IJCSE International Journal of Computer Sciences and Engineering Open Access
Research Paper Vol.-6, Issue-8, Aug 2018 E-ISSN: 2347-2693

Cloud Audit Server based verification for enhancing Information Security in cloud

Ramanjaiah Ganji[1], Suresh Babu Yalavarthi [2]

[1]Department of Computer Science & Engineering, Research Scholar, Acharya Nagarjuna University, Guntur. India.
[2]Prof. in Computer Science, Department of Computer Science, J.K.C. College, Guntur, Andhra Pradesh, India.

[*]*Corresponding Author: ramanjaiah@gmail.com, Tel.:9848332853*

Available online at: www.ijcseonline.org

Accepted: 17/Aug/2018, Published: 31/Aug/2018

***Abstract*—** Cloud computing enables internet based data storage, accessing, portability and processing. The flexibility cloud endows comes with a few security challenges. Though plans like "Proofs of Retrievability" and "Provable Data Possession" has been created to ascertain security they can't bolster dynamic information. By and large a significant number of the peril models accept that having a fair data owners are concentrating on the exploitative cloud authority organization. In certain scenarios the client might be untrustworthy for getting the advantages by means of pay from the supplier. This research work focuses on an open inspecting plan that endows information support and reasonable discretion in data safety issues. This paper primarily focuses on developing a signature based plan to configure reasonable discretion conventions that ensures data integrity and security. The security verification establishes that the proposed method is secure and the information flow and dispute arbitration are sensible.

***Keywords*—**Cloud computing, thrid party auditor, Third party arbitrator, Data Verification, dispute arbitration.

I. INTRODUCTION

The utilization of the Cloud computing is to allow users to access applications over internet. It enables any organization to scale up or scale down the utilization of the resources such as storage, an application or a virtual machine. One of the significant features of cloud computing stage independency which gives a user the freedom to use an application or a product without physically installing it in the local machine. This feature brings portability and flexibility to any business application. The use of cloud computing minimizes the weight on the client as it stores and deal with the information effectively. In any case, there are numerous dangers for the information which is put away in the cloud i.e. the information put away in the cloud might be altered by the client or the cloud supplier. Numerous current plans proposed different threat models having the public auditing plans like Provable Data Possession (PDP) conspire [1] and Proof of Retrievability (POR) [2] which gives designation of the auditing scheme to the Third Party Auditor (TPA) with the goal that the weight on the customer is diminished yet they generally center around the deceptive cloud service provider (CSP) and they are considered about the untrustworthy client. Yet, quite possibly the client might be extortion as he need the cash pay consequently from the cloud specialist organization and the Cloud provider may likewise be deceptive as he can sold the transmission capacity to alternate clients by erasing the current documents of the clients for the free storage space. Along these lines, there might be semi-trust on both client and the cloud service provider. These plans for the most part give high computational overhead expenses.

The other methods [3] developed have ability give probabilistic confirmation by reaching to certain portion of the document. This method clearly exhibits effective auditing ability over the already existing plans. Some existing plans like Proofs of Retrievability [4] and Encrypted information confirmation [5] give private undeniable access that requires the data owner to possess a private key in order to fulfill the inspecting assignment, which may be the finished privilege of the data owner because of constrained calculation capacity of the private key. On the contrary other open examining plans allow any user holding a private key to play out the evaluation so that anyone can review the assignment. This process enables an honesty check by an outside TPA and report the result for the benefit of the data owner.

For most of the part PDP and POR are planned to review the static information thus they can't give the data dynamics bolster. In any case, there is a typical prerequisite of the dynamic information on the grounds that the versatility and the practicability are restricted for the static information. Zheng al [6] proposed a reasonable PoR plan to stop a user

from blaming a fair CSP and it isn't fruitful on the account of private auditing. Kupccu [7] proposed general mediation conventions with robotized installments fair signature exchange conventions. So we are utilizing the signature exchange thought for protocol fairness and focus on consolidating powerful information dynamics bolster and sensible debate arbitration into a singular investigating design. So we are introducing a Third Party Authority (TPA) in this model, a trustworthy specialist mechanism for resolving conflicts, paid by both client and the cloud service provider. Notwithstanding these we are embracing the possibility of signature to guarantee metadata rightness.

