Ensuring User Data's Safety in the Cloud: A Privacy-Focused Approach to Remote Data Integrity Checks

Kondragunta Rama Krishnaiah, Professor, Department of Computer Science and Engineering, R K College of Engineering, Vijayawada - 521456, Andhra Pradesh, India, email: kondraguntark@gmail.com

Abstract

Remote Data Integrity Checking (RDIC) is a crucial technique that allows us to verify the integrity of data stored in the cloud. Over the years, various RDIC protocols have been proposed in academic literature, but they have faced challenges related to complex key management. In our project, we introduce an innovative approach to Identity-Based RDIC, leveraging secure key generation to simplify the system's complexity and reduce the cost of establishing and managing the public key infrastructure (PKI) in RDIC schemes. Our proposed solution revolves around three main entities: the cloud server, the user, and the third-party auditor (TPA). The TPA plays a critical role in providing evidence of data integrity, ensuring that our data remains unaltered and secure within the cloud environment. Additionally, we have implemented robust security mechanisms to safeguard against unauthorized access and data tampering. Should any unauthorized modifications occur, whether by the cloud server or an unauthorized individual, the user can retrieve their original data from a trusted backup server, thus preventing data loss or corruption. The Third-Party Auditor (TPA) takes responsibility for monitoring the integrity of the cloud data on behalf of the cloud users. This is particularly useful in scenarios where users may not have the time or resources to continuously monitor their data's integrity, as the TPA conducts regular checks and provides auditing reports back to the cloud users. This way, users can have peace of mind knowing that their data is being regularly validated and protected. By leveraging Identity-Based RDIC and incorporating the TPA, our solution aims to address the challenges of traditional RDIC protocols and provide a more efficient and secure data integrity checking mechanism for cloud storage environments.

Keywords: Cloud storage, data integrity, privacy preserving, identity-based cryptography.

1. INTRODUCTION

Cloud computing is developing as a prevailing innovation class in the business network. While the advantages of cloud computing are clear, it likewise presents new security challenges. See [1] for an exhaustive overview. Cloud stockpiling administrations, which enable data proprietors to relocate their data from neighborhood stockpiling frameworks to the cloud, soothe the weight of capacity administration and support. They offer versatile, pay-on-request, area in ward stockpiling administration for users [2]. Be that as it may, this new sort of data facilitating administration triggers numerous new security challenges [3]. To be sure, the Cloud Security Alliance (CSA) [4] sees Data Loss and Leakage as the second among the best seven security dangers to cloud computing. For instance, Business Insiders detailed 1 that a few data were annihilated in an EC2 cloud administrations crash in 2011. Furthermore, it isn't compulsory for the specialist co-ops to report these episodes. In cloud stockpiling setting, because of the loss of physical possession of data, a noteworthy worry of cloud users is whether their data are put away in the cloud securely. On the off chance that the cloud servers are not completely trusted, the integrity of put away data couldn't be guaranteed. Thusly, there is a requirement for the improvement of conventions enabling the data proprietors to check that their data are accurately put away in the cloud.

Customary cryptographic advances for data integrity checking, for example, message validation codes and computerized marks are not perfect to Remote data integrity checking (RDIC) because the first record is required in the confirmation strategy. It is a costly exercise to download the whole record from the cloud for confirmation. Blum exhibited a plan empowering data proprietor to confirm the integrity of remote data without express learning of the whole data [5]. Provable data possession (PDP) [6, 7], presented by Ateniese et al., is a procedure for approving data integrity over remote servers. In a commonplace PDP framework, the data proprietor creates some metadata for a record, which will be utilized later for integrity checking by means of a test reaction convention with the remote server. Data proprietor at that point sends his document to a remote server, which might be untrusted, and erases the record from its neighborhood stockpiling. To create a proof that the server has the record in its unique frame, the server figures a reaction to a test from the verifier. The verifier approves that the document isn't being altered by means of checking the rightness of the reaction. Ateniese et al. likewise proposed two PDP plots by using the RSA based homomorphic straight authenticators. In the meantime, Juels et al. proposed the thought of confirmation of retrievability (POR) [8], in which blunder amending codes and spot-checking are utilized to accomplish the properties of possession and retrievability of records. PDP and POR have turned into an examination hotspot of secure cloud stockpiling and various plans have been proposed [9, 11]. Other than integrity checking, three propelled highlights, specifically, data elements, open certainty and protection against verifiers are additionally considered for handy purposes.

