ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

Preserving an Efficient Data Integrity in Cloud Computing By Using Merkle Hash Tree Algorithm

S.Sneha
Department of Software Engineering, Information Technology,
Sridevi Women's Engineering College.

Dr.K.Ramakrishna
Department of Software Engineering, Information Technology,
Sridevi Women's Engineering College.

Y.Gnyanadeepa
Department of Software Engineering, Information Technology,
Sridevi Women's Engineering College.

Abstract:

Cloud Computing has been seeing as yefuture-generation architecture of IT endeavor. It moves the application software and databases to the centralized immensely colossal data centers, where the management of the data and accommodations may not be plenarily trustworthy. This unique paradigm establishes many incipient security challenges, which have not been well understood. This work studies the quandary of ascertaining the integrity of data storage in Cloud Computing. In particular, we consider the task of sanctioning a third party auditor (TPA), on behalf of the cloud client, to verify the integrity of the dynamic data stored in the cloud. The prelude of TPA eliminates the involution of the client through the auditing of whether his information put in ye cloud is surely intact, which can be consequential in achieving economies of scale for Cloud Computing. The fortification for data dynamics via the most general forms of data operation, such as block modification, insertion and effacement, is withal a consequential step toward practicality, since accommodations in Cloud Computing are not constrained to archive or backup data only. While prior works on ascertaining remote data integrity often lacks the fortification of either public audit ability or dynamic data operations, this paper achieves both. We first name yetroubles plus-possible security quandaries of direct extensions with plenarily dynamic data modifies from prior works plus then express how to build an graceful verification scheme for the seamless integration of these two salient characteristics in our protocol design. In particular, to achieve efficient data dynamics, we ameliorate the subsisting proof of storage examples by controlling yeclassical Merkle Hash Tree construction for block tag authentication. To fortify efficient handling of multiple auditing tasks, we further explore the technique of bilinear aggregate signature to elongate our main result into a multi-utilizer setting, where TPA can perform multiple auditing tasks simultaneously. Extensive protection plus functioning analysis express that the proposed schemes are highly efficient and provably secure.

Keywords:

Information storage, public auditability, information dynamics, cloud computing.

1.INT RODUCTION :

Cloud Computing:Cloud computing is innovation that utilizes advanced computational power and amended storage capabilities. Cloud computing is a long stargazed imagination of computing public utility, which enable the sharing of accommodations over the cyber world. Cloud is an immensely colossal group of interconnected computers, which is a major vicissitude in how we store information plus run application. Cloud computing is a shared out pool of set up computing resourcefulness's, on-demand network access and provisioned by the accommodation provider [1]. The advantage of cloud is cost savings. The prime disadvantage is security. Cloud computing is utilized by many software industries. Since the security is not provided in cloud, many companies take over their alone protection structure. The data placed in the cloud is accessible to everyone, security is not ensured.

To ascertain security, cryptographic techniques cannot be directly adopted. Sometimes the cloud accommodation provider may obnubilate the data corruptions to maintain the reputation. To eschew this quandary, we introduce an efficacious third party listener to audit the utilizer's outsourced information when needed. In order to solve the quandary of data integrity checking, lots systems are suggested below dissimilar schemes plus protection models [2]–[1]. In all these works, great efforts are made to design solutions that meet sundry requisites: high scheme efficiency, stateless verification, unbounded utilization of queries plus retrievable of information, etc. Considering the role of the verifier in ye model, total the systemintroduced afore fall into two categories: private audit ability and public audit ability. Albeit schemes with private audit power can reach higher system.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

efficiency apartment, public audit ability sanctions anyone, not just the client (data owner), to challengeable ye cloud server for rightness of information storage while keeping no private information. Then, clients are able to delegate the evaluation of the accommodation performance to an independent third party auditor (TPA), without devotion of their calculation imaginations. In ye cloud, the users themselves are unreliable or may not be able to afford the overhead of playing frequent unity assures. Thus, for practical use, it seems more rational to equip the verification protocol on public auditability, which is awaited to play a more paramount role in achieving economies of scale for Cloud Computing. Furthermore, since efficiency condition, the outsourced data themselves should not be required by the verifier for the verification purport. Third party Auditor (TPA):Third Party Auditor is scarcely inspector. There are 2 classes: private auditability plus public auditability. Albeit private auditability can accomplish more prominent system efficiency, public auditability approves anybody, not just ye client (data owner), to challenge the cloud server for the correctness of information storage as continuing no private information. To let off the encumbrance of management of data of the data owner, TPA will audit yeinformationfrom client. It rejects the involution of the client by auditing that whether his informationput in the cloud are indeed intact, which can be paramount in achieving economies of scale for Cloud Computing. The released audit study would avail owners to measure ye jeopardy of their subscribed cloud data accommodations, plus it will withal be good to the cloud adjustment provider to ameliorate their cloud predicated accommodation platform. Therefore TPA will avail information owner to determine that his informationassafety in the cloud and management of data will be facile plus less constraining to information owner.

