International Journal of Innovative Research in Computerand Communication Engineering

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | Impact Factor: 7.488 |

||Volume 8, Issue 6, June 2020||

A Secure Data Auditing Scheme to Integrate without any Private Key Storage over Cloud Storage

T. Sujilatha[1], V. Kasthuraiah[2]

Associate Professor, Head of the Dept, Dept. of Computer Science and Engineering, Gokula Krishna College of Engineering, Sullurpet, India[1]

PG Scholar, Dept. of Computer Science and Engineering, Gokula Krishna College of Engineering, Sullurpet, India[2]

Abstract—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, smartcard) 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.

KEYWORDS*:* Cloud Storage, Data Integrity Auditing, Data Security, Biometric Data.

I. INTRODUCTION

Cloud storage can provide powerful and on-demand data storage services for users [1]. 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 [2].

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 [3–5], the privacy protection of data and user identities [6–8], key exposure resilience [9], 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.

A feasible method is to use biometric data, such as fingerprint and iris scan, as the private key. Biometric data, as a part of human body, can uniquely link the individual and the private key. Unfortunately, biometric data is measured with inevitable noise each time and cannot be reproduced precisely since some factors can affect the change of biometric data. For example, the finger of each person will generate a different fingerprint image every time due to pressure,

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | Impact Factor: 7.488 |

||Volume 8, Issue 6, June 2020||

moisture, presentation angle, dirt, different sensors, and so on. Therefore, the biometric data cannot be used directly as the private key to generate authenticators in data integrity auditing.

II. RELATED WORK

A teniese et al. firstly proposed the notion of Provable Data Possession (PDP). They employed the random sample technique and homomorphic linear authenticators to design a PDP scheme, which allows an auditor to verify the integrity of cloud data without downloading the whole data from the cloud. Juels and Kaliski proposed the concept of Proof of Retrievability (PoR).In the proposed scheme, the error correcting codes and the spot-checking technique are utilized to ensure the retrievability and the integrity of the data stored in the cloud. Shacham and Waters constructed two PoR schemes with private verifiability and public verifiability by using pseudorandom function and BLS signature. To support user-interactions, including data modification, insertion and deletion, Zhu et al. constructed a dynamic data integrity auditing scheme by exploiting the index hash tables. Sookhak et al. also considered the problem of data dynamics in data integrity auditing and designed a data integrity auditing scheme supporting data dynamic operations based on the Divide and Conquer Table. In public data integrity auditing, the TPA might derive the contents of user's data by challenging the same data blocks multiple times. To protect the data privacy, Wang et al. exploited the random masking technique to construct the first public data integrity auditing scheme supporting privacy preserving. Li et al. proposed a data integrity auditing scheme which preserves data privacy from the TPA. Yu et al. proposed a cloud storage auditing scheme with perfect data privacy preserving by making use of zero-knowledge proof.

To relieve the user's computation burden of authenticator generation, Guan et al. constructed a data integrity auditing scheme using indistinguish ability obfuscation technique, which reduces the overhead for generating data authenticators. Li et al. proposed a data integrity auditing scheme which contains a cloudstorage server and a cloud audit server.In this scheme, the cloud audit server helps user to generate data authenticators before uploading data to the cloud storage server. Shen et al. designed a light-weight data integrity auditing scheme, which introduced a Third-Party Medium to generate authenticators and verify data integrity on behalf of users.

The data sharing is used widely in cloud storage scenarios. To protect the identity privacy of user, Wang et al. proposed a shared data integrity auditing scheme based on the ring signature. Yang et al.designed a remote data integrity auditing scheme for shared data, which supports both the identity privacy and the identity traceability. By using thehomomorphic verifiable group signature, Fu et al. proposed a privacy-aware remote data integrity auditing scheme for shared data. In order to achieve efficient user revocation, Wang et al.designed a shared data integrity auditing scheme supporting user revocation by making use of the proxy re-signature. Based on the identity-based setting, Zhang et al. constructed a cloud storage auditing scheme for shared data supporting real efficient user revocation. To realize the data sharing with sensitive information hiding, Shen et al. designed an identity-based cloud storage auditing scheme for shared data.

Other aspects, such as eliminating certificate management and key exposure resilience [9–11] in data integrity auditing have also been studied. However, all of existing remote data integrity auditing schemes do not take the problem of private key storage into account. In this paper, we explore how to achieve data integrity auditing scheme without private key storage for secure cloud storage.

