IJDCST @Sep-Oct, Issue- V-2, I-8, SW-09
ISSN-2320-7884 (Online)
ISSN-2321-0257 (Print)

Level of privacy preserving in addition to Public Auditing Support for Information Storage employing RSA inside Cloud Processing

[1] A.Siva Sankar, [2] G.J.Suuny Deol
[1] M.Tech (CSE), Sri Mittapalli college of Engineering , Tummalapalem, Guntur (dist).
[2]Asst.Professor, Sri Mittapalli college of Engineering , Tummalapalem, Guntur (dist).

Abstract: The distributed computing model speaks to another standard transformation in web based administrations that conveys profoundly versatile conveyed figuring stages in which computational assets are offered 'as an administration'. Security is viewed as one of the top positioned open issues in receiving the distributed computing model incorporates information Integrity secrecy. Wang proposed an empowering open review capacity and information motion for capacity security in distributed computing. They accomplished the uprightness assurance of information stockpiling with backing of open review capability and element information operations. However their convention needs in giving protection of information which is one of the issue for the cloud information stockpiling. In this we proposed a protection saving open evidence for uprightness of information stockpiling in distributed computing. We are utilizing RSA open cryptography to give secrecy of information. Our plan is more secure than existing framework.

Index Terms: Cloud Computing, Data Confidentiality, Data Integrity, RSA, TPA

Introduction: Cloud model speaks to another standard change in web based administrations that conveys very adaptable disseminated processing plat- structures in which computational assets are provisioned 'as an administration'. This new information stockpiling "Cloud" achieves numerous testing issues which have profound impact on the security and execution of the aggregate framework. One of the greatest concerns with cloud information stockpiling are the information secrecy and uprightness confirmation at untrusted servers at server side. This is for sparing cash and storage room the administration supplier may erase seldom got to documents which fit in with a typical customer. Along these lines, the customers require that their information stay secure over the CSP. Scrambling the delicate information before outsourcing to servers utilizing cryptographic plans can deal with the issue of secrecy. Be that as it may, the honesty of customers information in the cloud may be at danger because of the accompanying reasons: The tradition al cryptographic primitives for information uprightness and accessibility focused around hashing and plans are not relevant on the out- sourced information without having a duplicate of the information. To take care of the issue of information honesty, numerous plans are proposed under distinctive plans and security model.For the most part the model ,categories: private auditability and open auditability. The private auditability attains higher effectiveness and

open certainty permits Third gathering Auditor (TPA) rather than customer (information manager) to test the cloud server for rightness of information stockpiling while not keeping private data. Despite the fact that the current plans expect to give honesty confirmation to distinctive information stockpiling frameworks, the issue of supporting both open unquestionable status and information motion has not been completely tended to.

As of late, Wang etal. proposed Enabling Public Auditability and Data Dynamics for Storage Security in Cloud Computing utilizing Merkle hash tree however they doesn't consider the issue of information privacy, which is one of the paramount issue for cloud information stockpiling.

