RESEARCH ARTICLE OPEN ACCESS

Data Integrity Checking Protocol with Dynamic Public Verifiability

M.D.Boomija, P.V.Pooja, K. Preethi
Prathyusha Engineering College, Chennai
Tamil Nadu - India

ABSTRACT
Cloud Computing is one of emerging technology nowadays. Checking the data integrity remotely is became a decisive part in cloud computing. Newly, lengthy of works pointing on providing data dynamics and public verifiability for this type of protocols. The previous protocols are also able to provide these features but only with the help of third-party auditor and without security of data. With Reference, F. SEBE [1] propose a remote data integrity checking protocol that supports data dynamics. From this, we adapt protocol to support public verifiability and proposed a public verifiability without the help of third-party auditor along with security of that protocol does not leak any private information to third-party verifiers. Through an Analysis, we show an error free and security of the protocol. And going through theoretical and experimental results, we displayed that the proposed protocol has a good performance.

Keywords :- Data integrity, data dynamics, public verifiability.

I. INTRODUCTION

Storing tons and tons of data on the cloud has become a trend nowadays. A prolific number of clients store their crucial data in remote servers in the cloud, without leaving a snaps in their local computers. The data stored in the cloud is so important that the clients must guarantee it is not lost or corrupted. While it is easy to check data integrity after downloading the data to be checked, downloading bulk amounts of data just for ensuring data integrity is a waste of communication bandwidth. Hence, many works [1], [2], [4], [5], [6], [7], [8] have been done on designing remote data integrity checking protocols, which allow data integrity to be checked without completely downloading the data.

Remote data integrity checking is introduced in Ref [10], [11], which standalone propose RSA-based protocols for resolving this problem. After that Shah. [12] Propose a remote storage auditing method based on precomputed challenge-response input pairs. Recently, many works [1], [3], [4], [5], [6], [7], [8], [9], [13], [14], [15] focus on providing three advanced features for remote data integrity checking protocols: data dynamics [5], [6], [8], [14], public verifiability [3], [8], [9], [14], and privacy against verifiers [9], [14]. The
Protocols [5], [6], [7], [8], [14] which works data dynamics at the block level, including operations like insertion, modification, and block deletion. The protocol of [3] supports data append operation. In addition, [1] can be easily adapted to support data dynamics. Protocols in [9], [13] can be adapted to support data dynamics by using the techniques of [8]. On the other hand, protocols in [3], [8], [9], [13], [14], [15] support public verifiability, by which anyone (not just the client) can perform the integrity checking operation. The protocols in [9], [13], [14], [15] support privacy against third-party verifiers. We compare the proposed protocol with selected previous protocols (see Table 1).

In this paper, we have the following main contributions:

- We propose a remote data integrity verifying protocol for cloud storage, which can be taken as an adaptation of SEBE's protocol [1]. The proposed protocol inherits the support of data dynamics from [1], and supports public verifiability and privacy against third-party auditors, while at the same time it does not required to use a third-party auditor.
- We have given a security analysis of the proposed protocol, which displays that it is secure against the external port server and private against third-party verifiers. .
- We have theoretically analysed and experimentally tested the capacity of the protocol. Both theoretical analysis and experimental results demonstrate that our protocol is efficient.

The rest of the paper is organized as below. In Section 2, technical preliminaries are presented. In Section 3, the proposed remote data integrity ensuring protocol is presented. In Section 4, a formal analysis of the protocol is presented. In Section 5, we explained the support of data dynamics of the proposed protocol. In Section 6, the protocol's complexity is analysed in the mode of communication, computation, and storage costs; furthermore, experimental results are presented for the efficiency of the protocol. And finally, conclusions and possible future work are presented in Section 7.

II. EXISTING SYSTEM

Security issues will be occurring during the transmission. User can able to modify the data during the transmission. Unwanted data will be delivered to the user who will access the cloud server.

We will not predict whether the data should be correct or not. Security is the major issue to be discussed in the Cloud Computing process. Internet threats are increased so data security is to be discussed is to be maintained.

Disadvantages:

- In the existing system does not support the RSA algorithm
- It does not encrypt the data to public key, private key & shared key
- It does not verify the data from the cloud server.

III. PROPOSED SYSTEM

Existing system fails to predict the data consistency. So we introduce a new concept, to monitor the packets by verifier.

