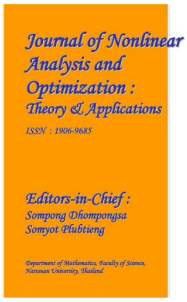
Journal of **N**onlinear **A**nalysis and **O**ptimization
Vol. 15, Issue. 1, No.15 : 2024
ISSN : **1906-9685**



IMPROVING SECURITY IN CLOUD STORAGE:AUDITING BY IDENTITY HIDDEN DATA AND SECURE SHARING

Mrs. M.Rama , Assistant professor CSE, Vaagdevi College of Engineering(Autonomous),India
Koduri Sreeya , UG Student, CSE, College of Engineering(Autonomous),India
Gogikar Shushma , UG Student, CSE, College of Engineering(Autonomous),India
Manda Teja, UG Student, CSE, Vaagdevi College of Engineering(Autonomous),India
Naliganti sidhartha, UG Student, CSE, Vaagdevi College of Engineering(Autonomous),India

ABSTRACT

With cloud storage services, users can remotely store their data to the cloud and realize the data sharing with others. Remote data integrity auditing is proposed to guarantee the integrity of the data stored in the cloud. In some common cloud storage systems such as the Electronic Health Records (EHRs) system, the cloud file might contain some sensitive information. The sensitive information should not be exposed to others when the cloud file is shared. Encrypting the whole shared file can realize the sensitive information hiding, but will make this shared file unable to be used by others. How to realize data sharing with sensitive information hiding in remote data integrity auditing still has not been explored up to now. In order to address this problem, we propose a remote data integrity auditing scheme that realizes data sharing with sensitive information hiding in this paper. In this scheme, a sanitizer is used to sanitize the data blocks corresponding to the sensitive information of the file and transforms these data blocks' signatures into valid ones for the sanitized file. These signatures are used to verify the integrity of the sanitized file in the phase of integrity auditing. As a result, our scheme makes the file stored in the cloud able to be shared and used by others on the condition that the sensitive information is hidden, while the remote data integrity auditing is still able to be efficiently executed. Meanwhile, the proposed scheme is based on identity-based cryptography, which simplifies the complicated certificate management. The

security analysis and the performance evaluation show that the proposed scheme is secure and efficient.

1. INTRODUCTION

MOTIVATION

However, the data stored in the cloud might be corrupted or lost due to the inevitable software bugs, hardware faults and human errors in the cloud [1]. In order to verify whether the data is stored correctly in the cloud, many remote data integrity auditing schemes have been proposed

PROBLEM DEFINITION

In order to address above problems, we design 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.

OBJECTIVE OF PROJECT

In our 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[2], the sanitizer firstly sanitizes the blinded data blocks into a uniform format and also sanitizes the data blocks corresponding to the organization's sensitive information to protect the privacy of organization

2. LITERATURE SURVEY

Cloud computing represents today's most exciting computing paradigm shift in information technology. However, security and privacy are perceived as primary obstacles to its wide adoption. Here, the authors outline several critical security challenges and motivate further investigation of security solutions for a trustworthy public cloud environment.

We introduce a model for provable data possession (PDP) that allows a client that has stored data at an untrusted server to verify that the server possesses the original data without retrieving it. The model generates probabilistic proofs of possession by sampling random sets of blocks from the server, which drastically reduces I/O costs. The client maintains a constant amount of metadata to verify the proof. The challenge/response protocol transmits a small, constant amount of data, which minimizes network communication. Thus, the PDP [3] model for remote data checking supports large data sets in widely-distributed storage systems. We present two provably-secure PDP schemes that are more efficient than previous solutions, even when compared with schemes that achieve weaker guarantees. In particular, the overhead at the server is low (or even constant), as opposed to linear in the size of the data. Experiments using our implementation verify the practicality of PDP and reveal that the performance of PDP is bounded by disk I/O and not by cryptographic computation.

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 toward 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. Our preliminary experiment conducted on Amazon EC2 instance further demonstrates the fast performance of the design.