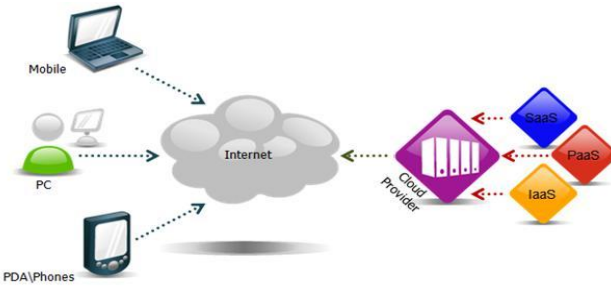

Fig1: Simple architecture of cloud

Related Work: As of late, a large number of developing investment has been sought after in the connection of remotely put away information check. Ateniese et al. are the first to consider open auditability in "provable information ownership" model for guaranteeing ownership of records on untrusted stockpiles. In their plan, they use RSA-based homomorphic labels for inspecting of outsourced information. In their work Ateniese et al. propose an element rendition of the former PDP plan. On the other hand, the framework forces from the earlier bound on the quantity of questions and doesn't help for completely powerful information operations i.e. it just permits just fundamental piece operations with constrained usefulness and square insertions are not backed. In Enabling Public Auditabity and Data Dynamics for Storage Security in Cloud Computing Wang et al. consider dynamic information stockpiling in an appropriated situation and the proposed test is in charge of both to focus the information rightness and to place conceivable blunders. Moreover, open auditability is not underpinned in their plan. Shacham and Waters composed an enhanced Por plan with bunches of verifications of security in the security model characterized in. They utilize freely unquestionable homomorphic authenticators fabricated from BLS marks which is focused around the evidences which might be collected into a little authenticator worth and open retrievability is accomplished. Still, the creators consider just static information records. Erway et al.were the first to actualize the developments for element provable information ownership. They improve the PDP demonstrate in to help provable upgrades to put away information documents utilizing rank-based confirmed skip records. This plan is basically a completely alterable form of the PDP result. All the current plans are go for giving honesty confirmation to distinctive information stockpiling frameworks, the issue of supporting both open undeniable nature and information progress has not been completely tended to. As of late, Wang et al. proposed Enabling Public Auditability and Data Dynamics for Storage Security in Cloud Computing utilizing Merkle hash tree. Their plan attained the honesty of information in cloud with backing of open undeniable nature and

element information operations. Then again, their plan don't consider the secrecy of information put away in cloud.

System Model: The delegate system structural planning for distributed storage is shown in Fig 2.three diverse system substances might be distinguished as takes after:

Client: who saves the information in the cloud that could be either endeavor or individual clients.

Cloud Server (CS): a substance which is overseen by the cloud administration supplier (CSP) to give information stockpiling administration having huge storage room and processing assets

Outsider Auditor: who has skill and abilities that clients might not have, is trusted to survey and uncover danger of distributed storage benefits for the clients upon solicitation.

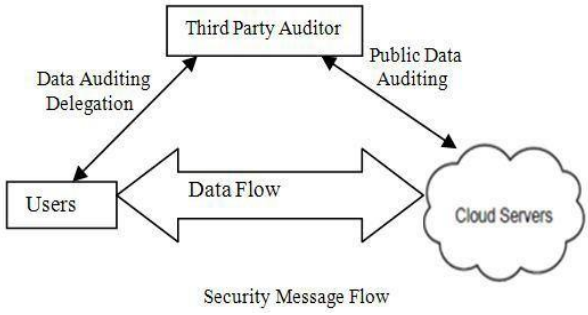

Fig 2: Cloud storage system architecture

Threat Model: The dangers can originate from two separate sources: inward at- tacks and outside assaults. For interior assaults, a CSP might act naturally intrigued, untrusted and perhaps vindictive. Not just it longing to move information that has not been or is infrequently gotten to a lower level of capacity than concurred for fiscal reasons, yet it might likewise endeavor to shroud an information misfortune episode because of administration mistakes, disappointments and so on. For outer assaults, information uprightness dangers may originates from pariahs who are past the space of CSP.

Design Goals:

1. Capacity trustworthiness: to guarantee clients that their information are put away properly and kept in place all the time in the cloud.

2. Classifiedness: giving security to the information by utilizing encryption procedure, for example, RSA.

Preliminaries and documentations:

- F- the data file to be stored , which is denoted as a se- quence of n blocks m1….mn ε Zq for some large prime q.
- fkey(.)- PseudoRandom Function (PRF) defined as: {0,1}*× key→Zq.
- πkey(.)-PseudoRandom Permutation (PRP) defined as:{0,1}*× key→{0,1}log2(n).
- H1(.),H2(.)→map to point collision-free hash functions defined as:{0,1}*→G, where G is a group.

Bilinear Map:- let G1, G2, and G be multiplicative cyclic groups of prime order q and let g1 and g2 be generators of G1 and G2 respectively and e be a bilinear map if e:G1×G2→G is a map with following properties:

Computable: There is a polynomial computable time algorithm to compute e(u,v) ε G1 for any (u, v) ε G.

