ISSN XXXX XXXX © 2017 IJESC

Research Article Volume 7 Issue No.10

Enabling Non Vulnerable Data Possession on Remote Storage Cloud Servers to Obtain Cloud Data Integrity and Eminent Data Dynamics

Kumar .M[1], S. Baskaran[2]
Student[1], Associate Professor[2]
Department of Computer Science and Engineering
Kuppam Engineering College, Chittoor, India

Abstract:
Cloud is technology where burden of the server is reduced and performance bottleneck is improved. In cloud environments, cloud servers are breach-able, completely not trustworthy and it's a tedious process for data security hence it's very important for data owners to trust the possession of data which has been done to remote cloud servers on outsourced data files. The ideal goal of technical work is mainly focused on giving permission for protocol to check the remote data possession with the help of third party verifiers on access to uncorrupted file that the remote server can access. The system designed also helps in auditing the cloud storage along with very lightweight communication and computational cost and audited result simultaneously make sure of guaranteed storage correctness including fast localization of erroneous data (identification of unauthorized remote server).

Keywords: Cloud Computing, Data dynamics, Cloud service providers, third party auditor, cloud attacks, non-vulnerable data, data integrity, eminent data

I. INTRODUCTION

Cloud environment is merely most essential for fast growing technology which usually help to reduce computational aspects and the computational cost aspects. The cloud environment can be based on different services like infrastructure as a service platform as a service and software as a service and usage mainly depends on the type of service you're opting for the environment. The cloud can also be distinguished based on where exactly it exists i.e. it can be public cloud, private cloud, hybrid cloud and community cloud.

II. LITERATURE SURVEY

Author Q. Wang at all and Erway at all worked on "Enabling public verifiability and data dynamics for storage security in cloud computing" and "Dynamic provable data possession" respectively. The valuable inputs they suggested for the cloud environment actually optimized the security and integrity constraints of any typical model of data encryption standards. The opted work is to cope up the security features which are intended to enable the dynamics of data integrity while data is possessing throughout the network and main concern is to assure the data possessed to cloud server is actually designated. Although the formal analysis and security correctness is present. The proposed scheme is highly robust on cloud server attacks and failures as well malicious data resilient and the designated works also supports secure and dynamic operations on outsourced data.

III. EXISTING SYSTEM

The Existing cloud system make use of three objects
1. User
2. Cloud server and
3. Cloud service provider

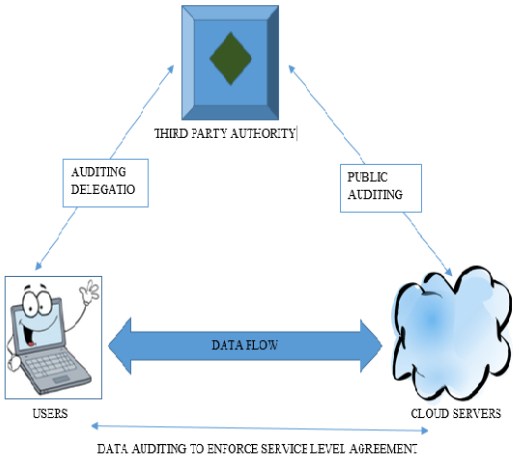

Figure (1)

The objectives of each object is the amount of data or information to be stored, management of each services interaction with cloud server by cloud service provider and provide data storage services respectively. The 3rd party auditor has expert auditing capabilities that usually cloud server doesn't have and is trusted authority to assess the security of cloud storage service behalf of request made by user. Here users mainly relay is on cloud server for maintenance and storage of cloud data. Hence third party auditor must be reliable and work independently and should not conspire either with cloud server or with the users during auditing process and the audit should happen without the aid of local copy which avoids the burden of users supposed to be taken. The possible leakage of any outsourced data from users through auditing protocol should be prohibited by 3rd party authority. To achieve delegation of audits the diligent cloud server supposed to respond to 3rd party audits on time and user must provide certificate audition granting rights to 3rd party public key meanwhile all audits from 3rd party authenticated against certificate for threats which arrives from external attacks and

also ensure data integrity from outsiders who are beyond the control domain of cloud service providers.

IV. PROPOSED SYSTEM

The proposed paradigm sure of very flexible and explicit data dynamics support which includes deletion insertion and modification of data and desired goal is to give protection against untrusted service providers. The adaption of proposed protocol for distributed verification of erasure data we need to cope up to achieve public verification and data dynamics against 3rd party verifiers in distributed servers on various operation like corrupted data deletion during the storage correctness validation.