2. RELATED WORK:

Data Privacy and Verification in cloud have been handled extensively in lots subsisting brings. On surveying the field of public audit ability it is evident that considering the third party auditor as ye vulnerably sensitive element is not treated anyplace. The precedent works do not address totally yeprotection threats plus are totally fixing on single server scenario. Most of them do not consider dynamic data operations plusyedilemma of strengthening some public audit ability and dynamism have been recently addressed where the information is vulnerably sensitive in ye hands of third party auditor.

The following are some cognate papers in the field of public audit ability in cloud.

2.1 Remote Informationownership at untrustedshops:

In thispaperstates that cloud storage can achieve the goal that acquiring totally storage imaginations in a plug-plus -play way, it becomes a focus of attention. When users store their data in cloud memory,theylargely business about whether yeinformation is intact. This is the destination of remote data possession checking systems. This paper suggests an effective RDPC system which has several advantages as follows. First, it is effective in terms of calculation plus communicating. Second, it sanctions verification without the desideratum for the challenger to compare against the pristine data. Third, it utilizes only diminutive challenges and replications, plus utilizer's need to shop only 2 secret keys plus various arbitrary numbers. Conclusively, a challenge updating method is proposed predicated on Euler's theorem.

2.2 Public Verifiability for Storage Security:

This work [5] states that by data outsourcing, users can be mitigated from the encumbrance of local data storage and maintenance. It withal eliminates their physical check of storehouse dependableness plus protection, which traditionally has been anticipated by some enterprises and individuals. This unique paradigm establishes many incipient security challenges, which need to be pellucidly understood and resolved. This work studies the quandary of ascertaining the integrity of data storage in Cloud Computing. To ascertain the correctness of data, we consider the task of sanctioning a third party auditor, on place of ye cloud consumer, to verify the integrity of the data stored in the cloud. This scheme ascertains that the storage at the client side is minimal which will be benign for thin clients.

2.3 Public Auditability for Storage Security:

This paper [6] studies the quandary of ascertaining the integrity of data storage in Cloud Computing. It considers the task of sanctioning a third party auditor, to assert the integrity of the dynamic information stored in the cloud. This paper achieves both public auditability and dynamic information procedures.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

It first names the difficulties and potential security quandaries of direct extensions with plenarily dynamic data updates from prior works plus then expresses how to build an elegant check scheme for ye seamless integration of these 2 salient features in our protocol design. Extensive security and performance analysis express that the suggested systems are extremely effective plus provably assure.

2.4. Privacy Preserving Data Integrity Checking:

This paper [7] proposes protocols that sanction a third-party auditor to periodically verify the data stored by an accommodation and avail in returning the information intact to auser. The protocols are privacy-preserving i.e. it never exposes the information messages to the auditor. This solution abstracts the encumbrance of verification from ye customer, relieves some ye customer's plus memory service's trepidation of information leak, plus provides a method for autonomous arbitration of information memory contracts. The solution provides storage accommodation accountability through autonomous, third-party inspecting plus arbitrement. The protocols have three consequential operations, initialization, audit, and extraction, and it primarily fixates on the latter two. For audits, the auditor interacts with theaccommodation to check that the stored information is integral. For extraction, the auditor interacts with the accommodation and customer to check out that the information is integral plus bring back it to the customer.