2. RELATED WORK

The evidence for data integrity in the cloud stockpiling condition has assembled a considerable measure of research consideration. For confirmation of remote data integrity in the inconsistent cloud, many examining methods have been proposed. As the measure of data created is surpassing the data stockpiling limit, it is costly for little associations to always modernize their equipment and keep up vast stockpiles at whatever point advantageous data is produced. This entanglement is additionally exasperated with utilization of overwhelming data transfer capacities for expansive record exchanges with the framework having just restricted CPU and battery control [1]. In 2004, Boneh et.al proposed [11] single watchword accessible encryption where all users can send the data utilizing open keys yet just valid users can look through the data utilizing their distributed private keys. Be that as it may, the encryption strategy utilized upgrades the multifaceted nature of this technique. In a Provable Data Possession Scheme has been proposed by the creators. This plan guarantees whether the data put away in the Cloud archives is completely held by the remote cloud server by creating metadata and looking at the hash esteems. This plan has high overhead and expends additional time in light of the fact that the hash is kept running over the whole record. In 2007, Ari Juels et.al proposed [10] a plan by utilizing sentinels for Proof of hopelessness for expansive records. It utilized a solitary key independent of the document size or number that should be checked. While this plan expends less time, it can't deal with dynamic data and an expanded number of inquiries. Additionally, this includes encryption of record utilizing a mystery key, or, in other words to deal with particularly when the document is substantial. In the creators have talked about conservative evidence of retrievability utilizing two POR plans based on the homomorphic direct authenticators for private and open confirmation. Despite the fact that the speed of confirmation is upgraded here, however the handling costs too are high. Dodis et al. in his plan lessened the measure of the message, yet the plan still experienced the confinement because of linearity between the lengths of verification reaction to the quantity of components in the data square. Likewise, this plan could just help private check of data bringing about expanded overhead on the data owner. In the creators have proposed a security saving approval framework for the cloud which forms the smaller scale data and sends unknown data to the

cloud specialist co-op for reconciliation with extra data to get the outcomes. In the creators discuss Privacy Aware Data Storage and Processing in Cloud Computing Architectures. This paper manages sealed cryptographic coprocessor, arranged by bona fide third party to give a safe stage free from unapproved get to. The creators in have talked about a contextual analysis "RACS" for Cloud Storage Diversity to keep away from merchant secure and basically limit the expenses. In this, creators proposed a safe reviewing plan, however experienced the confinements of high computational costs which were specifically relative to gathering and data estimate. Wang et al, in proposed a plan for open auditability and data elements for capacity security in cloud computing. Nonetheless, this plan likewise experiences high computational costs direct to the measure of the data. In Ming Ii et.al proposed an Authorized Private Keyword Search over Encrypted Data in Cloud Computing. Here different data owners encode their reports and utilize watchwords and records to permit looks. This plan likewise bolsters multi-dimensional range inquiries in any case, experiences the restrictions of high computational expenses and unnecessary overhead. Ning Cao et.al in proposed protection safeguarding multi-watchword look where a user can look through the cloud data with various question catchphrases. This plan likewise experiences the confinements of high computational expenses and accepts that the cloud server can be confided in without fail. In reference the creators propose a strategy for giving a decision of encryption calculations to the users, who can choose any calculation to anchor their data. The creators here have not thought about the expense of keeping up every one of these calculations. Likewise, this strategy must be utilized for static data and set number of questions. In creators discuss 3 level securities for the user; however, they have not considered the cost part of this model. Likewise, it experiences the difficulties, for example, Data bolts by cloud supplier, adaptation to non-critical failure and calamity recuperation systems. In the creators have proposed a fluffy identity-based data reviewing instrument, where a user's identity can be seen as an arrangement of graphic characteristics, in any case, creators have not considered the expense and multifaceted nature part of this model. Likewise, they have not tried it in a continuous domain. In Salah H. Abbdal et. al, have proposed a component to check the data integrity utilizing TPAs based on homomorphic direct confirmation and an elliptic bend advanced mark calculation to help open unquestionable status. In spite of the fact that this strategy diminishes the time intricacy, they have not considered the expense and overhead included. Yong Yu et al. in have proposed ID based remote data integrity checking procedure which utilizes key-homomorphic cryptographic crude to lessen the framework cost and multifaceted nature for a PKI Framework. Despite the fact that, the security is upgraded, the surprising expense gauge isn't mulled over. In the creators have proposed a novel open check conspire for the cloud stockpiling utilizing lack of definition jumbling. Be that as it may, the creators have not examined how to oppose a malignant auditor and how to decrease the cloud server overhead.