Cloud computing poses many challenges on integrity and privacy of users' data though it brings an easy, cost-effective and reliable way of data management. Hence, secure and efficient methods are needed to ensure integrity and privacy of data stored at the cloud. Wang et al. proposed a privacy-preserving public auditing protocol in 2010 but it is seriously insecure. Their scheme is vulnerable to attacks from malicious cloud server and outside attackers regarding to storage correctness. So they proposed a scheme in 2011 with an improved security guarantee but it is not efficient. Thus, in this paper, we proposed a scheme which is secure and with better efficiency. It is a public auditing scheme with third party auditor (TPA) [4] , who performs data auditing on behalf of user(s). With detail security analysis, our scheme is proved secure in the random oracle model and our performance analysis shows the scheme is efficient.

Proofs-of-Retrievability enables a client to store his data on a cloud server so that he executes an efficient auditing protocol to check that the server possesses all of his data in the future. During an audit, the server must maintain full knowledge of the client's data to pass, even though only a few blocks of the data need to be accessed. Since the first work by Juels and Kaliski, many PoR schemes have been proposed and some of them can support dynamic updates. However, all the existing works that achieve public verifiability are built upon traditional public-key cryptosystems which imposes a relatively high computational burden on low-power clients e.g.,i?źmobile devices. In this work we 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.,i?źa 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.

3. EXISTING SYSTEM:

With the explosive growth of data, it is a heavy burden for users to store the sheer amount of data locally. Therefore, more and more organizations and individuals would like to store their data in the cloud. However, the data stored in the cloud might be corrupted or lost due to the inevitable software bugs, hardware faults and human errors in the cloud [1]. In order to verify whether the data is stored correctly in the cloud, many remote data integrity auditing schemes have been proposed.

3.1 Disadvantages

1. Scalability Issues: The existing system may struggle to handle large-scale data storage and sharing requirements. As more users and data are added to the system, the computational and storage overhead for identity-based integrity auditing and sensitive information hiding can become a bottleneck, leading to performance degradation.

2. Complexity and Overhead: Implementing identity-based integrity auditing and sensitive information hiding typically involves complex cryptographic [5] operations and additional metadata management. This can introduce overhead in terms of processing power and storage, potentially slowing down data access and sharing operations.

3. Key Management Challenges: Managing cryptographic keys for identity-based integrity auditing and data sharing can be challenging. If not handled properly, it can lead to key exposure risks or difficulties in key revocation and rotation. The existing system may not provide an efficient and user-friendly key management mechanism, making it less practical for users.

4. PROPOSED SYSTEM:

Remote data integrity auditing is proposed to guarantee the integrity of the data stored in the cloud. In some common cloud storage systems such as the electronic health records system, the cloud file might contain some sensitive information. The sensitive information should not be exposed to others when the cloud file is shared. Encrypting [6] the whole shared file can realize the sensitive information hiding, but will make this shared file unable to be used by others. How to realize data sharing with sensitive information hiding in remote data integrity auditing still has not been explored up to now. In order to address this problem, we propose a remote data integrity auditing scheme that realizes data sharing with sensitive information hiding in this paper. In this scheme, a sanitizer is used to sanitize the data blocks corresponding to the sensitive information of the file and transforms these data blocks' [9] signatures into valid ones for the sanitized file. These signatures are used to verify the integrity of the sanitized file in the phase of integrity auditing. As a result, our scheme makes the file stored in the cloud able to be shared and used by others on the condition that the sensitive information is hidden, while the remote data integrity auditing is still able to be efficiently executed

4.1 ADVANTAGES

1. Improved Usability: The proposed system may offer a more user-friendly interface and seamless integration with cloud storage services. This could make it easier for users to manage and share their data securely without requiring extensive knowledge of complex cryptographic operations.

2. Enhanced Scalability: By optimizing resource usage and employing efficient cryptographic techniques, the proposed system can potentially offer better scalability. It may handle larger volumes of data and more users while maintaining acceptable performance levels, addressing one of the disadvantages of the existing system.

3. Robust Security: The proposed system could enhance security by implementing advanced encryption and access control mechanisms. It may also include robust key management solutions, reducing the risk of key exposure and ensuring the confidentiality and integrity of sensitive data.