3. PROBLEM STATEMENT:

3.1 Security Model:

Following the security model defined in [8], we verbally express that the checking scheme is secure if (i) there subsists no polynomial-time algorithm that can cheat ye verifier with non-negligible chance; (ii) there subsists a polynomial-time extractor that can recuperate the pristine data files by carrying out multiple challenges-replications. The user or TPA can sporadically dispute yememory server to ascertain the correctness of the cloud data, and the pristine files can be recuperated by interacting with the server. The authors in [8] additionally define the correctness and soundness of their scheme: the scheme is veridical if the verification algorithm accepts when interacting with the valid prover (e.g., the server returns a valid replication) and it is sound if any cheating server that convinces the client it is storing the data file is authentically storing that file. Note that in the "game" between the adversary and the client, yeantagonist has fully access to yedata stored in the server, i.e., the adversary can play the component of yeprover (server). The goal of the antagonist is to cheat the verifier prosperously, i.e., endeavoring to engender valid replications and pass the informationcheck without being discovered. Our security model has subtle but crucial difference from that of the subsisting PDP or PoR models [2]–[9] in the verification process. As mentioned above, these systems do not believe dynamic information-procedures, plus ye block insertion cannot be fortified at all. This is because yestructure of the signatures is necessitated with the file index information i. Therefore, once a file block is inserted, yecalculation overhead is unaccepted since the signatures of all the following file blocks should be re-computed with the incipient indexes. To deal with this circumscription, we abstract the index information in the calculation of signatures plus use H(mi) as ye tag of block mi in lieu of H(name||i) [9] or h(v||i) [3], so individual information procedure on whatever file block will not impress the others. Recall that in subsisting PDP or PoR models [2], [9], H(name||i) or h(v||i) should be engendered by the client in the verification process. However, in our incipient construction the client has no capacity to calculate H(mi) minus the data. In order to achieve this blockless verification, the server should surmount ye job of computing H(mi) plus then bring back it to the prover. The consequence of this variance will lead to an earnest quandary: it will give the adversary more opportunities to cheat the prover by manipulating H(mi) or mi. Due to this structure, our protection example differs from that of the PDP or PoR models in both the verification plus the information updating procedure. Categorically, ye tags in our scheme should be authenticated in to each one protocol performance early than computed or pre-stored by the verifier (The details will be shown in section III). In the following descriptions, we will utilize server and prover (or client, TPA and verifier) interchangeably.

4.PROPOSED SCHEME:

In this part, we represent our protection protocols for cloud data storage accommodation with the aforementioned research goals in mind. We commence with some rudimentary solutions aiming to provide integrity assurance of ye cloud informationplustalk about their demerits. Then we present our protocol which fortifies public auditability and data dynamics. We withal show how to extent our main scheme to fortify batch auditing for TPA upon delegations from multi users.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

4.1 System Model:

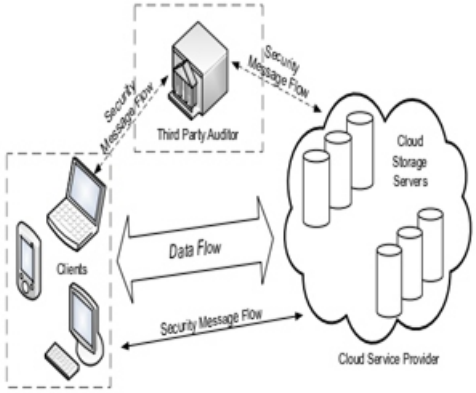


Fig. 1: Cloud data storage architecture

A representative network architecture for cloud data storage is illustrated in Fig. 1. 3unlike network entities can be identified as follows:

• Client:

an entity, which has sizably voluminous data files to be stored in the cloud and relies on the cloud for data maintenance and computation, can be either single consumers or organizations;

• Cloud Storage Server (CSS):

an entity, which is managed by Cloud Accommodation Provider (CSP), has sequential storememoryplus computationalinformation to maintain the clients' information;

• Third Party Auditor (TPA):

Aentity, which holds expertise pluscapacitiesthat clients do not have, is trusted to assess and expose risk of cloud storage accommodations on behalf of the clients upon request. In the cloud paradigm, by putting the astronomically immense data files on the remote servers, the clients can be assuaged of the encumbrance of storage pluscalculation. As user no longer have their information locally, it is of critical consequentiality for the clients to ascertain that their information are being rightlyplacedplusorganized. That is, user should be equipped with certain security denotes so that they can periodically verify the correctness of the remote data even without the esse of local copies.

In case those clients do not compulsorily have the time, feasibility or resources to monitor their information, they can delegate yes upper vising job to a trusted TPA. In this paper, we only consider verification schemes on public auditability: whatever TPA in possession of a public key can act as a verifier. We postulate that TPA is impartial while a server is unsecured. For application uses, the clients may interact with the cloud servers via CSP to approach or recall their pre-stored information. More importantly, in practical scenarios, the client may frequently perform block-level procedures on yeinformationregisters. Ye most natural forms of these operations we conceive in this paper are change, insertion, plusself-effacement. Note that we don't address the issue of data privacy in this paper, as the topic of information privacy inside Cloud Computing is orthogonal to the quandary we study here.

