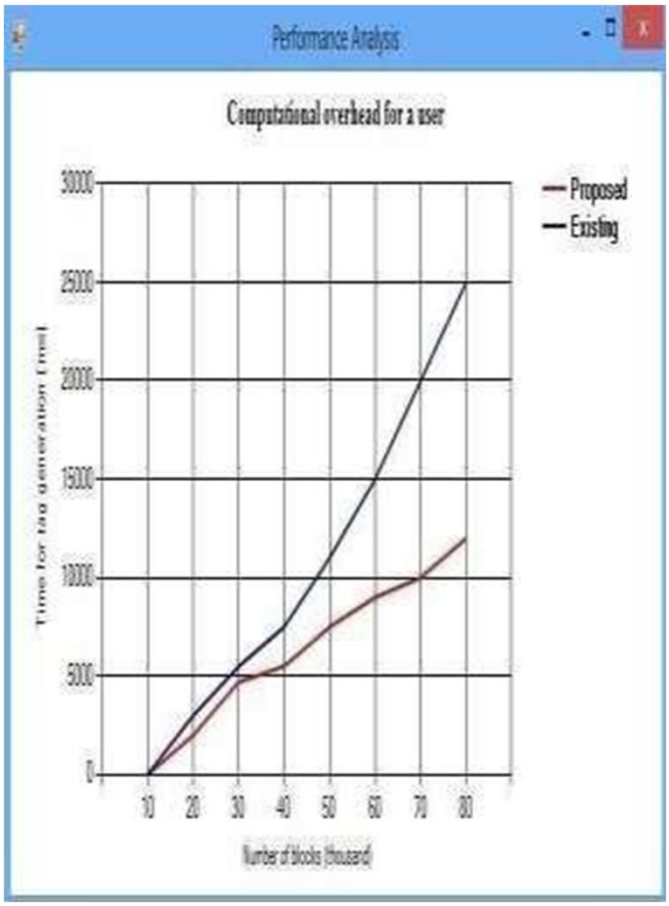

Fig.1.2 Home page

Fig.1.3 Cloud storage

Fig.1.4 Sanitizer

Fig1.5 Proof generation and verification

Fig 1.6 Data Retrieval

VI. CONCLUSION

In this project, proposed an identity-based data integrity auditing scheme for secure cloud storage, which supports data sharing with sensitive information hiding. In this scheme, the file stored in the cloud can be shared and used by others on the condition that the sensitive information of the file is protected. Besides, the remote data integrity auditing is still able to be efficiently executed. The security proof and the experimental analysis demonstrate that the proposed scheme achieves desirable security and efficiency.

VII. REFERENCES

[1]. 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.

[2]. C. Wang, S. S. M. Chow, Q. Wang, K. Ren, and W. Lou, "Privacy-preserving public auditing for secure cloud storage," IEEE Transactions on Computers, vol. 62, no. 2, pp. 362–375, 2013.

[3]. G. Yang, J. Yu, W. Shen, Q. Su, Z. Fu, and R. Hao, "Enabling public auditing for shared data in cloud storage supporting identity privacy and traceability," J. Syst. Softw., vol. 113, no. C, pp. 130–139, Mar. 2016.

[4]. B. Wang, B. Li, and H. Li, "Oruta: Privacy-preserving public auditing for shared data in the cloud," in 2012 IEEE Fifth International Conference on Cloud Computing, June 2012, pp. 295–302.

[5]. Y. Yu, M. H. Au, G. Ateniese, X. Huang, W. Susilo, Y. Dai, and G. Min, "Identity-based remote data integrity checking with perfect data privacy preserving for cloud storage," IEEE Transactions on Information Forensics and Security, vol. 12, no. 4, pp. 767–778, April 2017.

[6]. J. 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.

[7]. Q. Wang, C. Wang, K. Ren, W. Lou, and J. Li, "Enabling public auditability and data dynamics for storage security in cloud computing," IEEE Transactions on Parallel and Distributed Systems, vol. 22, no. 5, pp. 847–859, May 2011.

[8]. J. Sun and Y. Fang, "Cross-domain data sharing in distributed electronic health record systems," IEEE Transactions on Parallel and Distributed Systems, vol. 21, no. 6, pp. 754–764, June 2010.

[9]. J. 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.

[10]. C. Guan, K. Ren, F. Zhang, F. Kerschbaum, and J. Yu, "Symmetric-key based proofs of retrievability supporting public verification," in Computer Security– ESORICS 2015. Cham: Springer International Publishing, 2015, pp. 203–223.

[11]. Vinu Sundararaj, Selvi uthukumar, & Kumar,R.S. (2018). An optimal cluster formation based energy efficient dynamic scheduling hybrid MAC protocol

[12]. Vilaplana, J., Solsona, F., Teixido, I., & ritus,J. (2014). A queuing theory model for cloud computing. Journal pf Supercomputing, 69(1), 492-507.

[13]. Xie, J., Dan, L., Yin, L., & Sun,Z . (2015). An energy optimal scedulimg for cllabrative execution in mobile computing. In 2015 international conference and workshop on cloud computing and communication(pp.1-6). IEEE.

[14]. Zhang, W., Wen, Y., & Wu, D.(2013). Energy-efficient scheduling policy for collaborative execution in mobile cloud computing. Iin 2013 Proceedings IEEE(pp. 190-194).

[15]. Tayal, S. (2011). Tasks scheduling optimization for the cloud computing systems. IJAEST-International Journal of advanced Engineering Sciences and Technologies, 1(5), 111-115.

[16]. Xu, B., Peng, Z., Xiao, F., & Yu, P(2015). Dynamic deploymaent of virtual machines in cloud computing using multi-objective optimization. Soft computing, 19(8), 2265-2273.

[17]. Altamimi, M., Abdrabou, A., Naik, K., & Nayak, A. (2015). Energy cost models of smartphones for task. IEEE Transactions Emerging topics in Computing, 3(3), 384-398.

[18]. Y. Zhang, J. Yu, R. Hao, C. Wang, and K. Ren, "Enabling efficient user revocation in identity-based cloud storage auditing for shared big data," IEEE Transactions on Dependable and Secure Computing,2019 .

[19]. A. Fu, S. Yu, Y. Zhang, H. Wang, and C. Huang, "Npp: A new privacy-aware public auditing scheme for cloud data sharing with group users," IEEE Transactions on Big Data, 2017.

[20]. Y. Li, Y. Yu, G. Min, W. Susilo, J. Ni, and K.K. R. Choo, "Fuzzy identity-based data integrity auditing for reliable cloud storage systems," IEEE Transactions on Dependable and Secure Computing, 2017.