III. PROPOSED WORK

System Model

As illustrated in Fig. 1, the system model involves three types of entities: the user, the cloud, and the TPA. The cloud provides enormous data storage space to the user. The user has a large number of files to be uploaded to the cloud. The TPA is a public verifier who is delegated by the user to verify the integrity of the data stored in the cloud.

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | Impact Factor: 7.488 |

||Volume 8, Issue 6, June 2020||

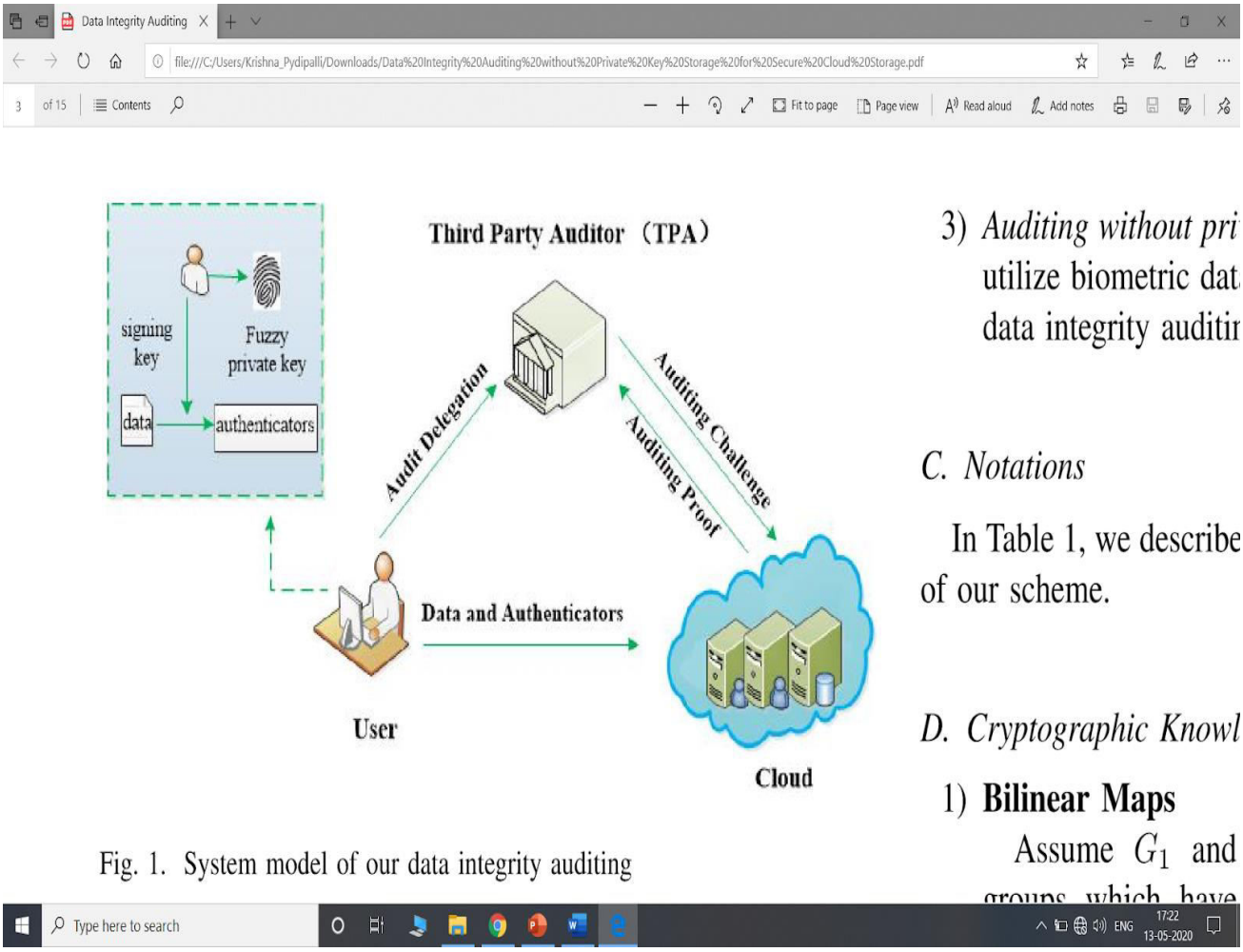


Fig. 1. System model of our data integrity auditing

Fig. 1. System model of our data integrity auditing

This systemmodel consists and implements the following modules:

Data Owner: In this module, Data owner has to register to cloud and logs in, Encrypts and uploads a file to cloud server and also performs the following operations such as Upload File with Blocks, View All Upload File with Blocks, Perform Data Integrity Auditing, View Transactions.