4.2 Merkle Hash Tree:

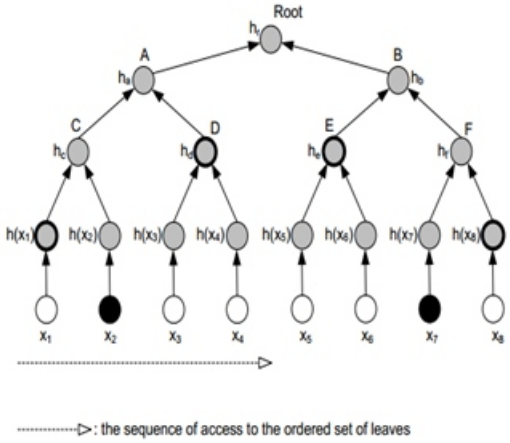


Fig. 2: Merkle hash tree secrecy of informationattributes.We treat the leaf nodes h(x1), . . . , h(xn) as the left-to-right sequence.

A Merkle Hash Tree is a well studied secrecy structure [8], which is proposed toexpeditiously plusfirmly prove that a set of attributes are undamaged plus unaltered. It is built as a binary tree whereas the leaves in ye MHT are the hashes of authentic informationmeasures. Fig. 2 depicts ancase of certification. The verifier on the authentic Hr requests for {x2, x7} and necessitates yecertification of yegotblocks. The prover provides the verifier with the auxiliary hallmark information (AAI) Ω2 =< h(x1), hd> and Ω7 =< h(x8), he >. Ye verifier can then check x2 and x7 by 1 computing h(x2), h(x7), hc = h(h(x1)||h(x2))), hf = h(h(x7)||h(x8))), ha = h(hc||hd), hb = h(he||hf) plushr = h(ha||hb), and then checking if the calculated Hr is equipollent to the authentic one.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

MHT is commonly used to authenticate ye values of informationforgets. Still, in this paper we promote employ MHT to authenticate both the values and the positions of informationforgets. We treat ye leaf nodes as the left-to-right succession, so any leaf node can be uniquely determined by complying this sequence plus the way of calculating ye root in MHT.

4.3 Implementation:

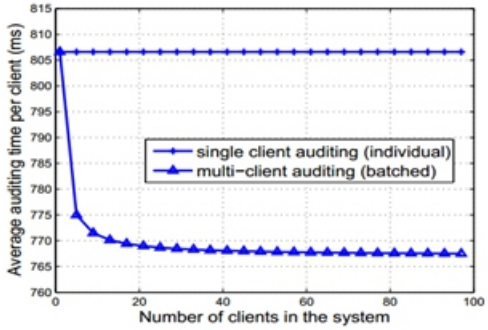
(pk, sk) ← KeyGen(1k). This probabilistic algorithm is run by yeuser. It takes as input security parameter 1k, and returns public key pk and private key sk. (Φ, sigsk(H(R))) ← SigGen(sk, F). This algorithm is run by the client. It takes as input private key sk and a file F which is an injuctively authorized amassment of blocks {mi}, and outputs the signature set Φ, which is an injuctively authorized amassment of signatures {σi} on {mi}.

It additionally outputs metadata-the signature sigsk(H(R)) of the root R of a Merkle hash tree. In our construction, the leaf nodes of the Merkle hash tree are hashes of H(mi). (P) ← GenProof(F, Φ, chal). This algorithm is run by the server. It takes as input a file F , its signatures Φ, and a challenge chal. It outputs a data integrity proof P for the blocks designated by chal.

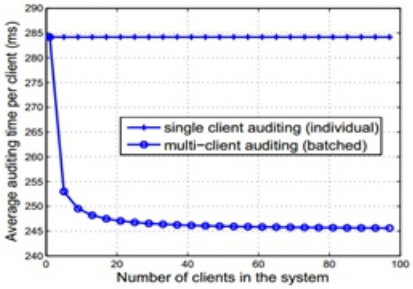
{TRUE, ERRONEOUS} ← V erifyProof(pk, chal, P). This algorithm can be run by either yeuser or ye third party auditor upon receipt of the proof P . It takes as input the public key pk, the challenge chal, plusye proof P returned from the server, and outputs TRUE if the integrity of the file is verified as veridical, or MENDACIOUS otherwise. (F ′, Φ′, Pupdate) ← ExecUpdate(F, Φ, update). This algorithm is run by ye server. It accepts as input a file F , its signatures Φ, and a data operation request “update” from client.

