Block Chain Based Enabled Auditor Tracing over Sanitized outsource data with enhanced data integrity mechanism in cloud storage

J. Sharon Christina , Raja A

IV th Sem M.Tech , Assistant Professor in Department of Computer Science and and Engineering,
C ByreGowda Institute of Technology,
Kolar, India.

Abstract : With circulated capacity organizations, customer's stores the data in cloud data center remotely and comprehend offering data to others. Data remote decency looking for guarantee the genuineness of the data proposed to set aside in the cloud. In some normally appropriated systems capacity, for instance, the Electronic Prosperity RECORDS (EHRs) structure, archive cloud may take tricky correct data. The tricky right data are not share introduced iii cloud. Encoding they mutual record be comprehend the whole smooth data concealing; in any case, it will make this normal archive unfit can be used by others. The best strategy for recognize the data granting to sensitive information stowing endlessly in data remote trustworthiness looking at still has been not examined till now. We introduced data remote dependability assessing an arrangement that recognizes data giving to fragile information stowing endlessly in the paper. In the arrangement are disinfect the information are squares contrasting with the tricky information for the archive and changes in inform blocks" marks in genuine oties to clean the record. These imprints are used to affirm the uprighties to the cleaned record in the time of genuineness looking at. Consequently, our arrangement makes the record set aside are the cloud-fit to be shared and used by relying others upon to the essential the smooth data is concealed, there data remote decency looking at so far prepared to viably be executed. Then, the plot proposed relies upon character-based cryptography, which smoothes out the jumbled confirmation of the board. The security assessment and the presentation appraisal show that the proposed plot is secure and capable.

IndexTerms **- Cloud storage, data integrity auditing, data sharing, sensitive information hiding,**

I. INTRODUCTION

In the organizations gave by the cloud, customs have the advantage to rewtely can moreover outfit sharing off with others. remote data genuineness reviewing plan is proposed to ensure the reliability of the data which is taken care of in the cloud. For an example of the conveyed stockpiling structures, the Electronic Prosperity Records may contain some fragile information. This fragile information must not be introduced to various customers when the cloud record is shared. The encryption of the whole common report can conceal the tricky information, yet this will weaken various customers to use this shared record. Till now there is no remote data uprightness looking at an arrangement to share the data by disguising the sensitive information. To address this particular issue, a remote data trustworthiness assessment plan has been proposed in this paper grants data conferring to unstable information stowing ceaselessly [2]. The critical point in this arrangement is, there is a sanitizer that is used to clean the squares of data which is contrasting with the fragile information to the record and these data blocks makes" marks change into the considerable imprints to the purged archive. This changed imprints are used for the check of the trustworthiness of the purified archive during the time of decency examining [4). In this way in like manner, this arrangement can enable the record set aside in shared with the condition that the data is triply
Similarly as the reunite data trustworthiness inspecting. This proposed plot relies upon character-based cryptography, which makes the befuddled confirmation the board essential [5]. The assessment of security and the apprise of execution show this proposed plot is significantly increasingly powerful and secure [6].

II. LITERATURE SURVEY

Composing is a basic development in the method of programming creating. It is imperative to realize the time factor and quality alongside the economy of the organization required before the improvement of any apparatus. The further advance, once these things are known, is to choose the working framework and the language which could be made use to build up the accepted apparatus. When the undertaking is begun to be worked by the developer he requires bunches of help remotely which can be procured from the cultivated senior programming engineers or from destinations and from certain books. All the above considerations and thoughts are considered before the framework being worked to create the proposed framework in a superior manner.

Provable Data Possession at Untrusted Stores

G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner,Z. Peterson, and D. Song, present a model for provable As the information is developing extremely quicker, it is oppressive for all the clients to store the specific measure of information in the neighborhood stockpiling Frameworks, consequently, a considerable lot of people and the associations need to be cl0ud data[8]. In any case, the information which is put away by the cloud may get defiled or even maybe lost because of the invaluable programming mistake, equipment imperfection and blames in the cloud of human.Therefore, the PDP model is referred to colossal enlightening assortments in comprehensively scattered limit systems. We present the 2 to secured data of

PDP model in which that capable than its take action in past , regardless, differ between & plots in achieve progressively helpless affirmations. its server is low (0r even consistent), as activity introduced to straight in the size of the data.