In this paper we are proposing another auditing plan to address the issues of information elements support and debate intervention at the same time. We give fairness guarantee and dispute arbitration in our plan which guarantees that both the data owner and the cloud can't get rowdy in the evaluating procedure or else it is simple for a third-party arbitrator to discover the conning party.

In this paper we will discover whether CSP or cloud is getting into mischief by contrasting the hash estimations of the put away information. For the most part hash esteems are created for the information which is transferred by the owner. On the off chance that the owner or the Cloud service provider alters the information then the hash estimation of the current information is changed as the information is adjusted. By this the authority will discover the altered information and can discover by whom the information is adjusted i.e., the proprietor or the cloud specialist organization. The rapid escalation of adoption and market perceptions around Cloud necessitates an expedient survey of the possibilities of strengthening the data security aspects. [8] In the Section - II of this paper the related works and the previous works studied are discussed. The Section –III describes the proposed work along with required algorithms. The Section – IV describes of the results obtained and discussion. The Section – V presents the conclusion of the paper.

II. RELATED WORK

Compact proofs of retrievability

In a proof-of-retrievability system (POR) a verifier authorizes the validity of the data the client intends to store in the data center. In this phenomenon the main challenge is to create systems that provide effectiveness and security in handling data. Such system should build processes to extract client's data from any location that passes every step of verification. In this paper, we present the first POR Method with full proofs of security against arbitrary adversaries in the strongest model proposed by Juels and Kaliski. In the first scheme [2], developed as part of this work from BLS Signatures which is secure in the random oracle model, features a POR protocol in which the users query and servers response are both exceptionally short. This method also supports public verifiability: any client, who need not be the file owner, can play the role of a verifier. Our second method, construct Pseudo Random Functions (PRFs) is highly secure in the standard model, which enables private verification. Proof-of-retrievability protocol provides shorter servers response than the first model, but in this model the clients query is longer than the ones in the previous model. These two models depend on homomorphic properties that merge into a small authenticator value.

Official Arbitration with Secure Cloud Storage Application

Secure cloud storage mechanism uses both static and dynamic proof of storage schemes. In this model, a user outsources storage of data to a server where the data may be inadvertently corrupted; data may be corrupted due to hardware or software problems, delete due to unused access parts of etc. Many current schemes can solve only part of the existing problems. The user requests for a cryptographic proof of integrity from the server. But there is no specific solution when the proof fails to verify. We argue that in such a case, both the client and the server should be able to contact an official court, providing cryptographic proofs, so that the Judge can resolve this dispute. We show that this possession is stronger than public verifiability and in the same way the official arbitration should manage a harmful client as well. We clearly show this formalization difference, and then present several methods that work for different static and dynamic storage alternatives in a general way. Our Implemented schemes are very competent, diminishing the validity of argument aligned with their use, where the overhead for adding the ability to resolve such clash at a court is only 2 ms and 80 bytes for each update on the stored data, using Micro computer hardware. As a final point, we show that a clash may arise in several circumstances, such as when two parties exchange items (e.g., e-commerce) or agree on something (e.g., contract-signing). In this paper we present a method that is capable of expanding our official arbitration protocols for a general case, including dynamic authenticated data structures.

III. METHODOLOGY

In this paper we will discover whether CSP or cloud is getting into mischief by contrasting the hash estimations of the put away information. For the most part hash esteems are created for the information which is transferred by the owner. On the off chance that the owner or the Cloud service provider alters the information then the hash estimation of the current information is changed as the information is adjusted. By this the authority will discover the altered information and can discover by whom the information is adjusted i.e., the proprietor or the cloud specialist organization.

Algorithms:

Arbitration on Integrity Proof

Let Sigc = Sigskc (seq; Ω) and Sigs = Sigsks (seq; Ω) Signifies the user and the CSP's signature within the last potent update, where seq alludes to the final sequence number. Proper when a signature exchange trade completes, the consumer has the signature Sigs of the server, and the server has the signature Sigc of the consumer. In the midst of the mediation, seqc and seqs display the progression number sent by the purchaser and the CSP. We renowned the overall public key of each and every social gathering is in some location stock in PKI, as a consequence it can be easily gotten by the opposite get together (checking the TPAR). In addition, all by way of our traditions, we renowned the messages transmitted amongst three parties are in an affirmed cozy channel.

