International Journal of Modern Agriculture, Volume 9, No. 4, 2020
ISSN: 2305-7246

AUTHENTIC AND INVULNERABLE SERVICE IN CLOUD COMPUTING

R.Karthikeyan

Asst professor Dept of Computer science and Engineering BIHER, Chennai

Email:rkarthikeyan78@yahoo.com

Abstract

Cloud storage is a type of online storage which helps to store on multiple virtual servers, rather than being hosted on dedicated servers Since we have remote database it provides us security problems. So to maintain data integrity, proposed design consists of efficient methods that enable on-demand data correctness verification. In this paper, we put more focuses on the support of file-oriented cloud applications. The proposed system ensures effective security measures, utilizing the token pre-computation process and data encryption. Our security analysis shows that both internal and external attacks require exponential computational costs; that is, our scheme is computationally secure against these attacks. So, the distributed protocols for storage correctness assurance will be of very essential to gain robust and secure cloud storage systems. Therefore, the steps toward future cloud computing is flexible and cost-effective for organization used by reliable and secure cloud computing.

Key words: Cloud computing, Data integrity, Dependable distributed storage, Data dynamics, Error localization.

Introduction

Cloud computing is an internet-based development that involves delivering hosted services over the internet and provides on demand network access to multiple resources. The cloud infrastructures are much more powerful and reliable than personal computing devices, but a wide range of threatening is existing for data integrity The internet-based online services to provide huge amount of storage space and customizable computing resources. The services and applications required by those resources are configured and installed on remote locations and are accessible via cloud. The computing platform shift eliminates the responsibility of local machines for data maintenance at the same time. Cloud computing moves the application software and databases to the large data centers, where the administration of the data and services may not be fully trustworthy. The examples for Cloud Computing vendors are Amazon Simple Storage Service (S3) and Amazon Elastic Compute Cloud (EC2) It has the advantage of reducing cost by sharing computing and storage resources, combined with an on-demand provisioning mechanism.

Cloud computing is the delivery of computing as a service rather than a available which shares resources, software and information to computers and other devices as a utility (like the electricity grid) over a network. Cloud computing provides computation, software, data access, and [1]-[2] storage services that do not require the details about the end-user available location and configuration of the system that delivers the services. Parallels to this concept we can consider a electricity grid which uses end-users to consume power without needing to understand the component devices or infrastructure required for the services. So, cloud computing varies much from hosting services and assets at ISP data center. It reveals that the computing systems are found to be logical at particular place or in virtual resources [2] forming a Cloud and user community accessing can be done usig both intranet or Internet.

Cloud computing describes a about the newer supplement, reality of consumption, and details of delivery model for IT work places which uses Internet protocols, and it typically involves provision for the scalable and virtualized resources. This is a type of byproduct and consequence of the ease-of-access to remote computing sites provided by the Internet. It may consider about the web-based tools or applications that users can access and use through a web browser for installion locally on their own computers. Cloud computing providers deliver applications through internet, while the business software and data are stored on devices at particular location. In certain legacy applications they are delivered through a screen-sharing technology, [3]-[4] while the computing resources are consolidated at a remote data center location; in other cases, entire business applications have been coded using web-based technologies such as AJAX. Most of the cloud computing infrastructures provides services passed through shared data-centers and appearing as a single point of access for consumers' computing needs. Very often the commercial offerings are in need to meet service-level agreements (SLAs), but specific terms are less often negotiated by smaller companies.

Cloud computing has been changing how most people use the web and [19] how they store their files. It provides the runs sites such as Face book, Amazon and Twitter and the core that allows us to take advantage of Google Docs and Gmail. The concept of the cloud has been around us for a lengther period of time in many different situations in the business world. It mostly means a grid of computers may serve like a service-oriented architecture to provide software and stored data. Most websites and server-based applications run on particular computers or servers. The cloud uses its stored datasources from the computers as a collective virtual computer, [16] where the applications can run independently from a specific type of computer or server configurations. Basically they are floating around in a “cloud of resources”, making the hardware less important to how the applications work. The cloud takes advantage of that to bring it to the next level. The cloud consists of layers mostly, back-end layers and front-end or user-end layers. The front-end layers are the ones you see and interact with. [23] When you access your email on Gmail. For example, you are using software running on the front-end of a cloud [19]. The same is true when you access your Facebook account. The back-end layer consists of systems such as hardware and the software that fuels the interface you see on the front end.