Cloud Server: In this module the cloud will authorize both the owner and the user and also performs the following operations such as View and Authorize Users, View and Authorize Owners, View All File's Blocks, View All Transactions, View All Attackers, View Time Delay Results, View Throughput Results.

TPA: In this module, the TPA performs the following operations such as View Metadata Details, View All Transactions, View All Attackers.

Data User: In this module, the user has to register to cloud and log in and performs the following operations such as Search Data, Download Data.

Design Goals

To enable data integrity auditing without private key storage for secure cloud storage, our scheme should achieve the following goals:

1) Auditing correctness: to ensure that when the cloud properly stores users' data, the proof it generates can pass the verification of the TPA.

2) Auditing soundness: to assure that if the cloud does not possess users' intact data, it cannot pass the verification of the TPA.

3) Auditing without private key storage: to allow the user to utilize biometric data as fuzzy private key to accomplish data integrity auditing without private key storage.

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | Impact Factor: 7.488 |

||Volume 8, Issue 6, June 2020||

Overview of the Proposed Scheme:

Our proposed data integrity auditing scheme without private key storage is constructed based on the MBLSS and the linear sketch. As shown in Fig. 2, our proposed data integrity auditing scheme consists of the following three procedures: Key Generation, Signature Generation and Audit.

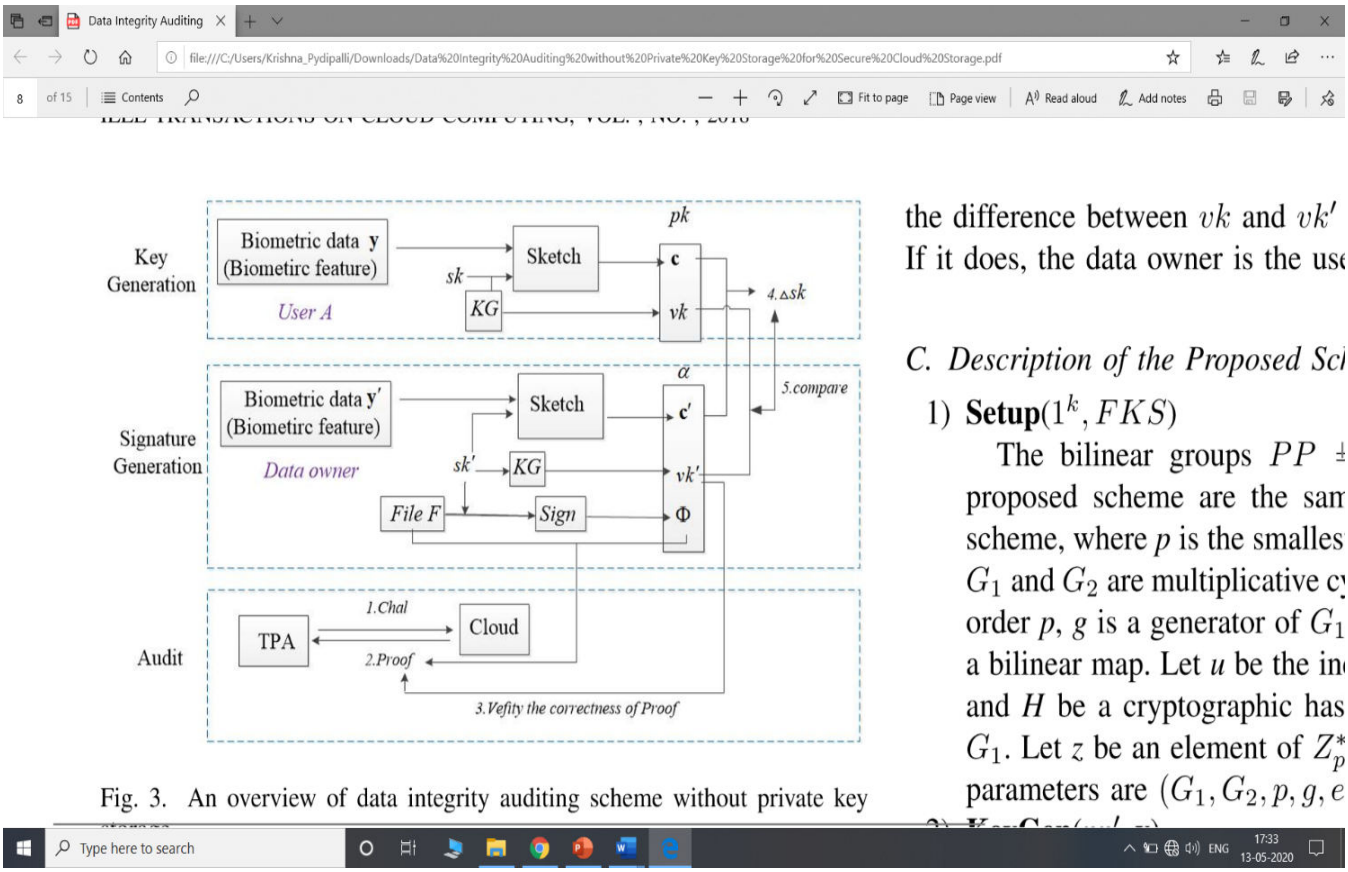


Fig. 3. An overview of data integrity auditing scheme without private key

Fig. 2. An overview of data integrity **auditing scheme without private key storage**