We in the beginning portray the intervention convention of case 1, where the debate just includes proof difference. At the point when the customer finds a disappointment of confirmation check amid a reviewing, he contacts the TPAR to dispatch mediation. Since confirming verification legitimacy needs to access to get hash estimations of tested pieces, and checking signatures, it is important for each party to send the TPAR the most recent update he has kept, alongside the signature marked by the other party.

The arbitration protocol proceeds as follows:

1. The TPAR requests {seqc;Ωc; Sigs} from the client. By then he checks the signature Sigs of the CSP. If it is invalid, the TPAR may repel the client for misbehaving; generally the TPAR proceeds.
2. The TPAR requests {seqs;Ωs; Sigc} from the CSP. At that point he checks the mark Sigc of the customer. In the event that the signature Sigc does not check effectively, the TPAR may rebuff the CSP for acting mischievously; generally the TPAR continues.
3. If seqc= seqs, at that point the TPAR requests from the client the tested set Q that causes debate on prove affirmation and retransmit it to the CSP to run the auditing plan. The CSP forms the confirmation as showed by ProofGen and returns it to the TPAR for check.
4. If there is a jumble in seqc and seqs. The TPAR can verify that the party who gives a tinier progression number is playing out a replay strike, he may repel the cheating party. Specifically, if seqc>seqs, the client is conning by replaying an old check from the CSP; if seqs>seqc, the CSP is cheating by replaying an old signature from the client. In this manner, what ought to be forestalled in the convention are conceivable replay assaults propelled by a malignant party. As we have incorporated an arrangement number in the traded signature for each refresh, we can check whether a replay assault is propelled or not by grouping number match. In the event that the two marks check effectively and their succession numbers coordinate (seqc = seqs) at that point we have Ωc = Ωs.

Arbitration on Dynamic Update

Case 2 and case 3 includes the disappointment of a signature trade in the current round of refresh, so it is important for the TPAR to finish the refresh and signature trade. To achieve this, the effectively traded signatures in the past round ought to be confirmed to continue the current round.

The initial two stages of the convention is the same as that of the arbitration protocol on integrity proof, the TPAR asks for {seqc;Ωc; Sigs} from the customer and {seqs;Ωs; Sigc}from the CSP. In the event that the TPAR finds any invalid signature, he rebuffs the relating party. As per the consequence of arrangement number correlation (seqc and seqs), we isolate the convention into two circumstances.

The sequence numbers match (seqc= seqs).

1. The TPAR asks for the refresh record {seqc +1,op,k, tk1, mk1 ,σk1 ,Q1} from the customer.
2. For block modification and insertion, the TPAR checks the rightness of (tk1,mk1 ,σk1) by confirming e(σk1,g)=e(H(tk1).umk1,v).On the off chance that falls flat, the TPAR may rebuff the customer for bamboozling; generally, the TPAR steady with each other. For piece cancellation, this progression can be overlooked.
3. The TPAR transmits {seqc +1,op,k, tk1, mk1 ,σk1 ,Q1} to the CSP, and requests (μ1,σ1) on the little test set Q1 from the CSP. At that point he checks the legitimacy of μ1 and σ1 as per calculation Proof Verify. In the event that falls flat, the TPAR may rebuff the CSP for denying the refresh; generally, the TPAR continues.
4. The TPAR refreshes the file switcher to Ω1, at that point he asks for and confirms new signatures Sig′c=Sigskc (seqc+1;Ω′) and Sig′s = Sigsks (seqs+1;Ω′)from the two parties. The TPAR may rebuff the parties who sends an invalid signature. On the off chance that the two signatures check, the TPAR advances Sig′c to the CSP, and Sig′s to the customer. The customer and the CSP can continue in the following round of refresh.

The sequence numbers mismatch (seqc≠ seqs).