An understanding of the available techniques provides us about the computing systems and networks reliability, fault-tolerant and secure will be crucial to those who design and [19] deploy the next generation of mission-critical applications and Web Services. Reliable Distributed Systems reviews and describes the logic behind the concepts, available details of principles and versatile applications of modern distributed computing systems and architectures. The IP suite can be viewed as a set of layers, in which very layer having the property that it can particularly uses the functions of the layer below, and only exports functionality to the layer found above the particular layer. A system that provides about the details of protocol behavior consisting of layers is known as a protocol stack and it may be provided either in hardware or software [20], or a mixture of both. Typically, only the lower layers are provided in hardware, whereas the higher layers provided in software.

The rest of the paper is organized as follows. Section 2 which overviews the related work. Then we describe some of the methods related to our project in Section 3. Finally, Section 4 presents concluding remarks and outlines the directions for future work.

Related work

Ateniese et al. [3] has given that the provable data possession (PDP) model is for providing the files on untrusted storages. Their scheme utilized public key for checking the stored data file. Anyhow the pre-computation of tags imposes heavy computation overhead that can be expensive for whole file., Ateniese et al. [6] in his subsequent work described a PDP scheme that uses only symmetric key based cryptography. This type of methodology may allows for block updates, deletions and appends to the stored file for our work. However, their scheme focuses on single server scenario and does not provide assurance for server failures, so that they may leave both the distributed scenario and data error recovery issue unexplored. The details of data dynamics has been provided in the two recent work [7] and [8].

Juels et al. [2] has provided a “proof of retrievability” (POR) model for correcting the remote data integrity. Their scheme combines spot-checking and error correcting code to check the details of the possession and retrievability files on archive service systems. Shacham et al. [9] built on this model and constructed a random linear function which enables unlimited number of challenges and requires less communication of its smaller size of BLS signature. Bowers et al. [10] proposed an improved framework for POR methodology that related the Juels and Shacham's work. However in their subsequent work, Bowers et al. [13] extended POR model to distributed systems. Anyhow all the schemes concentrates on static data. The effectiveness of their schemes rests primarily on their pre-processing steps for outsourcing the data file F. Any change to the contents of F, should always pass through the error-correcting code and the corresponding random shuffling process, thus introducing significant computation and communication complexity. Recently, Dodis et al. [12] gave theoretical studies on generalized framework for different variants of existing POR work.

The cryptography work carried out by Bell are et al. [17] also provides a set of cryptographic building blocks such as hash, MAC, and signature functions which is used for storage integrity verification while supporting dynamic operations on data. Anyhow this type of work comes under the traditional data integrity protection mechanism, where local copy of data has to be kept for checking. So far it is not well understood how the work can be adapted to cloud storage scenario so that the users may not store at local sites but still need to ensure the storage correctness efficiently in the cloud. Lilli bridge et al. [15] presented a P2P backup scheme in which blocks of a data file are kept between the m + k peers using an (m, k)- erasure code. Peers can request random blocks from their backup peers and shows the keys for cryptographic hashes attached on each block. Their scheme can detect data loss but cannot check all data which is unchanged.

In other related work, Curtmola et al. [11] aimed to check the details of data possession of multiple replicas found throughout the stored system. They extended the PDP scheme to cover multiple replicas without covering the particular replica individually but giving warranty to the multiple copies of data are actually maintained. Filho et al. [18] proposed check the details of data integrity by using RSA-based hash to illustrate the uncheatable data possession in peer-to-peer file sharing networks. However, their proposal needs the entire data file, which may show the impractical for the server whenever the file is large. Schwarz et al. [14] proposed to ensure static file integrity across multiple distributed servers, using erasure-coding and block level file integrity verifications. However, our schemes further support data dynamics and explicitly study towards the problem associated with misbehaviour of server identification. Very recently, Wang et al. [16] gave a study on the present solutions on remote data integrity verification, and discussed their pros and cons under different design scenarios of secure cloud storage services.