Key Generation. It includes Setup and KeyGen algorithms. Firstly, the public global parameter pp0 is generated in Setup algorithm. In the KeyGen algorithm, the user A, who wants to store his data in the cloud, extracts biometric data y in the phase of registration. Next, this user randomly generates a key pair (sk,vk). Finally, this user generates a sketch c of private key sk using y, which is used to code and correct the error of biometric data. The public key pk of our proposed scheme includes (vk,c).

Signature Generation. It consists of the SignGen algorithm. The data owner generates the signature of the file F, and uploads this file along with its signature to the cloud. Specifically, the data owner randomly generates a signing key sk0 and its corresponding verification key vk0, where sk0 is used to generate the sketch and the authenticators. Then the data owner generates a sketch c0 of signing key sk0 using the biometric data y0 extracted from him. He generates a data authenticator set Φ for file F with signing key sk0. The signature α of file F is (Φ,vk0,c0). The data owner sends {F,α} to the cloud, and deletes them from the local storage.

Audit. The ProofGen algorithm and ProofVerify algorithm are executed in this phase. In the ProofGen algorithm, the TPA sends an auditing challenge chal to the cloud. Upon receiving the chal, the cloud returns an auditing proof P to the TPA. In the ProofVerify algorithm, the TPA firstly checks the correctness of the proof P using the verification key vk0. And then, in order to confirm the identity of the data owner, the TPA recovers Δsk from c and c0 by using the technique of coding and error correction. Finally, the TPA verifies whetherthe difference between vk and vk0 truly corresponds to Δsk. If it does, the data owner is the user A;otherwise, he is not.

IV. EXPERIMENTAL RESULTS & DISCUSION

In this section, we evaluate the performance of our proposed scheme in experiments.