It productions amodifiedfile F ′, updated signatures Φ′ and a proof Pupdate for the operation.{(TRUE, MENDACIOUS, sigsk(H(R′)))} ← VerifyUpdate(pk, update, Pupdate). Such algorithm is work by the client. It takes as input public key pk, the signature sigsk(H(R)), an procedure call for “update”, plus the proof Pupdate from server. If the verification successes, it outputs a signature sigsk(H(R′)) for the incipient root R′, or ERRONEOUS otherwise.

5. EXPERIMENTAL RESULTS:



(a)Tolerance rate ρ is 99%.



(b)Tolerance rate ρ is 97%.

Fig. 7: Performance comparison between individual auditing and batch auditing. The mean per user auditing timeis computed by dividing entire auditing time by the number of clients in the system. For both tolerance rate ρ = 99% and ρ = 97%, the detection probability is maintained to be 99%.

6. CONCLUSION:

To conclude, the problems of trusting a third party auditor in verifying the data can effectively be handled by restricting the access to ye owner’s information. This met informationcheck system isdesigned for such a purpose which restricts the third party to have access to yemeta information to be verified. The verification scheme can further be specialized using protection protocols tocheck ye auditor’s liableness plus confidentiality in handling the data and also can be checked for biasing. To achieve efficient data dynamics, we improvethe existing proof of storage models by manipulatingthe classic Merkle Hash Tree Structuresforblock tag secrecy. To support efficient handling ofmultiple auditing works, we promote research yetechnic of bilinear aggregate signature to extend our main resultinto a multi-user setting, where TPA fire domultiple auditing worksat the same time. Extensive protection plus functioning analytic thinking demonstrate that the proposedsystem ishighly efe and provably secure.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

7. REFERENCES:

[1] [11] Q. Wang, K. Ren, W. Lou, and Y. Zhang, "Dependable and secure sensor data storage with dynamic integrity assurance," in Proc. of IEEE INFOCOM'09, Rio de Janeiro, Brazil, Appril 2009, pp. 954–962.

[2] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song, "Provable data possession at untrusted stores, in Proc. of CCS'07. New York, NY, USA: ACM, 2007, pp. 598–609.

[3] G.Ateniese et al., ―Provable Data Possession at Untrusted Stores,‖ Proc. ACM CCS _07, Oct. 2007, pp. 598–609.

[4] Lanxiang Chen, Gongde Guo, "An Efficient Remote Data Possession Checking in Cloud Storage,"JDCTA: I

[5] Mihir R. Gohel and Bhavesh N. Gohil, "A New Data Integrity Checking Protocol with Public Verifiability in Cloud Storage," Advances in Information and Communication Technology, Volume 374, 2012, pp.240-246.

[6] Q. Wang, C. Wang, et al., "Enabling public auditability and data dynamics for storage security in cloud computing," IEEE Transactions on Parallel and Distributed Systems, vol. 22, 2011 pp.847–859.

[7] Z Hao, S Zhong, and N Yu, "A privacy-preserving remote data integrity checking protocol with data dynamics and public verifiability," IEEE Transactions on Knowledge and Data Engineering, vol. 99,2011.

[8] H. Shacham and B. Waters, "Compact proofs of retrievability," inProc.of ASIACRYPT'08. Melbourne, Australia: Springer-Verlag, 2008, pp. 90–107.

[9] H. Shacham and B. Waters, "Compact proofs of retrievability," in Proc. of ASIACRYPT'08. Melbourne, Australia: Springer-Verlag, 2008, pp. 90–107.

[10] R. C. Merkle, "Protocols for public key cryptosystems," Proc. of IEEE Symposium on Security and Privacy'80, pp. 122–133, 1980.

[11]Pardeep S, Sandeep K. Sood, et al., "Security Issues in Cloud Computing," Communications in Computer and Information Science, Volume 169, 2011, pp.36-45.

[12] Pardeep K, Vivek K.S., et al., "Effective Ways of Secure, Private and Trusted Cloud Computing", International Journal of Computer Science Issues, Vol. 8, May 2011, pp.412-421.

[13] Lanxiang Chen, Gongde Guo, "An Efficient Remote Data Possession Checking in Cloud Storage," JDCTA: International Journal of Digital Content Technology and its Applications, Vol. 5, 2011,pp.43-50.

[14] Mihir R. Gohel and Bhavesh N. Gohil, "A New Data Integrity Checking Protocol with PublicVerifiability in Cloud Storage," Advances in Information and Communication Technology, Volume 374, 2012, pp.240-246.

[15] Q. Wang, C. Wang, et al., "Enabling public auditability and data dynamics for storage security incloud computing," IEEE Transactions on Parallel and Distributed Systems, vol. 22, 2011 pp.847–859.