1. seqc<seqs. The server is bamboozling by replaying an old signature from the customer.
2. seqc> seqs + 1. The customer is conning by replaying an old signature from the CSP.
3. seqc= seqs+ 1. This happens when the CSP gets the client's successful request and decreases to invigorate and send his signature to the client.

There are three potential results here. (I) The invigorate record from the client is invalid, so the CSP decreases to revive and contacts the TPAR for arbitration. (ii) The refresh record from the client is significant, however the CSP responds with invalid check, so the client contacts the TPAR for intercession. (iii) The revive record from the client is authentic, however the CSP dangerously denies the invigorate, so the client contacts the TPAR for prudence. For the foreswearing of revive case (seqc = seqs + 1), it is troublesome for the TPAR to pick which party is in charge of the refresh disappointment. Since each gathering can carry

on malignantly to the next gathering and act well disposed to the TPAR, e.g., the customer can send an erroneous refresh record to the CSP in the current round and send a right refresh record to the TPAR in the accompanying mediation. For this situation, the TPAR just runs the convention of match circumstance (seqc = seqs) to complete the refresh and mark trade in the current round, so the two gatherings can continue with additionally adjusts of inspecting or refresh.

In our arbitration protocol, each party needs to send his signature on the most recent metadata to the next party. All things considered, giving only one party a chance to sign the index switcher and the other party just sign the succession number is likewise plausible, e.g., the customer's mark is Sigc= Sigskc (seqc;Ωc) and the CSP's mark is Sigs = Sigsks (seqs). At that point amid assertion, the customer sends {seqc; Sigs} to the TPAR and the CSP sends {seqs; Ωs; Sigc} to the TPAR.

Luckily, albeit the two parties have the potential plausibility to get out of hand, despite everything we can accept such debate are sometimes, or if nothing else not much of the time. All things considered, each party has some fundamental trust toward the other party, else they can't collaborate together to store and deal with customer's outsourced information. In this sense, the arbitration protocol is to refer conceivable debate, the O(n) correspondence may not be so genuine an issue as far as the administration situated normal for Cloud storage. Then again, the duping gathering ought to be extremely rebuffed to diminish the conceivable outcomes of future bad conduct.

IV. RESULTS AND DISCUSSION

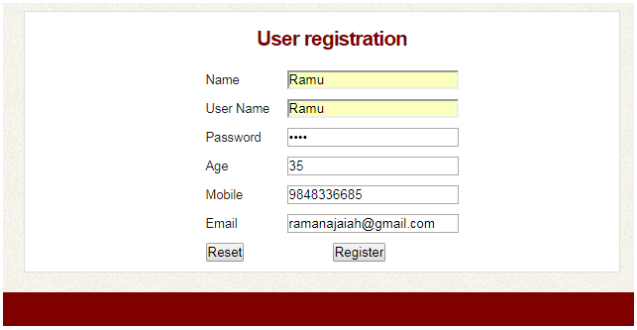


Figure 1: User registration

The user has to register first by using the name, password, age and email. Once the registration is over he/she can login into the application.

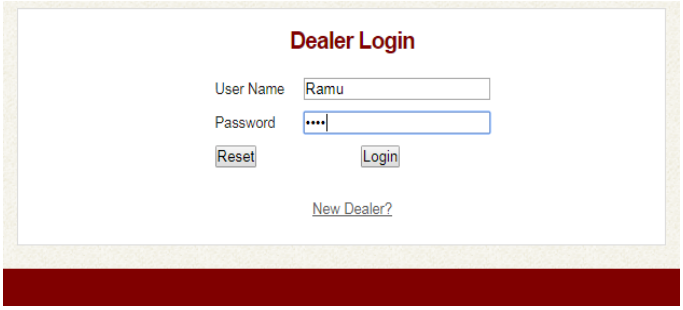


Figure 2: Dealer login

The dealer has to login by using his/her valid credentials.

Figure 3 :Upload file

Once he got login into the application he/she can upload the files that too text files only.