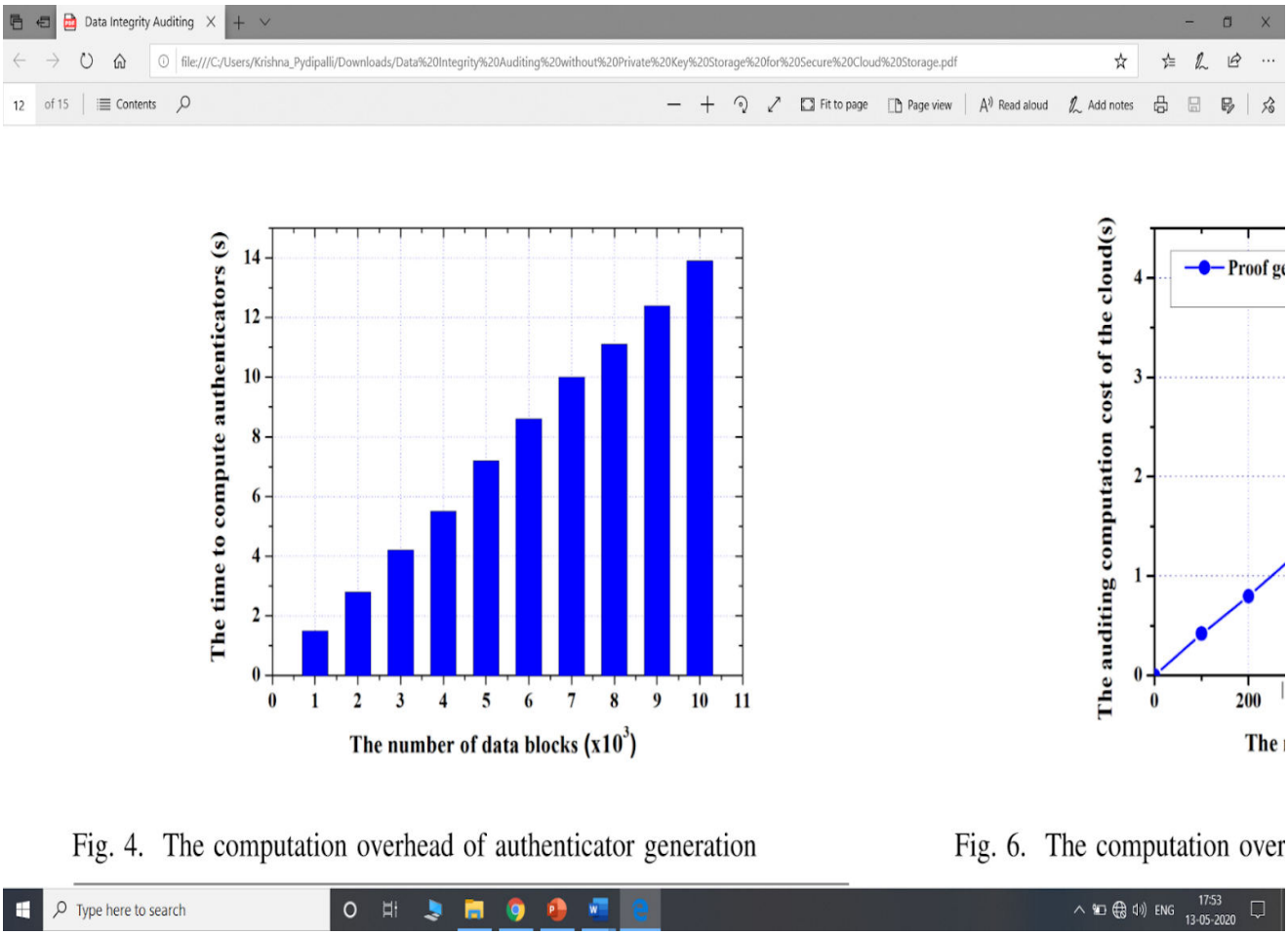