The designed protocol which follows security and performance measures are:

1. Audition by public
2. Ensures storages correctness
3. Eminent privacy
4. Group auditing
5. Diaphanous.

Users on the screen are:

1. ***Data owners:*** stores the data in cloud and send each share of data entries to the service providers

2. ***Data users:*** access the data from service providers as well have privileged access to public information of data owners to makes sure the shares received from service providers are trusted.

3. ***Cloud service provider:*** Acts as a mediator between Data owners and Data users and provides data storage services.

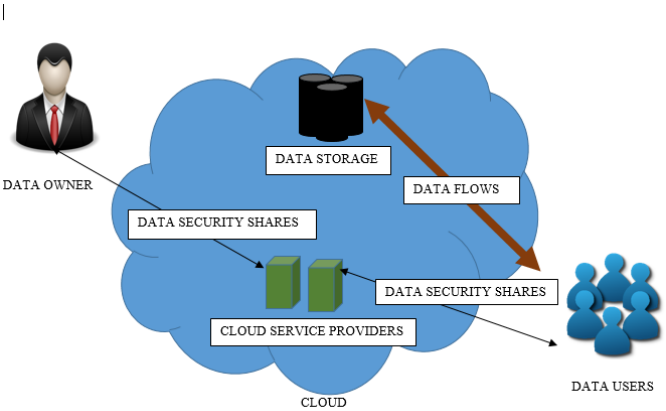

Figure (2)

Here malfunctioning server will be easily located when data corruption is detected by having fast localization of data errors.

V. PROTOCOL EVOLUTION

Usually in every cloud environment there would be a number of clients and servers involved in interactions here importance of the server is storage meanwhile every client will store the data in the server but there's a probability of not having any back up in there's client machines so to assist this crucial problem it's important for every client to check the integrity of the information they store at cloud servers which are not trustworthy. Hence we need an aid called third party for data verification as long as clients and cloud storage servers are concerned. The proposed approach must sense correct when server passes the data integrity request as long as trustable client and server concerned and designed approach is secure against the untrusted server when and while proposed protocol is guaranteed on trusted clients if only unsusceptible cloud storage server has complete access to uncorrupted data in order to send data integrity request to authorized clients and important aspects of our approach is protocol privacy against third party verification. The validation of remote data integrity involves the 5 major activities namely

1. SetUp
2. TagGen
3. Challenge
4. Gen-Proof
5. Check-Proof

Let f be the file that is uploaded and stored in the untrustworthy cloud server and is divided into equal lengths of n blocks: f = f1, f2 ..., fn, where n = Γ |f|/ 1 ⅂. Here 1 would be the length of each file block. We denote fp (•) a pseudo-random function defined as: f: {0, 1} p × {0, 1}log2(n) → {0, 1} d, and we call p and d as two security parameters. Further we say length of N in bits by |N|.

VI. FUNCTIONS ELABORATION

SetUp $(1^P) \rightarrow$ ***(Pp, Sp):*** This function generates the public key Pp and the secret key Sp where Pp is public to everyone whereas Sp is key kept secretly by the client.
(Given p-> Security Parameter)

TagGen (Pp, Sp, f) $\rightarrow$ D_f***:*** This function computes a verification and validation tag D_f which is known to everyone publicly and used for public verification of data integrity.
(Given Pp, Sp and f)

Challenge (Pp, D_f***)*** $\rightarrow$ ***chlng:*** This function is used by the verifier to generate the challenge chlng request as a integrity proof of file f which is send to sever.

GenProof (Pp, D_f***, f, chlng)*** $\rightarrow$ ***R:*** This function is used by the server to compute a response R to the challenge chlng as well R will be send back to the verifier for verification.

CheckProof (Pp, D_f***, chlng, R)*** $\rightarrow$ ***{"success", "failure"}:*** This function returns the success or failure status on validation and mainly used for data dynamics. Here Verifier validates the response R. If it's valid "success", if isn't "failure" meanwhile secret key Sp is not required for CheckProof function.

CheckMisbehave (r, enfle, f') $\rightarrow$ ***n:*** This function is used by the verifier who can detect the unusual behavior of server and if none of the rows specified in the process are deleted or modified the dispute will avoid the detection

Given (f: Matching factor, enfle: Encrypted file matrix, r: Number of rows user asks for checking) We have few security requirements analysis for protocol checking of remote data possession and we also ensure an eminent algorithm to perform file retrieval and error recovery from the actual affected servers:

1. Providing the security against server with verifiable public
2. Solitude against 3rd party verification
3. Identification of untrusted server
4. File retrieval process and recovery of errors

The protocol makes sure that none of the private confidential information of client's data is revealed to that 3rd party verifier when actually the verifier is not client itself.