Figure 4 : TPA login

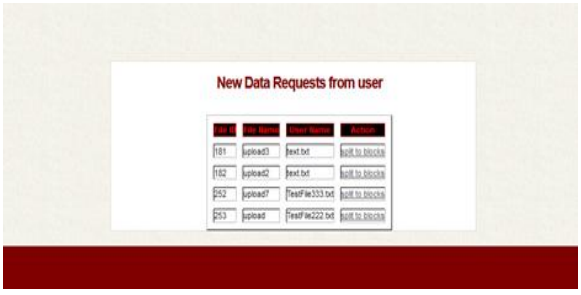


Figure 5: New Data Request

Figure 6: Encrypted Data

The TPA can login into the application by using his/her credentials. They can get the new request login from the user. The TPA can encrypt the data.

Figure 7 : Cloud Storage Server

Figure 8: CSS View Data:

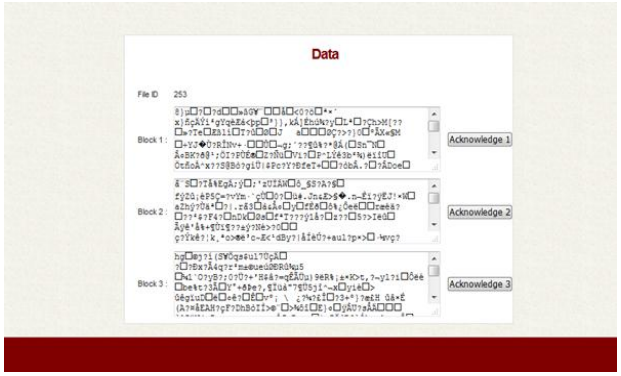
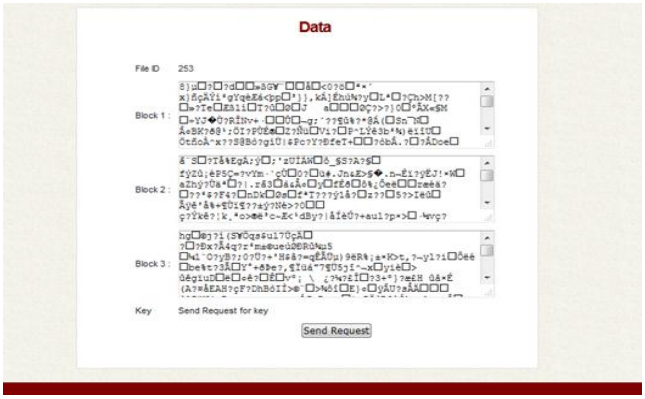


Figure 9: Data

The cloud storage server can login into the application by using his/her credentials. They can view the data and acknowledge the data which is splitted into blocks.