System design

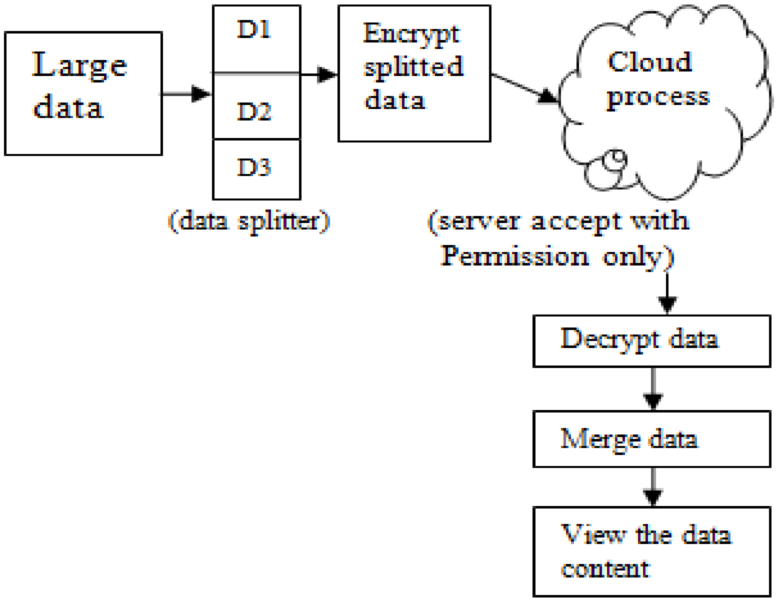


Fig a: System Architecture

International Journal of Modern Agriculture, Volume 9, No. 4, 2020
ISSN: 2305-7246

Data Splitter

In order to deploy the data in the cloud, data rifting or splitting has been done. Data splitting is taken in the form of blocks of data by means of Token Pre-computation Algorithm. Every data has been tokenized to ensure the faster computation and execution in the cloud environment.

Here, the token pre-computation process is used for the data or applications are processed in the cloud are taken in the form of tokens or small blocks of data.

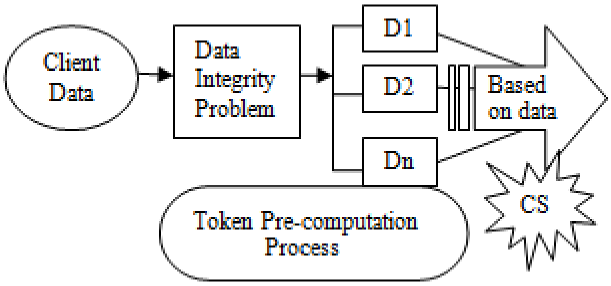


Fig1: Data Splitter

Figure 1 gives the details of Client data's are stored in the Cloud servers. For Data integrity problem, we should split the data into many. In order to store the data's into multiple servers, we should split the given data's. Based on the splitted data size, data's are stored into respective servers.

Data integrity in cloud based on selecting random bits in data blocks the client before storing its data file f at the client should process it and create suitable Meta data which is used in the later stage of verification the data integrity at the cloud storage. When checking for data integrity the client queries the cloud storage for suitable replies based on which it concludes the integrity of its data stored in the client.

The replaced algorithm of token pre-computation or Reed-Solomon algorithm is Berlekamp-Massey algorithm. It is used to find the Error Recovery. While the data is sent to the client from the server, sometimes the data get loss. That is, the data loss means error recovery. Therefore, to rectify the data errors is otherwise called Byzantine failure. According to the Reed-Solomon algorithm, the modification process is done by user's client.

Encrypt Splitted Data

Data's are securely stored in the cloud servers as it has less security options. In cloud servers client data's are stored as secured data's so the crypto processes have applied. For crypto process, we use BLOWFISH algorithm for the encryption and decryption process. Using BLOWFISH algorithm, data's are converted as crypto data's.