3. PRELIMINARIES

3.1 Remote data integrity checking for secure cloud storage

A publicly verifiable remote data integrity checking design [7, 11] for security cloud stockpiling is delineated in Fig 1. Three distinct elements, specifically, the cloud user, the cloud server and the third-party auditor (TPA) are engaged with the framework. The cloud user has expansive measure of data to be put away on the cloud server without keeping a nearby duplicate, and the cloud server has noteworthy storage room and calculation assets and gives data stockpiling services to cloud users. TPA has ability and capacities that cloud users don't have and is trusted to check the integrity of the cloud data for the benefit of the cloud user upon demand. They have their own commitments and advantages separately. The cloud server can act naturally intrigued, and for his own advantages, for example, to look after notoriety, the cloud server may conceal data defilement episodes to users. In

any case, we accept that the cloud server has no motivating forces to uncover the facilitated data to TPA in light of directions and budgetary impetuses. The TPA's activity is to play out the reviewing for sake the cloud user in the event that that the user has no time, assets or plausibility to screen his data. Be that as it may, the TPA is additionally inquisitive and may attempt to reason some data of the data amid the reviewing procedure.

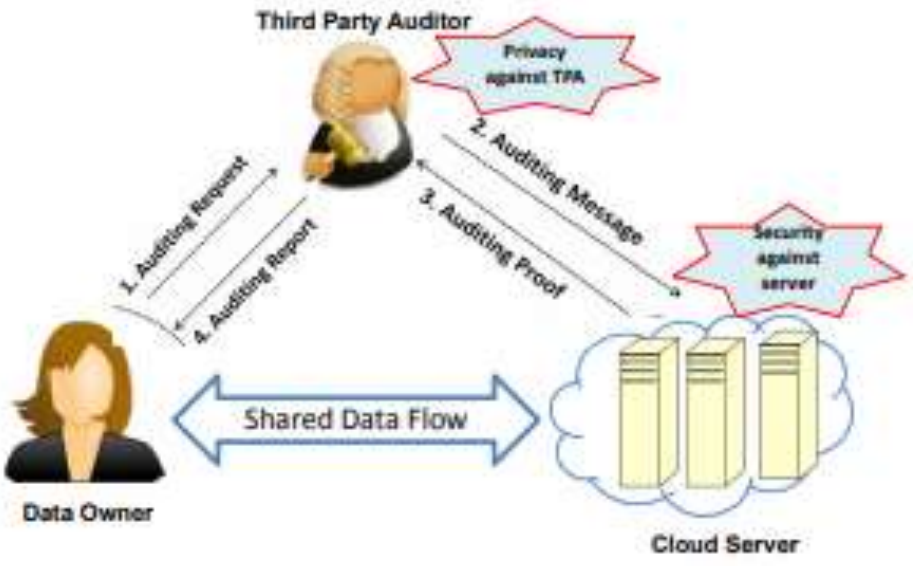

Fig. 1 The system model of publicly verifiable remote data checking

3.2 System components and its security

Following the definition in [9], an open remote data checking plan or reviewing plan is a tuple of five calculations, in particular, Setup, TagGen, Challenge, ProofGen and ProofCheck, which can be portrayed as pursues.

- Setup. On info a security parameter (k), this calculation creates the general population key (pk) and mystery key (sk) for the data owner. pk is open to everybody except sk is kept mystery by the data owner.
- TagGen. On info the key match (pk, sk) and a data square (mi), this calculation yields a tag (Dmi) for the square, which will be utilized for open confirmation of data integrity.
- Challenge. TPA produces a test chal to ask for the integrity evidence of the record by sending chal to the server.
- GenProof. The server processes reaction R utilizing chal, the document and the labels, and returns R to TPA.
- CheckProof. TPA approves reaction R utilizing chal, the labels and open key pk. Mystery key sk isn't required in a publicly verifiable data integrity checking plan.