Figure 10: Dealer View Data



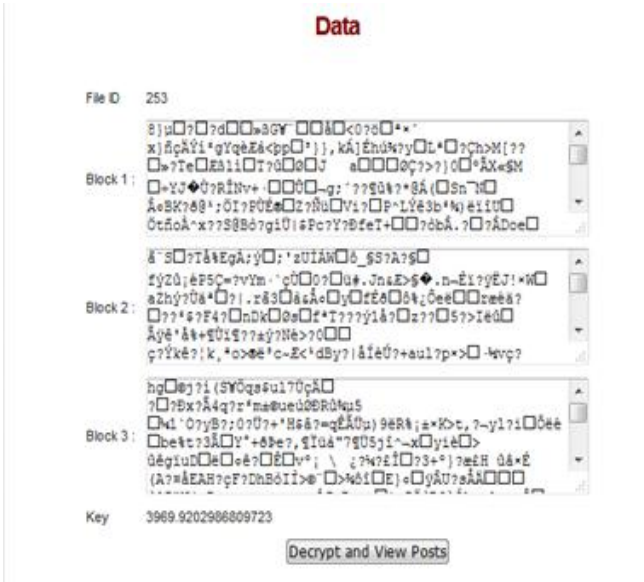


Figure 11 : Dealer View Data

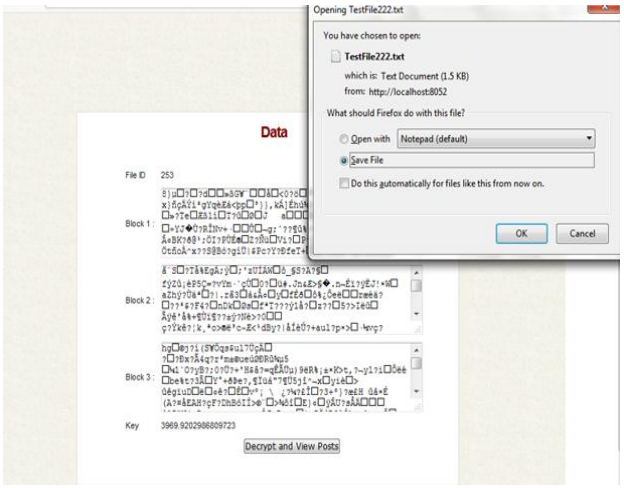


Figure 12: Dealer View Data

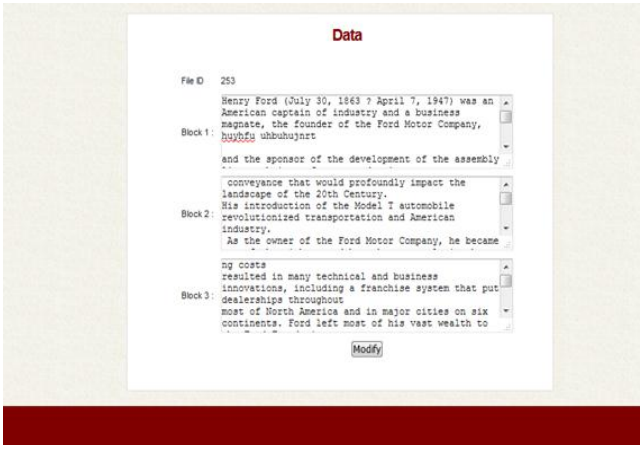


Figure 13 : Dealer Modify Data

The dealer can login into the application and can view the data. He can also modify the data.

Figure 14: TPA New Data

Figure 15: Encrypted Data

Figure 16: CSS View Data

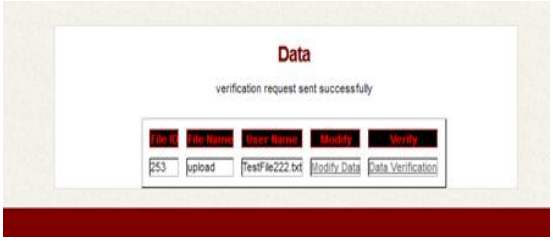


Figure 17: Dealer Verification Request

TPA view the verification requests and can check the data integrity and accept the request

Figure 18 TPA Accept Request:

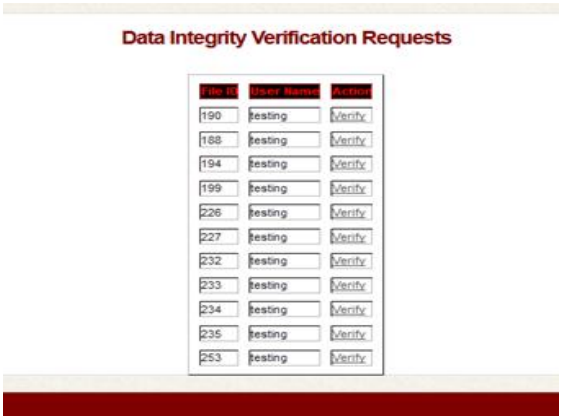


Figure 19 CSS Verify Request

CSS verify the data integrity and can verify whether the data which is splitted in the form of blocks is changed or not.