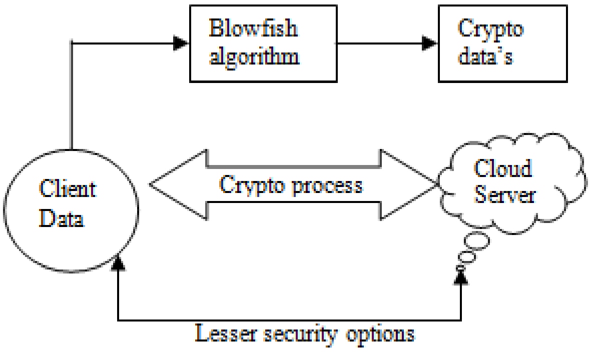


Fig2: Encrypt Splitted Data

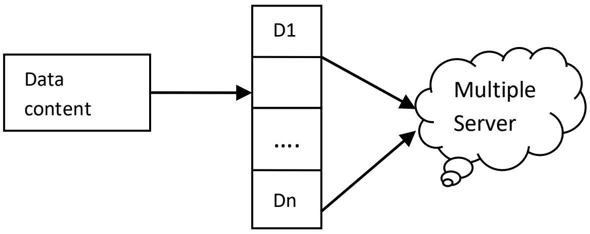
Since client data's are stored in the cloud server, they have lesser security options. To overcome this security, we have implemented the crypto process.

Blowfish is a symmetric block cipher that can be effectively used for encryption and safeguarding of data. It takes a variable-length key, from 32 bits to 448 bits, making it ideal for securing data. Since then it has been analyzed considerably, and it is slowly gaining acceptance as a strong encryption algorithm. Blowfish is unpatented and license-free, and is available free for all uses.

While no effective cryptanalysis of Blowfish has been found to date, more attention is now given to block ciphers with a larger block size, such as AES or Two fish. Blowfish is a variable-length key block cipher. It does not meet all the requirements for a new cryptographic standard. It is only suitable for applications where the key does not change often, like a communications link or an automatic file encryptor. It is significantly faster than DES when implemented on 32-bit microprocessors with large data caches, such as the Pentium and the PowerPC.

Data Access and Cloud Storage

Client data's are stored in the cloud servers. Since cloud Infrastructures are shared with clients, client data's are stored in the cloud servers**.**



(Organize the splitting data's)

Fig3: Client Data Access

This figure shows that we are organizing the data's that is being stored as splitted data's into the multiple servers. Here, we are keeping the single file content into multiple servers.

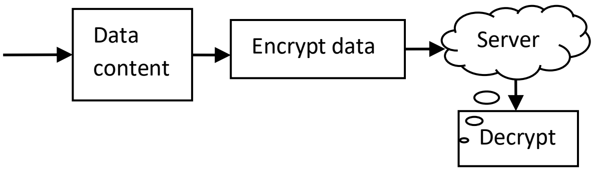
The client data have been applied security. These data's are stored in the cloud servers. In order to does that, the cloud server have to configure. Cloud server has been configured with VMware and its IP address should be configured. Through the Cloud Server IP address client files are stored in cloud servers.

In order to decrypt and process the cloud data, given data should be stored. This process consists of data access that is being stored from multiple servers.

Cloud storage is made up of many distributed resources, but still acts as one highly fault tolerant through redundancy and distribution of data highly durable through the creation of versioned copies typically eventually consistent with regard to data replicas.

Cloud Decryption

After receiving the respective splitted data's in the cloud server, it has to be decrypted. By BLOWFISH algorithm, the private key has been used to decrypt the respective splitted data's. Private key which is 56 bits in length which has faster in process of ciphering and deciphering the text.



(Blowfish algorithm)

Fig4: Cloud Decryption

In order to get view the original content of the data, the encrypted data should be decrypted. Each and every encryption data's should be decrypted. Decryption process is done by BLOWFISH Algorithm.

Using DES or Blowfish algorithm, Crypto process or data security process is applied. DES or Blowfish is a Symmetric Key Encryption process where a single private key is used for both encryption and decryption. Blowfish Algorithm is a Feistel Network, iterating a simple encryption function 16times. The block size is 64 bits, and the key can be any length up to 448 bits.

Raw Data Merge

Decrypted data's are finally merged. Splitted data's are merged to check whether the given source data is received properly.

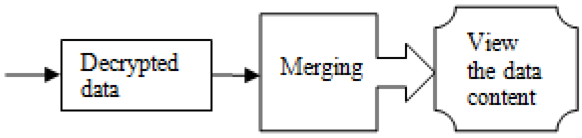


Fig5: Raw Data Merge

Merging process ensures user's data verification process.

In this design, each data is accessed by randomized key. But the dynamic data's are accessed by any key generation. For merging the data's, it uses the random key. Therefore, by considering the data vector as 3x3 and 2x2 regular matrixes, the given data content has been taken.