Dynamic Provable Data Possession

C. Erway, C. Papamanthou, and R. Tamassia consider the issue of suitably indicating the tolerability of informational index aside The critical point in this arrangement is, there is a sanitizer that is used to clean the squares of data which is contrasting with the fragile information to the record and these data blocks makes‟ marks change into the considerable imprints to the purged archive. This changed imprints are used for the check of the trustworthiness of the purified archive during the time of decency examining[4]. In this way in like manner, this arrangement can enable the record set aside in shared with the condition that the data is tricky and completely concealed, similarly as the remote data trustworthiness inspecting is up 'til now prepared to be successfully executed.which relaxes up the PDP m0del to help provable updates with putting perpetually information[3].We utilize another kind of affirmed word reference dependent right data. The rate cloud while dynamic data is a showcase change from 0(1) too(log n) (no log n)), for a record including n squares, with keeping up by equivalent (or better, freely) likelihood of shrewdness affirmation.0ur assessments see the data with stoppage is astoundingly L0w in the long run .We besides prompt the best way to deal with apply our DPDP plan to redistributed report frameworks and variety control structures (e.g., CVS).

III. SYSTEM MODEL AND SECURITY MODEL

A System Model 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) (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. 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.

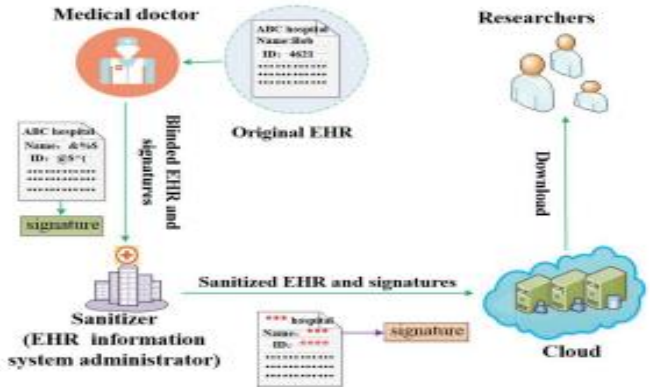

fig. 1: Example of EHRs

Here, we give an illustrative example for EHRs in Fig. 1. In this example, the sensitive information of EHRs contains two parts. One is the personal sensitive information (patient's sensitive information), such as patient's name and patient's ID number. The other is the organization's sensitive information (hospital's sensitive information), such as the hospital's name∗ . Generally speaking, the above sensitive information should be replaced with wildcards when the EHRs are uploaded to cloud for research purpose. The sanitizer can be viewed as the administrator of the EHR information system in a hospital. The personal sensitive information should not be exposed to the sanitizer. And all of the sensitive information should not be exposed to the cloud and the shared users. A medical doctor needs to generate and send the EHRs of patients to the sanitizer for storing them in the EHR information system. However, these EHRs usually contain the sensitive information of patient and hospital, such as patient's name, patient's ID number and hospital's name. To preserve the privacy of patient from the sanitizer, the medical doctor will blind the patient's sensitive information of each EHR before sending this EHR to the sanitizer. The medical doctor then generates signatures for this blinded EHR and sends them to the sanitizer. The sanitizer stores these messages into EHR information system. When the medical doctor needs the EHR, he sends a request to the sanitizer. And then the sanitizer downloads the blinded EHR from the EHR information system and sends it to the medical

doctor. Finally, the medical doctor recovers the original EHR from this blinded EHR. When this EHR needs to be uploaded and shared in the cloud for research purpose, in order to unify the format, the sanitizer needs to sanitize the data blocks corresponding to the patient's sensitive information of the EHR. In addition, to protect the privacy of hospital, the sanitizer needs to sanitize the data blocks corresponding to the hospital's sensitive information. Generally, these data blocks are replaced with wildcards. Furthermore, the sanitizer can transform these data blocks' signatures into valid ones for the sanitized EHR. It makes the remote data integrity auditing still able to be effectively performed. During the process of sanitization, the sanitizer does not need to interact with medical doctors. Finally, the sanitizer uploads these sanitized EHRs and their corresponding signatures to the cloud. In this way, the EHRs can be shared and used by researchers, while the sensitive information of EHRs can be hidden. Meanwhile, the integrity of these EHRs stored in the cloud can be ensured.

IV. THE PROPOSED SCHEME

An Overview In order to achieve data sharing with sensitive information hiding, we consider making use of the idea in the sanitizable signature [30] to sanitize the sensitive information of the file by introducing an authorized sanitizer. Nonetheless, it is infeasible if this sanitizable signature is directly used in remote data integrity auditing. Firstly, this signature in [30] is constructed based on chameleon hashes [31]. However, a lot of chameleon hashes exhibit the key exposure problem. To avoid this security problem, the signature used in [30] requires strongly unforgivable chameleon hashes, which will inevitable incur huge computation overhead [31]. Secondly, the signature used in [30] 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 in [30] is based on the PKI, which suffers from the complicated certificate management. 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. 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, 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