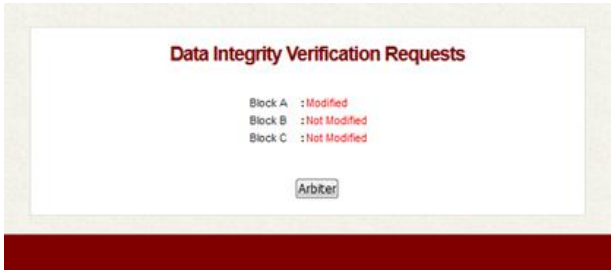


Figure 20: Data Integrity Verification Requests

Figure 21: Verification Status

Figure 22: Verification Status

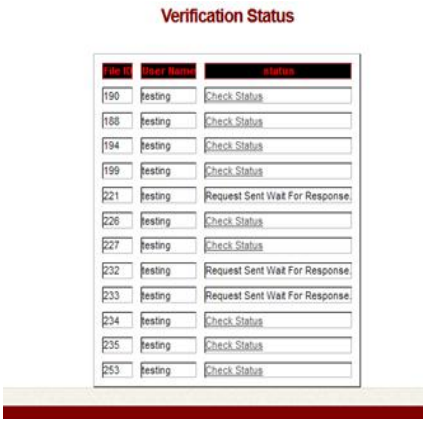


Figure 23: Dealer Verification Request

If the data is modified ten it redirect to the arbitrator who can find out the modification done by whom i.e., the user o r the Cloud service server.

V. CONCLUSION

In this paper we basically cantered around the data integrity and for that we have presented integrity auditing plan with public verifiability and the likewise cantered around the fair arbitration. As quite possibly both the customers and the CSP might be untrustworthy amid the evaluating and the information updation we expand the current risk show in our present research to give reasonable assertion between the users and the CSP. This phenomenon emerges to be a significant development for reviewing plans in the cloud by an organization. The development is achieved by outlining the arbitration protocols. The outlining is achieved by verifying the presence of the mark and by monitoring the hash estimations of the information which in turn can discover the change in the information. The productivity of our proposed conspire is high.

REFERENCES

[1] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song, "Provable data possession at untrusted stores," in Proc. 14th ACM Conf. Computer and Comm. Security (CCS07), 2007, pp. 598–609.

[2] H. Shacham and B. Waters, "Compact proofs of retrievability," in Proc. 14th Intl Conf. Theory and Application of Cryptology and Information Security: Advances in Cryptology (ASIACRYPT 08), 2008, pp. 90–107.

[3] Q. Wang, C. Wang, J. Li, K. Ren, and W. Lou, "Enabling publicverifiability and data dynamics for storage security in cloud computing,"in Proc. 14th European Conf. Research in Computer Security(ESORICS 08), 2009, pp. 355–370.

[4] A. Juels and B. S. Kaliski Jr, "Pors: Proofs of retrievability forlarge files," in Proc. 14th ACM Conf. Computer and Comm. Security(CCS07), 2007, pp. 584–597.

[5] M. A. Shah, R. Swaminathan, and M. Baker, "Privacy-preservingaudit and extraction of digital contents." IACR Cryptology ePrintArchive, Report 2008/186, 2008.
[6] Q. Zheng and S. Xu, "Fair and dynamic proofs of retrievability " inProc. 1st ACM Conf. Data and Application Security and Privacy(CODASPY 11), 2011, pp. 237–248.
[7] A.Kupcu, "Official arbitration with secure cloud storage application", The Computer Journal, pp. 138–169, 2013.
[8] Vishnu Patidar, Makhan Kumbhkar, "Analysis of Cloud Computing Security Issues in Software as a Service" International Journal of Scientifi International Journal of Scientific Research , Vol.2 , Issue.3, pp.1-4, 2014.

Authors Profile

Ramanjaiah Ganji, received Masters in Technology (M.Tech) Degree from Acharya Nagarjuna University, Masters in Computer Science (M.Sc) from the Faculty of Engineering from Bapatla Engineering College affiliated to Acharya Nagarjuna University. Received Bachelor's Degree in Computer Science and Engineering from The Institution of Electronics and Telecommunication Engineers (IETE). Currently he is working as Associate Professor, Department of Computer Science and Engineering, Chebrolu Engineering College, Chebrolu, Guntur affiliated to JNTUK. He has been teaching in the faculty of Computer Science and Engineering since 2008.

Suresh Babu. Yalavarthi, Ph.D. Computer Science in the Faculty of Engineering in 2014 from Acharaya Nagarjuna University, Andhra Pradesh. He Completed his Master Degree in Computer Applications. He is currently working as Professor in the Dept. of Computer Science J.K.C. College, Guntur, Andhra Pradesh. His area of research are Image Processing, Data Mining and Software Engineering. He has published several papers in IEEE journals and various National and International Journals. He is also serving as the Placement Officer of his college.