Bilinear: For all all $u \in G_1$, $v \in G_2$ and $x, y \in Z_q$

$e(u^x, u^y) = e(u,v)^{xy}$

Non-degenarate: If g1 is a generators of G1 and g2 is a ge- narators of G2, then

$e(g_1, g_2) \neq 1$

For any $u_1, u_2 \in G_1$, $v \in G_2$

$e(u_1, u_2, v) = e(u_1, v), e(u_2, v)$

Merkle Hash Tree (MHT): A Merkle Hash Tree(MHT) is a data structure[17], which is used to prove efficiently and securely that a set of elements are not damaged and not altered. It is binary search tree, where each of the leaf node contains hash value of authenticated data. While MHT is commonly used to authenticate the hash values of data blocks however, in this we further employ MHT to authenticate both their values and the positions of data blocks and compute the root in MHT.

Definitions: The proposed scheme follows: KeyGen(1k)→(pk,sk)- is a random key generation algorithm that is run by the client to setup the scheme which takes a large security parameter k as input and produces a public/private key pair (pk,pr) based on RSA .

Enc (F)→F'. The Client uses this algorithm to encrypt the un- processed file F with the seal key ek and encode it.

SigGen(pk,sk,m)→

σ_i is a (possibly random) algorithm run by client to generate verification of metadata which are signa- tures. It takes public key pk, secrete key sk and file block(m) as inputs and produce metadata as output i.e σ_i .

GenProof(pk,F',Q,Ø)→P is run by cloud server to generate integrity proof of data storage. It takes public key pk, file F', signatures Ø, and challenge query Q as inputs and produce output P, where P= (μ, σ)

VerifyProof(pk,chal,P)→{0,1} - This algorithm runs by TPA to validate the Proof of integrity from cloud server which takes public key pk, challenge query Q , and proof P and return output as 1 in case of the integrity of file is verified as correct otherwise 0.

ExecUpdate(F',Ø,update)→(F'',Ø',Pupdate).This algorithm run by server, it takes file F', signatures Ø, and a data opera- tion request "update" form user and produce updated file F'',

updated signatures Ø and a Proof Pupdate for the operation.

VerifyUpdate(pk,update,Pupdate)→this algorithm run by TPA. It takes public key pk, an operation request "update" and proof Pupdate from server if the verification success, it outputs 1 otherwise 0.

RSA Algorithm:

The RSA algorithm involves three steps: key generation, encryption and decryption.

Key generation: RSA involves a public key and a private key. The public key can be known to everyone and is used for encrypting messages. Messages encrypted with the public key can only be decrypted in a reasonable amount of time using the private key.

Proposed System: To ensure the integrity and Confidentiality of data storage we propose a new data verification protocol. It is designed based on merical hash functions. Our protocol consists of three phases:

1) Pre-processing
2) Data Integrity checking

Pre-Processing: The processing of this as follows: first, it generates the public and private key pair using Keygen function and then client encrypts the file using encryption function. After encrypting the data file, it computes the signatures over the encrypted file using SinGen algorithm. Finally, the client sends Signatures and public key TPA. Algorithm of Pre-processing is

Algorithm 1: Pre-Processing:

1 Client generate private key : g,
2 generate public key gβ
3 Encrypt the file {F} g
4 Create the root for MHT . {H(R)} ,{dk}
5 Sign the root sig(H(R))=H
6 Compute Signature f'{m'} {} where 1<i<N [H(m)]

Verification phase: In this the TPA verifies the integrity of data as follows: first, the TPA creates a challenge and sends the server. After receiving a challenge, the server computes response and returns back to the TPA then TPA checks the integrity comparing response with signatures.

The detailed verification phase is given in algorithm 2.

Algorithm 2 verification Phase:

1: Procedure
2: generate the challenge query Q={i,Vi}
3. send {Q, pk}iεI to Server
4: for each query server calculates

And

□□□□□□i□I Vi mi

□□G1

where □□i □□(H1 (i) u) □□G1 (i □□1.....,n)

5: server sends {F',σ,T,} to TPA

6: TPA verify

7: return 1

8: else

9: return 0

10: end if

11. end procedure

Security Analysis: In this section, we analyze that proposed scheme is more se- cure than existing schemes against data leakage and data loss/damage. Our proof consists of two parts: Integrity and Confidentiality.