B. Description of the Proposed Scheme In our scheme, an original file F is divided into n blocks (m1, m2,..., mn), where mi ∈ Z∗ p denotes the i-th block of file F. Assume the user's identity ID is l-bit, which is described as I D = (I D1, I D2,..., I Dl) ∈ {0, 1} l . In previous remote data integrity auditing schemes [5], [12], a signature SSig is used to guarantee the integrity of the file identifier name. In our scheme, we also employ a similar identity-based signature SSig to guarantee the integrity of the file identifier name and the correctness of verification values. Assume ssk is the signing private key used to generate file tag in signature SSig and is held by user. Under such an assumption, our scheme is more clear and simple. Let K1 be the set of indexes of the data blocks corresponding to the personal sensitive information of the file F. Let K2 be the set of indexes of the data blocks corresponding to the organization's sensitive information of the file F. In order to preserve the personal sensitive information of the file from the sanitizer, the data blocks whose indexes are in the set K1 should be blinded before the file is sent to the sanitizer. Assume the blinded file is F∗= m∗ 1, m∗ 2,..., m∗ n which is different from the original file F= (m1, m2,..., mn) in index set K1. That is to say, mi = m∗ i only if i ∈ [1, n] and i ∈/ K1; otherwise, mi = m∗ i . To unify the format, the sanitizer needs to sanitize the blinded data blocks with wildcards. Furthermore, to protect the privacy of organization, the sanitizer also needs to sanitize the data blocks corresponding to the organization's sensitive information. The sanitized file is F= m1, m2,..., mn which is different from the blinded file F∗= m∗ 1, m∗ 2,..., m∗ n in index set K1

K2. That is to say, m∗ i = m i only if i ∈ [1, n] and i ∈/ K1

K2; otherwise, m∗ i = m i . In general, there is not too much sensitive information in a file, which makes the sanitizer only need to sanitize a few fields. For example, the sensitive information of the EHRs only contain the fields such as patient's name, patient's ID number and hospital's name. Thus, in EHRs, only these fields containing the sensitive information need to be sanitized, and other fields do not need to be sanitized. The details of the proposed scheme are as follows.

1) Algorithm Setup(1k)

a) 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.

b) The PKG randomly chooses an element x ∈ Z∗ p, elements µ , µ1, µ2,...,µl, u, g2 ∈ G1 and a cryptographic hash function H : {0, 1} ∗ → G1. The process of private key generation.

c) The PKG computes the public value g1=gx and the master secret key msk = g2 x .

d) The PKG publishes system parameters pp = (G1, G2, p, e, g, µ , µ1, µ2,...,µl, u, g1, g2, H,
f) and holds the master secret key msk.

2) Algorithm Extract(pp, msk,ID) This process is illustrated in

After receiving the user's identity I D = (I D1, I D2,..., I Dl) ∈ {0, 1} l , the PKG randomly picks a value rI D ∈ Z∗ p and computes skI D = (skI D,skI D) = (g2 x · (µ l j=1 µj I Dj)rI D , grI D) as the private key of the user ID. The PKG sends it to

the user ID. b) The user ID verifies the correctness of the received private key skI D by checking whether the following equation holds or not. e(skI D, g) = (g1, g2) · e(μ l j=1 μj I Dj ,sk I D).

3) Algorithm SigGen(F,skID,ssk, name)

a) The user ID randomly chooses a value r ∈ Z∗ p, and calculates a verification value gr. Then the user ID randomly chooses a seed k1 ∈ Z∗ p as the input secret key of pseudo-random function f. The user ID employs the secret seed k1 to calculate the blinding factor αi = fk1 (i, name)(i ∈ K1) which is used to blind the data blocks corresponding to the personal sensitive information, where name ∈ Z∗ p is a random value chosen as the file identifier.

b) In order to preserve the personal sensitive information from the sanitizer, the user ID should blind the data blocks corresponding to the personal sensitive information of the original file F before sending it to the sanitizer. The indexes of these data blocks are in set K1. The user ID computes the blinded data block m∗ i = mi + αi for each block mi ∈ Z∗ p (i ∈ K1) of the original file F. The blinded file is F∗= m∗ 1, m∗ 2,..., m∗ n , where m∗ i = mi only if i ∈ [1, n] and i ∈/ K1; otherwise, m∗ i = mi .

c) For each block m∗ i ∈ Z∗ p (i ∈ [1, n]) of the blinded file F∗, the user ID calculates