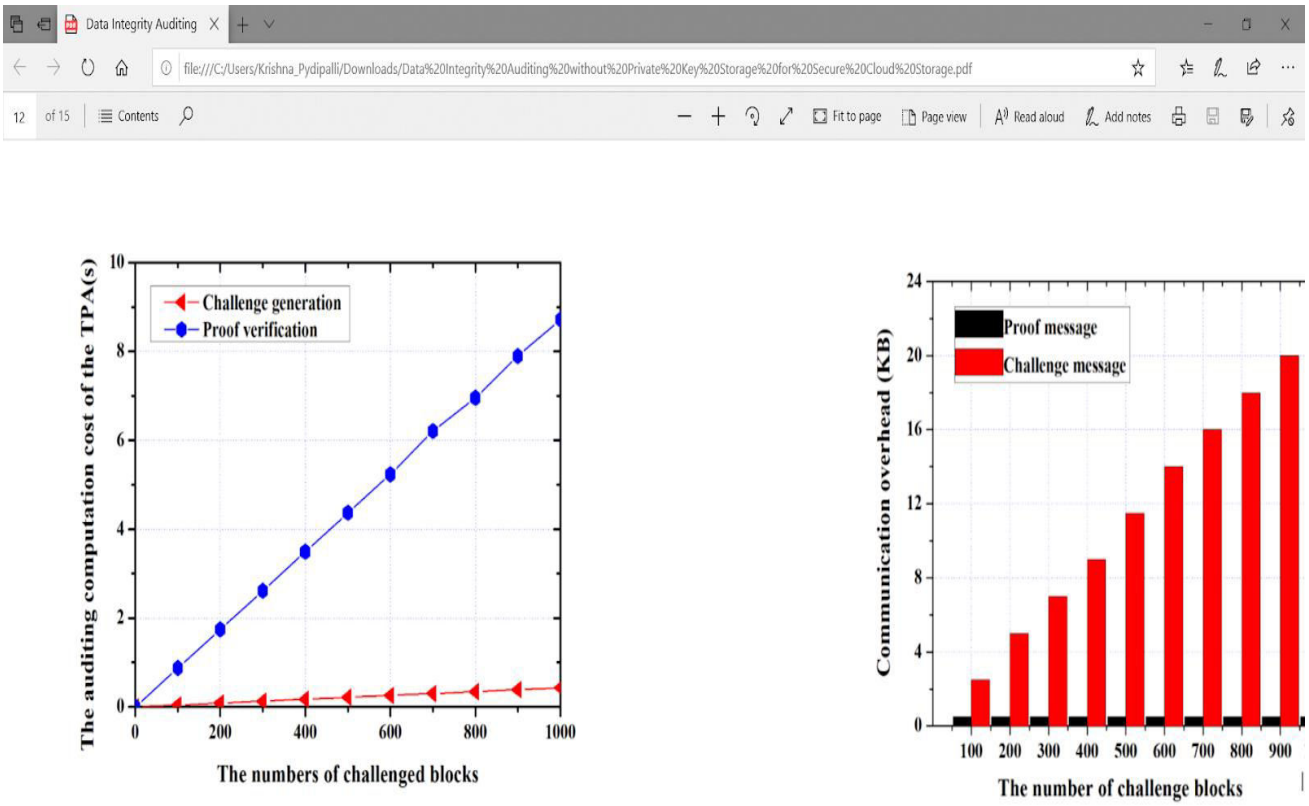