4. PROPOSED SYSTEM

We propose a substitution development of identity-based (ID-based) RDIC convention by making utilization of key homomorphism cryptology crude to reduce the framework quality and furthermore the incentive for building up and dealing with the overall population enter validation system in PKI based for the most part RDIC plans. We tend to formalize ID based RDIC and its security show together with security against a noxious cloud server and zero data protection against a third-party auditor. The arranged ID-based RDIC convention releases no data of the held learning to the auditor

all through the RDIC technique. The new development is confirming secure against the noxious server inside the nonexclusive group display and accomplishes zero data protection against the auditor. Inside and out security investigation results show that the arranged convention is evidently secure and sensible inside these present reality applications. We Extend this work with time range based third party auditor framework and recuperation of document once information integrity checking issue happen.

Integrity Scrutinizing Maneuver Algorithm

The proposed algorithm has two parts:

Algorithm 1: General

Step 1: The client (owner) of the data registers with the cloud services.

Step 2: A mystery key for login is sent to the email id of the client for secure access of the cloud.

Step 3: After login, client uploads its record to the cloud repository. A cryptographic encryption key is sent to the mail id of the client.

Step 4: Every client record transferred or downloaded to and from the cloud repository experiences the encryption plot by producing new keys for each bit of data.

Step 5: Admin reviews the whole working, screens client data and issues admonitions if there should be an occurrence of suspicious exercises.

Step 6: The Auditor confirms the client documents, adds metadata to the record and stores them in the repository.

Algorithm 2: Generation of Metadata and Integrity Checking

Step 1: The client wants to store the file (F) in the cloud repository. Each file is divided into 'i' blocks.

Step 2: Each 'i' block is divided into 'j' bits.

Step 3: 'k' number of bits out of 'j' bits of 'i' blocks are selected for the construction of Metadata.

Step 4: Generation of metadata is done by the function H(m,n) which is elucidated as follows:

$$H(m, n) \rightarrow \{1...j\},\ m \in \{1...i\},\ n \in \{1...k\} \text{ ---------- (1),}$$

Where 'k' is the number of bits per block. Function H(m,n) gives the n th bit in the m th data block. Value for 'k' is a secret given by the Auditor. Each data block has 'k' bits and total bits for all 'i' blocks is given as (i*k) bits. "jm" represents the k bits of meta data form th data block.

Step 5: The metadata from the data block "jm" is encrypted and modified to metadata "Jm". Let "G" bethe function which generates k bit integer "αm" for each m. This is kept a secret with the Auditor and is defined as:

$$G: m \rightarrow \alpha m,\ \alpha m \in \{0 \ldots 2^i\} \text{----------(2)}$$

Step 6:For metadata"jm" of each data block the number αm is added to get a new k bit number given below:

$$Jm = jm + \alpha m \text{---------- (3)}$$

Step 7: The metadata generated is now clubbed together and affixed to the user's file F before saving it to cloud.

Step 8: If the client wants to verify integrity of the m th data block, the Auditor throws a to the cloud server by specifying the block number m and bit number n generated using the function "H", which is known only by the Auditor. The Auditor also specifies the position at which the metadata corresponding to block m is appended. Metadata is a k bit number.

Step 9: The cloud archive server sends the response for the verification to the client via Auditor.

Step 10: Metadata of the response is decrypted using "αm". The bit in decrypted metadata is compared with the bit send in the response by the cloud. If both matches, then the data is intact with its integrity maintained and otherwise if a mismatch is found. In other words, the cryptographic key generated by TPA is compared to the cloud archive's cryptographic key. If both matches, then the data integrity is not breached and vice versa.

5. METHODOLOGY

Data integrity of a remote data checks gives the security to the user data by keeping the different third-party auditor [6] and by then making the framework secure. Alongside the cloud stockpiling servers the record signature producing servers must be clubbed, the third party auditor will be given the entrance and made the whole substance accessible to him at whatever point the auditor asks for the document which thus will make the danger of data misfortune making uncertain and bringing about the expansion of expense. The framework engineering of the proposed framework is as appeared in Figure 2.