V. PERFORMANCE EVALUATION

In this section, we first give the functionality comparison among our scheme and several related schemes, and the computation overhead comparison between our scheme and Shacham and Waters scheme [4]. And then discuss the communication overhead and the computation complexity of our scheme. At last, we evaluate the performance of our scheme in experiments. A. Functionality Comparison We give the functionality comparison of our scheme with several related schemes [4], [20], [32]–[34]. As shown in Table II, our scheme is the only scheme that can satisfy all of the following properties: public verifiability, certificate management simplification, data sharing and sensitive information hiding. Note that schemes [4], [20], [32]–[34] all cannot support the sensitive information hiding. B. Performance Analysis and Comparison We define the following notations to denote the operations in our scheme. Let HashG1, ExpG1 and MulG1 respectively denote one hashing operation, one exponentiation operation and one multiplication operation in G1. Similarly, SubZ∗ p , MulZ∗ p and AddZ∗ p denote one subtraction operation, one multiplication operation and one addition operation in Z∗ p, respectively. Pair denotes one pairing operation. MulG2 and ExpG2 respectively denote one multiplication operation and one exponentiation operation in G2. n is the total number of data blocks. c is the number of challenged data blocks. d1 is the number of data blocks corresponding to the personal sensitive information. d2 is the number of data blocks corresponding to the organization's sensitive information. l is the length of user identify. |n| is the size of an element of set [1, n], |p| is the size of an element in Z∗ p, and |q| is the size of an element in G1.

VI. 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 efficient.

REFERENCES

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

[2] G. Ateniese et al., "Provable data possession at untrusted stores," in Proc. 14th ACM Conf. Comput. Commun. Secur., 2007, pp. 598–609.

[3] A. Juels and B. S. Kaliski, Jr., "Pors: Proofs of retrievability for large files," in Proc. 14th ACM Conf. Comput. Commun. Secur., 2007, pp. 584–597.

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

[5] C. Wang, S. S. M. Chow, Q. Wang, K. Ren, and W. Lou, "Privacypreserving public auditing for secure cloud storage," IEEE Trans. Comput., vol. 62, no. 2, pp. 362–375, Feb. 2013.

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

[7] C. Guan, K. Ren, F. Zhang, F. Kerschbaum, and J. Yu, "Symmetrickey based proofs of retrievability supporting public verification," in Computer Security—ESORICS. Cham, Switzerland: Springer, 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," J. Netw. Comput. Appl., vol. 82, pp. 56–64, Mar. 2017.

[9] J. Sun and Y. Fang, "Cross-domain data sharing in distributed electronic health record systems," IEEE Trans. Parallel Distrib. Syst., vol. 21, no. 6, pp. 754–764, Jun. 2010.

[10] G. Ateniese, R. D. Pietro, L. V. Mancini, and G. Tsudik, "Scalable and efficient provable data possession," in Proc. 4th Int. Conf. Secur. Privacy Commun. Netw., 2008, Art. no. 9.

[11] C. Erway, A. Küpçü, C. Papamanthou, and R. Tamassia, "Dynamic provable data possession," in Proc. 16th ACM Conf. Comput. Commun. Secur., 2009, pp. 213–222.

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

[13] J. Yu, K. Ren, C. Wang, and V. Varadharajan, "Enabling cloud storage auditing with key-exposure resistance," IEEE Trans. Inf. Forensics Security, vol. 10, no. 6, pp. 1167–1179, Jun. 2015.

[14] J. Yu, K. Ren, and C. Wang, "Enabling cloud storage auditing with verifiable outsourcing of key updates," IEEE Trans. Inf. Forensics Security, vol. 11, no. 6, pp. 1362–1375, Jun. 2016.
[15] J. Yu and H. Wang, "Strong key-exposure resilient auditing for secure cloud storage," IEEE Trans. Inf. Forensics Security, vol. 12, no. 8, pp. 1931–1940, Aug. 2017.
[16] J. Yu, R. Hao, H. Xia, H. Zhang, X. Cheng, and F. Kong, "Intrusionresilient identity-based signatures: Concrete scheme in the standard model and generic construction," Inf. Sci., vols. 442–443, pp. 158–172, May 2018.
[17] B. Wang, B. Li, and H. Li, "Oruta: Privacy-preserving public auditing for shared data in the cloud," in Proc. IEEE 5th Int. Conf. Cloud Comput. (CLOUD), Jun. 2012, pp. 295–302.
[18] 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, pp. 130–139, Mar. 2016.
[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 Trans. Big Data, to be published, doi: 10.1109/TBDATA.2017.2701347.
[20] B. Wang, B. Li, and H. Li, "Panda: Public auditing for shared data with efficient user revocation in the cloud," IEEE Trans. Serv. Comput., vol. 8, no. 1, pp. 92–106, Jan./Feb. 2015.