Fig. 4. The computation overhead of authenticator generation

Fig. 6. The computation over

Fig. 3. The computation overhead of authenticator generation

a) Authenticator generation. In order to evaluate the efficiency of authentication generation of our scheme, we compute the authenticators for different blocks from 0 to 1000 increased by an interval of 100. Fig. 3 shows that the computation overhead of authenticator generation linearly increases with the number of data blocks. The running time varies from 1.5s to 12.9s.



5. The computation overhead of the TPA in the phase of auditing

Fig. 7. The communication overhead of challenge messa

Fig. 4. The computation overhead of the TPA in the phase of auditing

b) Auditing. In order to evaluate the performance of auditing in our scheme, we respectively show the time spent on the TPA and the cloud. The experimental results are presented in Fig. 4 and Fig. 5. In the experiment, we choose to challenge different blocks from 0 to 1000 increased by an interval of 100.From Fig. 4, we have the observation that the auditing computation overhead of the TPA is mainly from challenge generation and proof verification. The running time of challenge generation ranges from 0.038s to 0.395s. The running time of proof verification is linear with the number of the challenged data blocks, ranging from 0.795s to 8.685s.

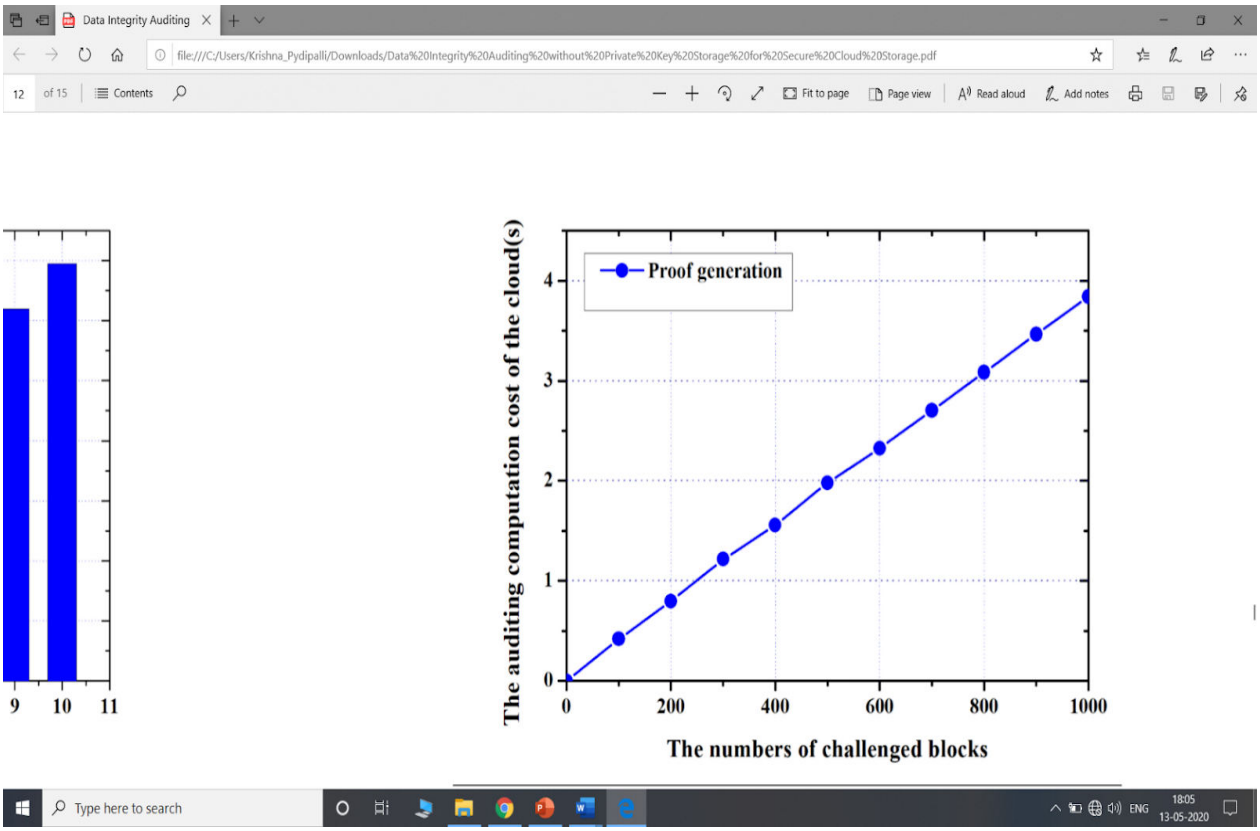


Fig. 5. The computation overhead of the cloud in the phase of auditing

As shown in Fig. 5, the running time ofproofgenerationrangesfrom0.401sto3.793sonthe cloud side. From the above experiments, we can infer that the auditing computation overhead of the TPA and the cloud both linearly increases with the number of the challenged blocks. The trade-off here is that, with more challenged blocks, the result of integrity auditing is more accurate, meanwhile, the auditing work gets more cumbersome.

V. CONCLUSION

In this scheme, we explore how to employ 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 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 experimental results show that our proposed scheme is provably secure and efficient.

REFERENCES

[1] H. Dewan and R. C. Hansdah, "A survey of cloud storage facilities," in 2011 IEEE World Congress on Services, July 2011, pp. 224–231.
[2] 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.
[3] 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.
[4] N. Garg and S. Bawa, "Rits-mht: Relative indexed and time stamped merkle hash tree-based data auditing protocol for cloud computing," Journal of Network & Computer Applications, vol. 84, pp. 1–13, 2017.
[5] 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.
[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.

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | Impact Factor: 7.488 |

||Volume 8, Issue 6, June 2020||

[7] 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.
[8] 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.
[9] 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.
[10] J. 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.

BIOGRAPHY

Ms.T. SujiLatha has received her M. Tech. in Computer Science from JNTU, Anantapur. She is dedicated to teaching field from the last 9 years. She has guided 10 P. G. and 30 U. G. students. Her research areas included in Networking and cloud Computing. At present she is working as Associate Professor and HOD in Computer Science and Engineering at Gokula Krishna College of Engineering, Sullurpet, SPSR Nellore (DT) Andhra Pradesh, India.

Mr.V. Kasthuraiah has received his B.C.A degree in Computer Science from Rayaloaseema Institute of information and management science, Tirupathi affiliated to Sri Venkateswara University, Tirupathi in 2003, Received his PG degree in Master of Computer Applications from Vels's College of Science, Chennai, affiliated to Madras University, Chennai in 2006 and pursuing M. Tech. degree in Computer Science from Gokula Krishna College of Engineering, Sullurpet, affiliated to JNTU, Anantapur., AndhraPradesh, India.