5. SYSTEM ARCHITECTURE:

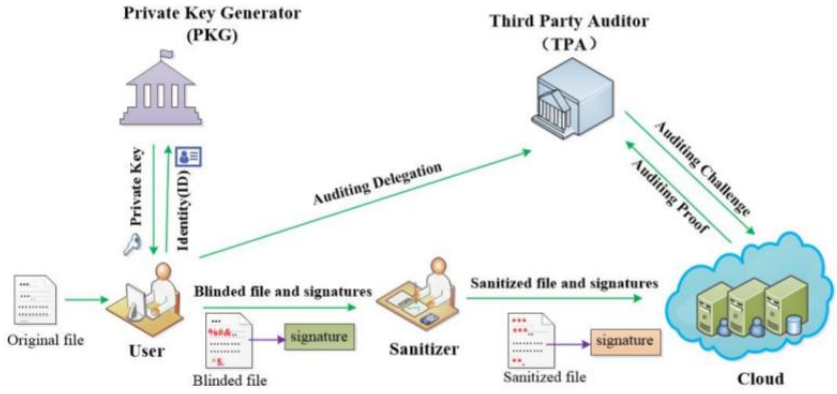


Fig 2.1 System architecture

6. IMPLEMENTATION

1.USER

2.PKG

3.SANITIZER

4.TPA

5.CLOUD

System design is transition from a user oriented document to programmers or data base personnel. The design is a solution, how to approach to the creation of a new system. This is composed of several steps. It provides the understanding and procedural details necessary for implementing the system recommended in the feasibility [7] study. Designing goes through logical and physical stages of development, logical design reviews the present physical system, prepare input and output specification, details of implementation plan and prepare a logical design walkthrough.

The database tables are designed by analyzing functions involved in the system and format of the fields is also designed. The fields in the database tables should define their role in the system. The unnecessary fields should be avoided [7] because it affects the storage areas of the system. Then in the input and output screen design, the design should be made user friendly. The menu should be precise and compact.