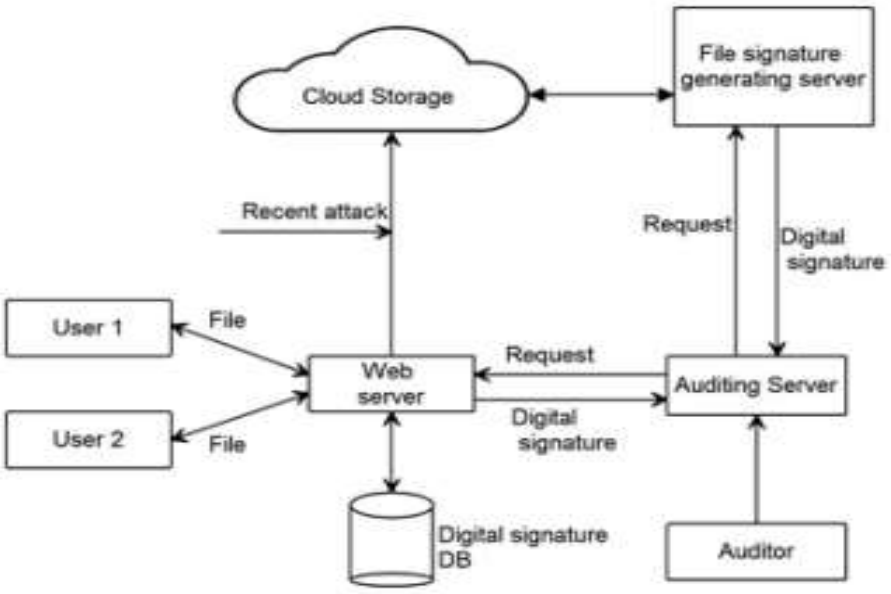

Figure 2: System Architecture

What's more, to dispose of this previously mentioned downside we are presenting the document signature server which thus will get the record content before passing it to the auditor and labeling the computerized mark to the document asked. What's more, by utilizing the computerized signature the substance presented to the auditor is restricted and the expense can be impressively diminished.

6. CONCLUSION

In this paper we have thought of a new thought of Privacy Preserving for Remote Data Based on Identity with High Performance in Cloud Storage which will furnish the ideal data protection with the low activity over the system close by high security for the data put away by the user. Also, gives the

down to earth was of proficiently exhibiting the protection and the execution esteems with the data security.

REFERNCES

[1] Diogo A. B. Fernandes, Liliana F. B. Soares, Joo V. Gomes, Mrio M. Freire, Pedro R. M. Incio, Security issues in cloud environments: a survey, International Journal of Information Security, doi:10.1007/s10207-013-208- 7 (2013) 1 - 58.

[2] M. Armbrust, A. Fox, R. Griffith, A. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, et al, A view of cloud computing, Communications of the ACM, 53 (4) (2010) 50–58.

[3] L. Wei, H. Zhu, Z. Cao, X. Dong, W. Jia, Y. Chen, A. V. Vasilakos, Security and privacy for storage and computation in cloud computing, Information Sciences, 258 (10) (2014) 371–386.

[4] Cloud Security Alliance, Top threats to cloud computing, http://www.cloudsecurityalliance.org, 2010.

[5] M. Blum, W. Evans, P. Gemmell, S. Kannan, M. Naor, Checking the correctness of memories, in: Proc. 32nd Annual Symposium on Foundations of Computer Science (FOCS 1991), pp. 90-99, 1991.

[6] G. Ateniese, R. C. Burns, R. Curtmola, J. Herring, L. Kissner, Z. N. J. Peterson, D. X. Song, Provable data possession at untrusted stores, in: Proc. 14th ACM Conference on Computer and Communications Security (ACM CCS 2007), pp. 598–609, 2007.

[7] G. Ateniese, R. C. Burns, R. Curtmola, J. Herring, O. Khan, L. Kissner, Z. N. J. Peterson, D. Song, Remote data checking using provable data possession, ACM Trans. Inf. Syst. Secur., 14 (2011) 1–34.

[8] A. Juels, B. S. K. Jr. Pors, Proofs of retrievability for large files, in: Proc. 14th ACM Conference on Computer and Communications Security (ACM CCS 2007), pp. 584–597, 2007.

[9] H. Shacham, B. Waters, Compact proofs of retrievability, in: Proc. 14th Annual International Conference on the Theory and Application of Cryptology and Information Security (ASIACRYPT 2008), pp. 90–107, 2008.

[10] G. Ateniese, S. Kamara, J. Katz, Proofs of storage from homomorphic identification protocols, in: Proc. 15th Annual International Conference on the Theory and Application of Cryptology and Information Security (ASIACRYPT 2009), pp. 319-333, 2009.

[11] Q. Wang, C. Wang, J. Li, K. Ren, W. Lou, Enabling public verifiability and data dynamics for storage security in cloud computing, in: Proc. 14th European Symposium on Research in Computer Security (ESORDICS 2009), pp. 355-370, 2009.