Verifier checks the blocks of a data randomly by sending a challenge request and verifying challenge response from that packet after verifier is authorize using its public key. If the challenge and challenge response is matched then the block is normal.

If challenge response is differed from the expected challenge response then the block is affected. Verifier will give alert to the entire user who is all using the cloud server.

This method we can access remote system without any losses or malicious. The Data stored in a Cloud Server is split into blocks.

The Integrity of the blocks are verified randomly by the Third Party Verifier. Verifier will give its public key then the Challenge to a particular Block. The lock will respond with Challenge Response. The Verifier verifies the CR, if it is Genuine then the data is safe condition; if not data Access is blocked

Advantages:

- Security and replicated data from the cloud server
- By using the RSA algorithm to encrypt the data to public key, private key & shared key
- We are verify the verification of data is must hacked

IV. TECHNICAL PRELIMINARIES

We consider a cloud storage system in which there are a client and an untrusted server. The client stores her data in the server without keeping a local copy. Hence, it is of critical importance that the client should be able to verify the integrity of the data stored in the remote untrusted server. If the server modifies any part of the client's data, the client should be able to detect it; furthermore, any third-party verifier should also be able to detect it. In case a third-party verifier verifies the integrity of the client's data, the data should be kept private against the third-party verifier.

Below we present a formal statement of the problem. Problem formulation. Denote by m the file that will be stored in the untrusted server, which is divided into n blocks of equal lengths: m ¼ m1m2 . . MN where n ¼ =le. Here, l is the length of each file block. Denote by FKð _ Þ a pseudo-random function which is defined as

F: {0, 1} ^k {0, 1} ^log2(n) -> {0,1}^d

in which k and d are two security parameters. Furthermore, denote the length of N in bits by |N|.

We need to design a remote data integrity checking protocol that includes the following five functions: setup, tag gen, Challenge, gen Proof, and Check Proof.

Set 1^k -> {Pk,Sk} Given the security parameter k, this function generates the public key pk and the

secret key sk. Pk is public to everyone, while sk is kept secret by the client. TagGen{pk,sk,m} -> Dm. Given pk, sk and m, this function computes a verification tag Dm and makes it publicly known to everyone. This tag will be used for public verification of data Integrity.

TABLE 1
Comparisons between the Proposed Protocol and Previous Protocols

	S-PDP[3]	[1]	[8]	DPDP[6]	[5], [13]	IPDP[14]	The Proposed Protocol
Type of guarantee	probabilistic[1] deterministic	deterministic	probabilistic[1]				deterministic[2]
Public verifiability	Yes	No	Yes	No	Yes	Yes	Yes
With help of TPA[3]	No	No	Yes	No	Yes	Yes	No
Data dynamics	append only	Yes	Yes	Yes	No	Yes	Yes
Privacy preserving	No	–	No	–	Yes	Yes	Yes
Support for sampling	Yes	No	Yes	Yes	Yes	Yes	No
Size of verification tags	O(n)[4]						
Communication	O(1)		O(clogn)	O(logn)	O(c)	O(s)	O(1)
Server block access	O(c)	O(n)	O(c)				O(n)
Server computation	O(c)	O(n)	O(clogn)		O(c)	O(c+s)	O(n)
Verifier computation	O(c)	O(n)	O(clogn)		O(c)	O(c+s)	O(n)
Client storage	O(1)	O(n)	O(1)				O(n)

[1] The probabilistic guarantee of data integrity is achieved by using the probabilistic checking method proposed in [3]. Because the blocks are randomly selected, the detection probability will be high if the server deletes a fraction of all the blocks.
[2] The protocol in [1] and the proposed protocol achieve deterministic guarantee of data integrity, because they check the integrity of all the data blocks. However, both the protocol in [1] and the proposed protocol can be easily transformed into a more efficient one by using the probabilistic checking method [3].
[3] A third party auditor [8], [9], [13], [14] has certain special expertise and technical capabilities, which the clients do not have.
[4] n is the block number, c is the sampling block number, and s is the number of sectors in a block.

V. MODULE DESCRIPTION

User & Data Owner:

The user is the client application who sends a request to the Cloud server in order to obtain the best result for the query. The query given here may be a keyword through which the search is performed and the best results are achieved. Data Owner is the owner of the Data where the data are modified and updated in the Cloud Server.