7.EXPECTED OUTCOMES

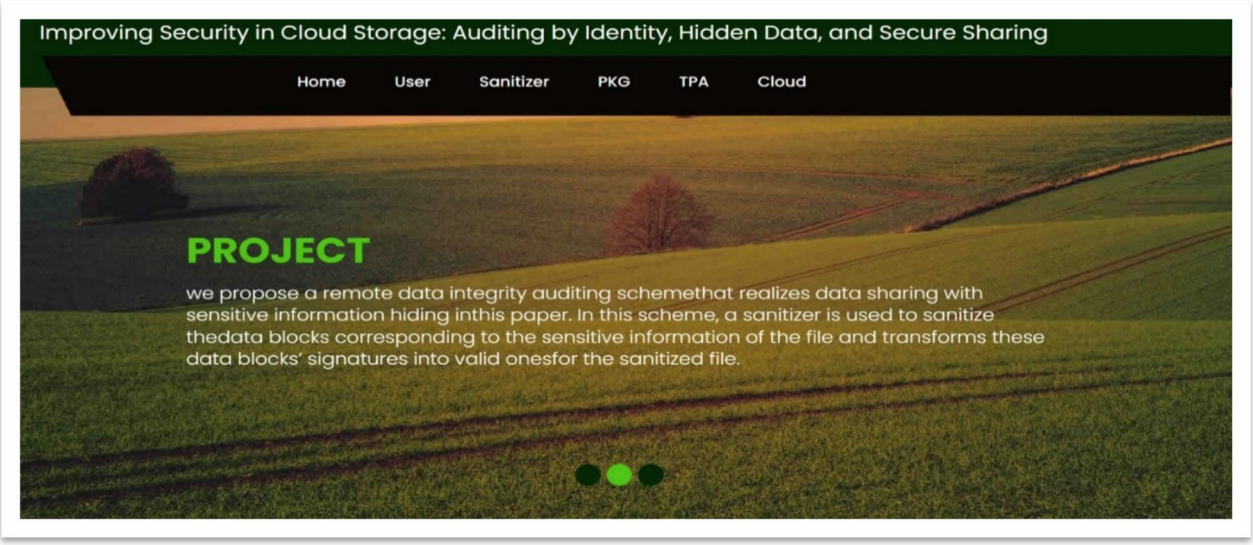


Fig 7.1 Home Page

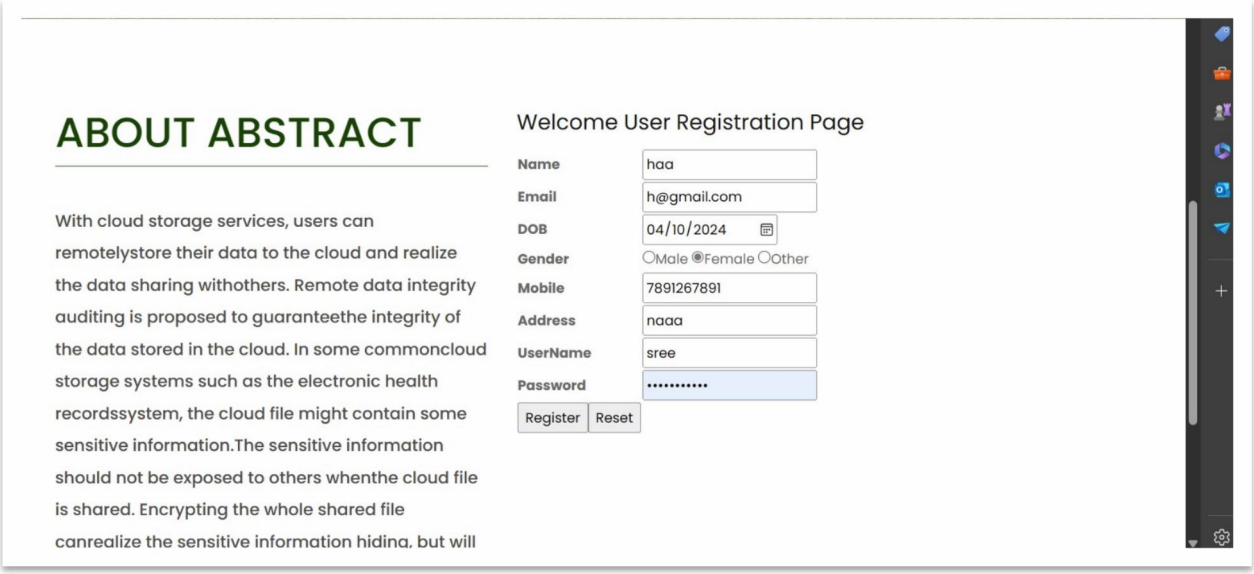


Fig 7.2 Apps Registration Page

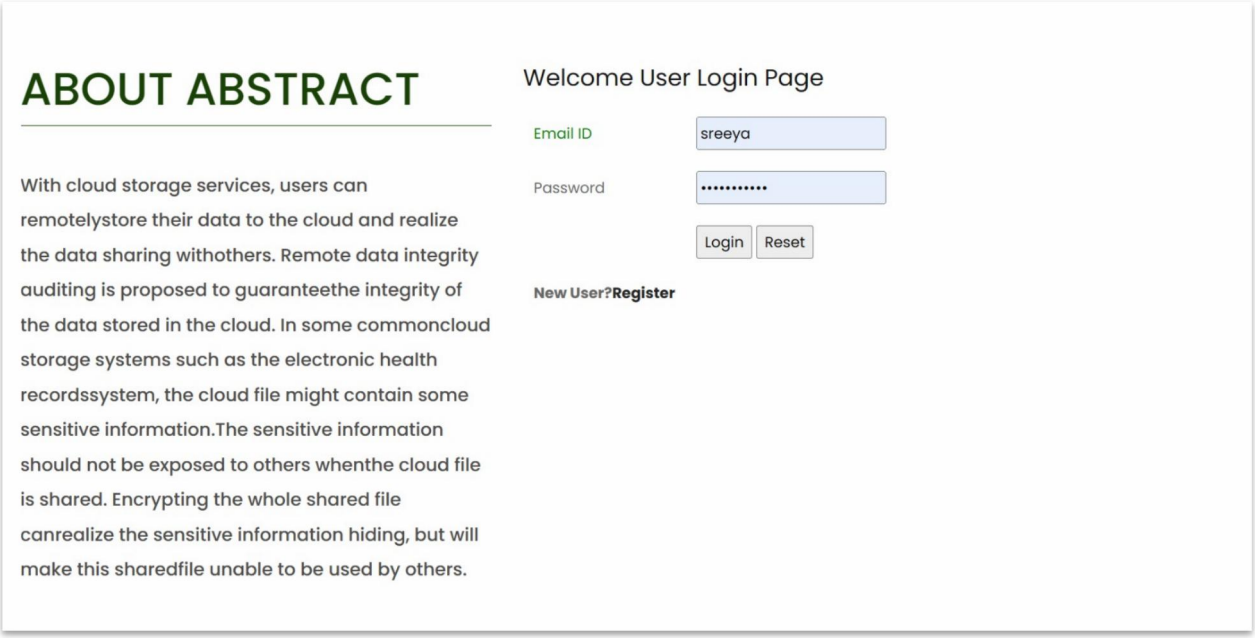


Fig 7.3 Login page

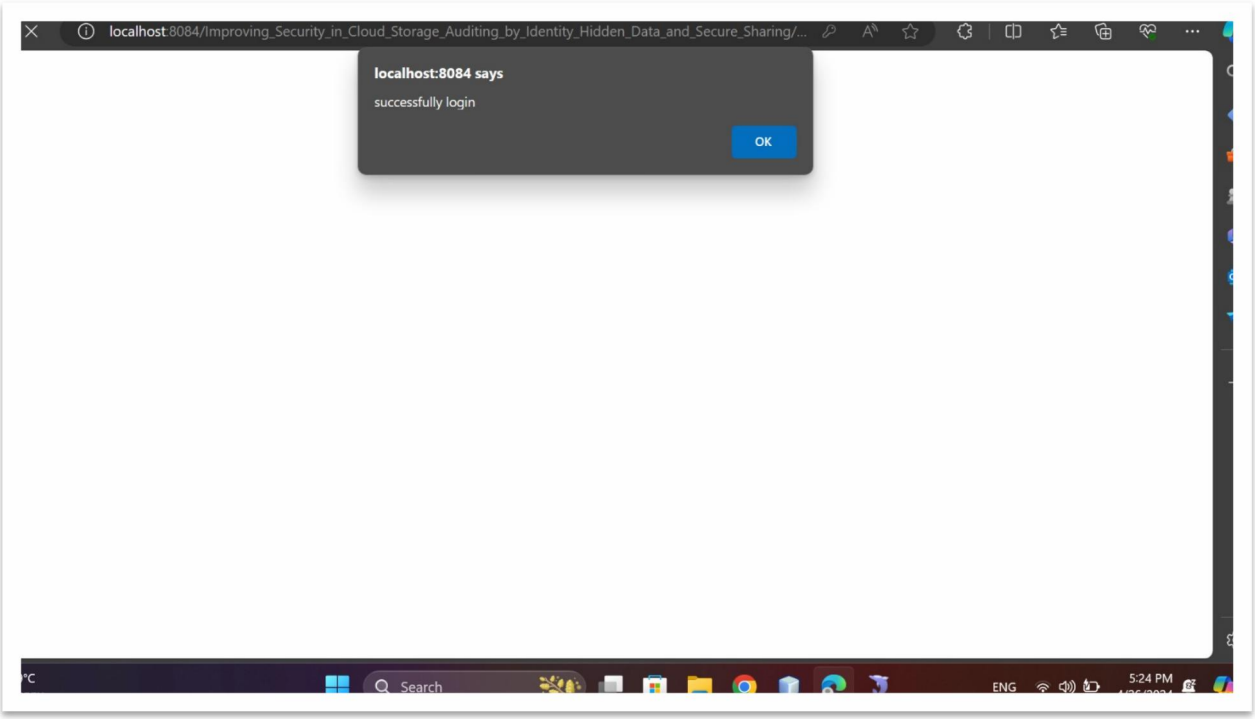


Fig 7.4 Login successful page

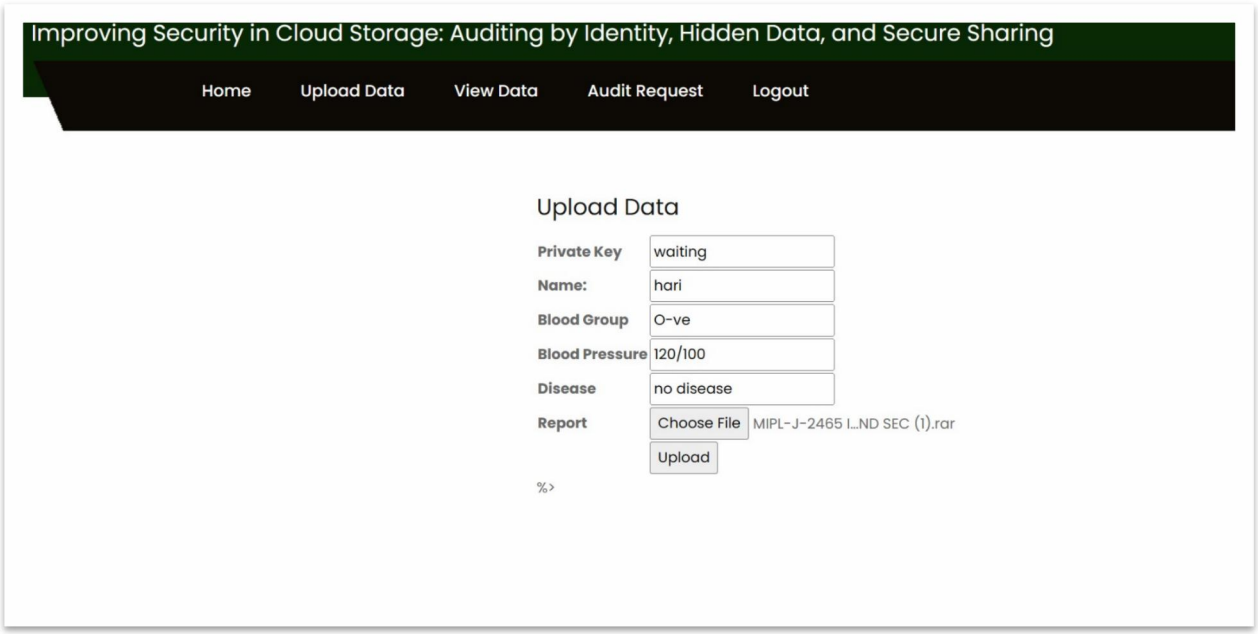


Fig 7.4 Upload data page

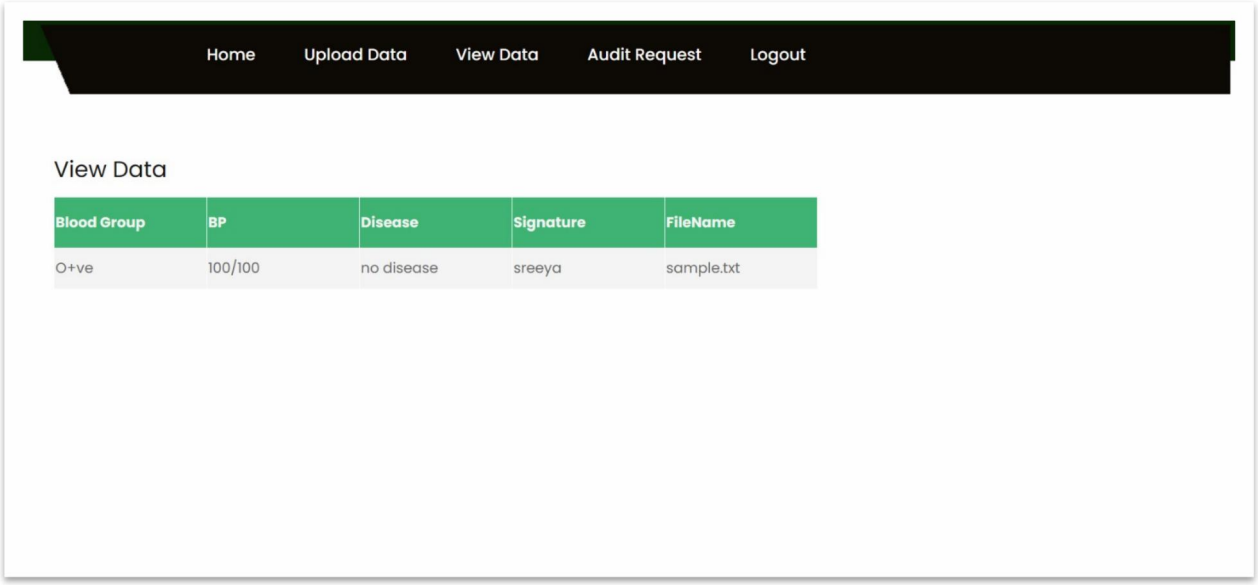


Fig 7.5 View data page

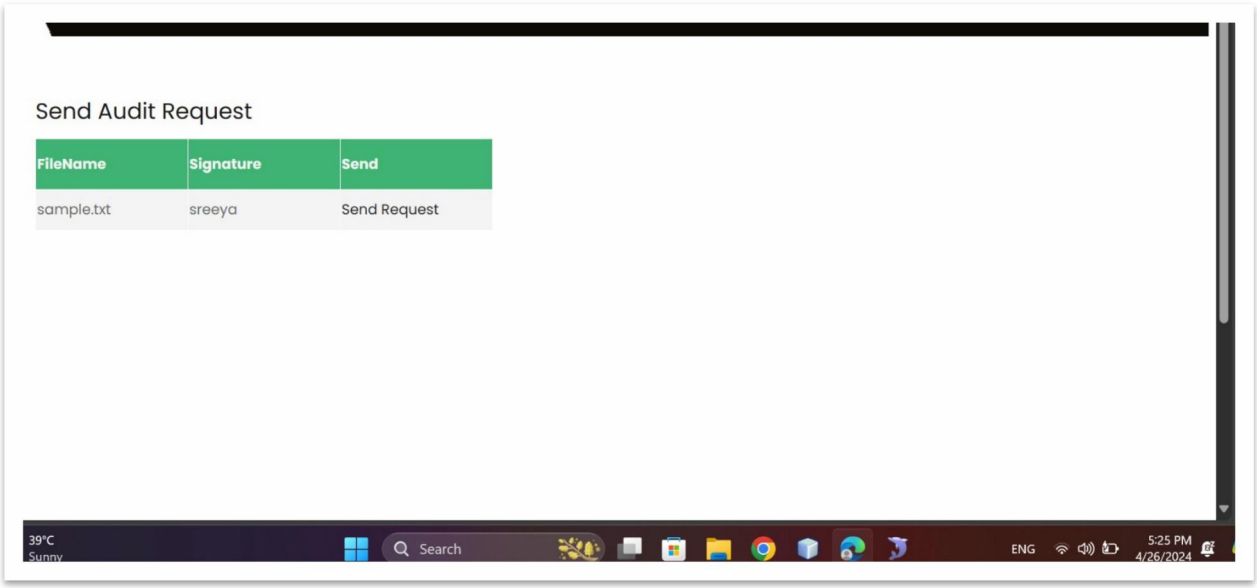


Fig 7.6 Send audit request

8. CONCLUSION

In this paper, we proposed an identity-based data integrity auditing scheme for secure cloud storage, which supports data sharing