Integrity of data storage guaranty: We need to prove that cloud server cannot generate valid response towards TPA without faithfully storing the data file.

Theorem 1: If cloud server passes the verification phase, then it must indeed possess the particular data stored correctly or not.

Proof. This theorem consists of three steps

1. Initially we show that there exists no server that can forge a valid response {σ, μ, T} to pass the verification using equation (1). The integrity of this statement follows from theorem available in 4.2[8]
2. Now, we show that if the response from {σ, μ, T} is valid, where μ=μ'+tH2(T) and T=(v2)t, then the important sam- ple blocks in μ' must be valid. This is obtained immedi- ately by verifying the response using equation (2)
3. Finally, in our scheme should detect all data corruptions if data has been corrupted during the verification phase. Assume that if server corrupts some of the blocks {mij}, in μ', where μ'=m'j. This is achieved by checking response with previously computed signatures using equation (3). If not. It indicates that data has been corrupted.

Theorem 2: The stored data cannot be leaked to unauthorized parties

Proof: We prove that theorem in two steps: first, we show that no information on μ' can be learned from μ, this is be- cause the file is encrypted by using RSA-algorithm where p and q values are chosen randomly and large. If attacker try to access a encrypted file, he need private key. If tries to get the private key by using public, however, it is impossible due RSA assumption. Therefore, according to our analysis, an adversary cannot get anything from

encrypted file. Hence, it is proved that proposed scheme is more secure against data leakage.

Execution investigation: According to the algorithm 1 and 2, we can demonstrate the overall workload of the computing and storage of each parties in our scheme as followed

Client: who stores the private keey and decryption key of file.

TPA: stores the user's public key g, encrypts/decrypts the file, computes the data blocks signature collection Φ, and verifies the both the equations during verification.

CSP: stores the signature Sigsk(H(R)), the encoded F' and the Φ; generates the verification information μ and ω and then computes the Ωi for recovering the MHT.

CONCLUSION:

This approach is very secure. In order to attain data integrity we are verifying through are using Third Party Auditor (TPA) on behalf of cloud client to verify the integrity of data stor- age using Merical Hash Tree i.e all the verification is done by using TPA only which reduces the cost as well as users burden and thus providing cost effective method to the user. The confidentiality is attain by using RSA based cryptography algorithm which computes very fastly. In this, the out sourced data is encrypted with public key and made into cipher text and at the client side the user decrypt the encoded data us- ing private key. Hence providing integrity and confidentiality to the user's data.

REFERENCES

[1] C. Wang, Q. Wang, K. Ren, and W. Lou, "Ensuring data storage security in cloud computing," in Proc. of IWQoS'09, July 2009, pp. 1–9.

[2] Amazon.com, "Amazon web services (aws)," Online at http:// aws.amazon.com/, 2009.

[3] 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, Alexandria, VA, October 2007, pp. 598–609.

[4] Amazon.com, "Amazon s3 availability event: jul2008,"Online http://status.aws.amazon.com/s3-20080720.html, July 2008.

[5] Balkrishnan. S, Saranya. G, Shobanas and Karthikeyan .S, "Introducing Effective Third Party Auditing (TPA) for Data Storage Security in Cloud",International Journal of computer

[6] Maha TEBAA, Saïd EL HAJJI, Abdellatif EL GHAZI "Homomorphic Encryption Applied to the Cloud Computing Security" Proceedings of the World Congress on Engineering 2012 Vol I WCE 2012, July 4 - 6, 2012, London, U.K.

[7] K Govinda, V. Gurunathprasad and H. sathishkumar, "Third Party Auditing for Secure Data Storage in Cloud Through Digital Signature

A.Siva Sankar received my B.Tech degree in CSE from BVC institute of technology of Sciences, in 2011.At present pursuing M.Tech Degree in Computer Science from Sri Mittapalli college of engineering in Tummalapalem, Guntur (Dist).

G.J.Sunny Deol - his qualification is M.tech, B.Tech.

And he has 5 years experience. He was interested in c-lanugauge, DBMS, Dataming, Data Structures, Mobile Computing. Presently working as Asst.Professor in the department of CSE at Sri Mittapalli College of Engineering, Tummalapalem, Guntur (Dist)