CloudServer:

Cloud Server is the major main server which contains the main data The Cloud Server will act as the main server to receive the query from the user. All the Data is stored in the main server. once the query is passed to the Cloud Server it will retrieve the Data. The data are spitted cloud.

Verifier:

Verifier is the server where the Data integrity is verified properly and which checks the Data with its integrity. Verifier will send Challenge Request of a Data and the response is verified by the Verifier. Verifier guarantees the Data integrity and ensures security in the process of Cloud Server

Verification of Data block Integrity:

Verifier will select a block randomly in a Block sequence. It sends the Challenge to the Particular Block and according to the Challenge given the corresponding Block will respond with the Challenge Response. This Verifier will check the expected Challenge Response with the Challenge Response given by that particular Block. If the expected Challenge Response and the Challenge Response given by the Block are same then the Data is safe and data integrity isgood**.**

Data Retrieval:

When Ever the data is requested by any user, the verifier will verify the data integrity and if the data is safe, only then the data is retrieved to the user safely by using RSA algorithm.

VI. CONCLUSION

In this project we proposed a remote data integrity checking protocol for cloud storage. This proposed protocol suitable for providing integrity protection of customer's important data. The proposed protocol supports data insertion, modification, and deletion at the block level, and also supports public verifiability and proved to be secure against an untrusted server. The proposed protocol has public verifiability which makes it very flexible and preserves the file privacy against the third party verifier. Currently we are still working on extending the protocols support data level dynamics. The difficulty is that there is no

clear mapping relationship between the data and tags in the current construction, data level dynamic scan be supported by using block level dynamics. Whenever a piece of data is modified the corresponding blocks and tags are updated. However this can bring unnecessary computation and communication costs. We aim to achieve a level dynamics at minimal costs in our future work.

FUTURE ENHANCEMENT

Finally, We believe that cloud data storage security is still full of challenges and of paramount importance and many research problems remain to be identified.

REFERENCE

[1] F. Sabe, J. Domingo-Ferrer, A. Martinez-Balleste, Y. Deswarte, and J.-J.Quisquater, "Efficient Remote Data Possession Checking in Critical Information Infrastructures," IEEE Trans. Knowledge and Data Eng.,vol. 20, no. 8, pp. 1034-1038, Aug. 2008.

[2] R. Buyya, C.S. Yeo, S. Venugopal, J. Broberg, and I. Brandic, "Cloud Computing and Emerging IT Platforms: Vision, Hype, and Reality for Delivering Computing as the Fifth Utility," Future Generation Computer Systems vol. 25, no. 6, pp. 599-616, 2009.

[3] G.Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song, "Provable Data Possession at Un trusted Stores," Proc. 14th ACM Conf. Computer and Comm. Security (CCS '07), pp. 598-609, 2007.

[4] R.Curtmola, O. Khan, R. Burns, and G. Ateniese, "MR-PDP: Multiple-Replica Provable Data Possession," Proc. 28th Int'l Conf. Distributed Computing Systems (ICDCS '08), 2008.

[5] G.Ateniese, R. Di Pietro, L.V. Mancini, and G. Tsudik, "Scalable

[6] C. Erway, A. Ku¨ pc¸u¨ , C. Papamanthou, and R. Tamassia, "Dynamic Provable Data Possession," Proc. 16th ACM Conf. Computer and Comm. Security (CCS '09), pp. 213-222, 2009.

[7] C. Wang, Q. Wang, K. Ren, and W. Lou, "Ensuring Data Storage Security in Cloud Computing," Proc. 17th Int'l Workshop Quality of Service (IWQoS '09),pp. 1-9, July 2009.

[8] Q. Wang, C. Wang, J. Li, K. Ren, and W. Lou, "Enabling Public Verifiability and Data Dynamics for Storage Security in Cloud Computing," Proc. 14thEuropean Conf. Research in Computer Security (ESORICS), Sept. 2009.

[9] Y. Deswarte and J-J.Quisqater, "Remote Integrity Checking," Proc. Sixth Conf. Integrity and internal control in Information systems(IICIS 04),pp. 1-11, 2004.

[10] D.L.G Filho and P.S.L.M. Barreto, "Demonstrating data possession and Uncheatable Data Transfer."CryptologyePrint Archive, Report 2006/150, http://eprint.iacr.org/ , 2006 .

[11] http://java.sun.com

[12] http://www.sourcefordgde.com

[13] http://www.networkcomputing.com

[14] http://www.roseindia.com

[15] http://java2s.com