Conclusions and futurework

In order to avoid the problem of data integrity and to maintain the system with the data security, we propose an effective and flexible distributed storage verification scheme data support to ensure the correctness and the availability of user's data in the cloud. Through detailed security and extensive experiment results, we show that our scheme is highly efficient and resilient to Byzantine failure, malicious data modification attack, and even server colluding attacks. Therefore, the steps toward future cloud computing is flexible and cost-effective for organization used by reliable and secure cloud computing.

As a future work, we can use concurrent or parallel processing for many users working at runtime. So that, it handles the technique as Multiple Concurrent Tasks. Such a technique supports the aggregation of multiple signatures by distinct signers on distinct messages into a single signature and thus allows efficient verification for the authenticity of all messages.

Acknowledgement

This work was supported in part by the US National Science Foundation under grant CNS-0831963, CNS-0626601, CNS-0716306, and CNS-0831628.

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.

A. Juels and J. Burton S. Kaliski, "Pors: Proofs of retrievability for large files," in Proc. of CCS'07, Alexandria, VA, October 2007, pp. 584–597.

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, Alexandria, VA, October 2007, pp. 598-609.

3. 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.

4. M. A. Shah, R. Swaminathan, and M. Baker, "Privacy-preserving audit and extraction of digital contents," Cryptology ePrint Archive, Report 2008/186, 2008, http://eprint.iacr.org/.

5. G. Ateniese, R. D. Pietro, L. V. Mancini, and G. Tsudik, "Scalable and efficient provable data possession," in *Proc. of SecureComm'08*, 2008, pp. 1 10.

6. 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, volume 5789 of LNCS*. Springer-Verlag, Sep. 2009, pp. 355–370.

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

8. H. Shacham and B. Waters, "Compact proofs of retrievability," in *Proc. of Asiacrypt'08, volume 5350 of LNCS*, 2008, pp. 90–107.

9. K. D. Bowers, A. Juels, and A. Oprea, "Proofs of retrievability: Theory and implementation," in *Proc. of ACM workshop on Cloud Computing security (CCSW'09)*, 2009, pp. 43–54.

10. R. Curtmola, O. Khan, R. Burns, and G. Ateniese, "Mr-pdp: Multiple-replica provable data possession," in *Proc. of ICDCS'08*. IEEE Computer Society, 2008, pp. 411–420.

11. Y. Dodis, S. Vadhan, and D. Wichs, "Proofs of retrievability via hardness amplification," in *Proc. of the 6th Theory of Cryptography Conference (TCC'09)*, San Francisco, CA, USA, March 2009.

12. K. D. Bowers, A. Juels, and A. Oprea, "Hail: A high availability and integrity layer for cloud storage," in *Proc. of CCS'09*, 2009, pp. 187–198.

13. T. Schwarz and E. L. Miller, "Store, forget, and check: Using algebraic signatures to check remotely administered storage," in *Proc. of ICDCS'06*, 2006, pp. 12–12.

14. M. Lilli bridge, S. Elnikety, A. Birr ell, M. Burrows, and M.Isard, "A cooperative internet backup scheme," in *Proc. of the 2003 USENIX Annual Technical Conference (General Track)*, 2003, pp. 29–41.

15. C. Wang, K. Ren, W. Lou, and J. Li, "Towards publicly auditable secure cloud data storage services," *IEEE Network Magazine*, vol. 24, no. 4, pp. 19–24, 2010.

16. M. Bell are, O. Goldreich, and S. Goldwasser, "Incremental cryptography: The case of hashing and signing," in *Proc. Of CRYPTO'94, volume 839 of LNCS*. Springer-Verlag, 1994, pp. 216–233.

17. D. L. G. Filho and P. S. L. M. Barreto, "Demonstrating data possession and uncheatable data transfer," Cryptology ePrint Archive, Report 2006/150, 2006, http://eprint.iacr.org/.

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

19. Sun Microsystems, Inc., "Building customer trust in cloud computing with transparent security," Online at https://www.sun. Com/offers/details/sun transparency.xml, November 2009.

20. M. Arrington, "Gmail disaster: Reports of mass email deletions," Online at http://www.techcrunch.com/2006/12/ 28/gmail-disasterreports-of-mass-email-deletions/, December 2006.