with sensitive information hiding. In our 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.

8.1 FUTURE SCOPE

The future scope for improving security in cloud storage auditing through identity-hidden data secure sharing lies in advancing encryption techniques, such as homomorphic encryption, which allows computations to be performed on encrypted data without decrypting it. Additionally, integrating artificial intelligence and machine learning algorithms for anomaly detection and behavior analysis can enhance security by identifying suspicious activities in real-time. Furthermore, exploring decentralized and blockchain-based solutions may provide greater transparency and resilience against attacks.

9. 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] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song, “Provable data possession at untrusted stores,” in Proceedings of the 14th ACM Conference on Computer and Communications Security, ser. CCS ’07, 2007, pp. 598–609.

[3] A. Juels and B. S. Kaliski, “Pors: Proofs of retrievability for large files,” in Proceedings of the 14th ACM Conference on Computer and Communications Security, ser. CCS ’07, 2007, pp. 584–597.

[4] H. Shacham and B. Waters, “Compact proofs of retrievability,” J. Cryptology, vol. 26, no. 3, pp. 442–483, Jul. 2013.

[5] 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.

[6] S. G. Worku, C. Xu, J. Zhao, and X. He, “Secure and efficient privacy-preserving public auditing scheme for cloud storage,” Comput. Electr. Eng., vol. 40, no. 5, pp. 1703–1713, Jul. 2014.

[7] 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.

[8[W. Shen, J. Yu, H. Xia, H. Zhang, X. Lu, and R. Hao, “Light-weight and privacy-preserving secure cloud auditing scheme for group users via the third party medium,” Journal of Network and Computer Applications, vol. 82, pp. 56–64, 2017.

[9] 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