1. Providing the security against servers which are publicly verifiable:

If CheckProof (Pp, D_f, chlng, R) = "success", then we must say the protocol is secure against server probabilistic time of polynomial for remote data possession checking.

2. Solitude against 3rd party verifier's verification:

If there exists a probabilistic time of polynomial PTP and simulator SR such that {SR (a, fR (a, b))} a, b € {0, 1} * ≡ {view ΠR (a, b)} a, b € {0, 1} *, then we call Π is the protocol that guarantees privacy against 3rd party verifiers verification, where the symbol (≡) denotes indistinguishability of computation.

3. Identification of untrusted server:

Assume due to compromise failure n servers are not behaving as expected and adversary modifies the blocks of data in X rows out of the one row in matrix encoded file. Consider r be number of divergent rows that user would like to asks for challenge checking and choose N as random discrete variable which is defined for the numbers of rows selected by the user that matched the rows adversary modified. And we can say matching probability at least single row is picked up by the user is analyzed first on matching one of the row modified by the adversary.

P^r m'= 1-P {N=0} =1− $\Pi^{r-1}_{i=0}$ (1-min {(x^) / (l-i), 1}) ≥ 1- ((l^m –^{x})/l) ^{r}

In the i^{th} verification none of the r rows are modified or deleted the adversary will avoid the detection whereas this process can be done by comparing the repose values $R_i^{(j)}$ along with predefined tokens $V_i^{(j)}$ where j € {1, . . ., n}

The localization of errors and misbehaving server's identification probability is computed in similarly. $\hat{}P^r$ m'= 1− $\Pi^{r-1}_{i=0}$ (1-min {(x^) / (l-i), 1})

Where z^ ≤ z is the matching probability, $\hat{}P^r$ m' is the probability of matching the modified rows in matrix of encoded file.

Pr^f (false negative probability) is considered such that $R_i^{(j)}$ = $V_i^{(j)}$ when at least a single block of z^ is modified.

Suppose if two dissimilar vectors of data collided, the given probability = ^ Pr^f =2^{-p}

Hence probability of identification for misbehaving servers is $\hat{}P_d$ = $\hat{}P^r$ m'. (1 − $\hat{}Pr^f$)

Here P_d is the probability of detection against data modification.

The computation of above formula is for misbehaving servers' localization which are integrated to check the remote data integrity which actually goals the more efficient and secured protocol.

4. File retrieval process and recovery of errors:

Whenever each and every time corruption of data is detected we ensure the identification of misbehaving servers by comparing the precomputed tokens with the received response values.

VII. ALGORITHM

1. Assume corruption of blocks has been detected among the r rows specified where assume s ≤ p are the identified misbehaving servers
2. Do download all r rows blocks from the servers;
3. Consider s servers as erasures and p as security parameter to recover the blocks
4. Try to resend the all blocks recovered to corresponding servers

5. End of procedure

Therefore, always user ask servers to send back blocks specified r rows in the challenge and always regenerate the accurate correct blocks by ensuring erasure correction as long as the identified number of misbehaving servers is < p. The blocks which are newly recovered will be eligible for misbehaving servers' redistribution to obtain the storage correctness.

VIII. CONCLUSION & FUTURE WORK

This data possession checking work is a privacy preserving protocol for data storages which has been done remotely. The main focus here is on preventing the data being disclosed by untrustworthy service providers whenever the owners of the data distribute the database entries of their own along with error recovery. The desired goals like availability, data integrity and quality of storage service for cloud users can be achieved with an eminent, flexible, effectively distributed and some exclusive data dynamics must support including updating, deletion and insertion of blocks data. The further innovation may aim to enhance the protocol work mainly to support dynamics at data level which is cost effective to cope the trusted transaction of data.

IX. REFERENCES

[1]. P. Mell and T. Grance, "Draft nist working definition of cloud computing," Referenced on June. 3rd, 2009 Online at http: //csrc.nist. gov/groups/SNS/cloud-computing/index.html, 2009

[2]. S.Wilson, "Application engine outage, "Online http:// www. cioweblog.com/50226711/appengineoutage.php, June 2008.

[3]. Q. Wang, C. Wang, J. Li, K. Ren, and W. Lou, "Enabling public verifiability and data dynamics for storage security in cloud computing," in Proc. of ESORICS'09, Saint Malo, France, Sep. 2009.

[4]. M. A. Shah, M. Baker, J. C. Mogul, and R. Swaminathan, "Auditing to keep online storage services honest," in Proc. of HotOS'07. Berkeley,CA, USA: USENIX Association, 2007, pp.1–6.

[5]. Erway, A. Kupcu, C. Papamanthou, and R. Tamassia, "Dynamic provable data possession," in Proc. of CCS'09, 2009, pp. 213–222.