International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2013): 6.14 | Impact Factor (2013): 4.438

Enabling Public Auditability, Dynamic Storage Security and Integrity Verification in Cloud Storage

Pranita Bakka[1], Dr. Jayashree Agarkhed[2]

[1]P.G.Student, Department of Computer Science & Engineering, PDA College of Engineering Kalburgi, Karnataka, India

[2]Professor, Department of Computer Science & Engineering, PDA College of Engineering Kalburgi, Karnataka, India

Abstract: ***Cloud computing is an information delivery model, where resources are retrieved from the web based-tools without direct connection to the server and it moves services - such as storage and applications to the server through the internet. One of the important issue in cloud computing is security of remotely stored data from client point of view. The data which is stored on server lacks data integrity, where often checking of data which is stored on server is carried out. This method of storing and managing the data possess many security challenge. This paper mainly focus on data storage security and integrity checking. The data integrity checking has been carried out using a trusted third party (TTP), is the trusted person has the ability to expose risk of cloud storage services on behalf of the clients upon request. Along with checking integrity of stored data, the proposed work also supports storage security dynamically. Motivation of the work, inferred from literature survey. RSA based storage security system (RSASS) uses auditing of text file publically by existing RSA algorithm and integrity checking can be done by using HASH function. Using this scheme the security for data which is stored on cloud by TTP, dynamic cloud storage security by changing cloud and stored data integrity checking is improved while compared with existing system.***

Keywords: Public auditing, Dynamic data storage, integrity proofs and cloud computing.

1. Introduction

There are many trends which are opened up in the era of Cloud Computing, which is based on Internet development and uses the technology of computer. The most cheapest and powerful processors, together with computing architecture that is software as a service (SaaS), are moving the data in to server in a huge scale. Meanwhile, bandwidth of network increasing and more reliable connections of network make the clients to subscribe high quality services from data and software which are remotely stored [1].

There are many benefits when we move data into the cloud like users need not care about the direct management of complexities associated with hardware. Among the several of vendors of Cloud Computing, Amazon Simple Storage Service (S3) and Amazon Elastic Compute Cloud (EC2) both are most popular example. These online services based on internet do provide large amounts space for storage and customizable computing resources, which results in, users are have ability to use cloud service providers for the availability and integrity of their data. From the point of data security, which is an important aspect of Quality of Service (QoS), Cloud Computing possess many security challenges. Firstly, primitives of traditional cryptographic for the purpose of protection of data security because of the user will loss control of data under Cloud Computing cannot be directly adopted. Secondly, Cloud Computing is not data ware house for third party. The data which is stored on cloud involved in operation like insertion, modification and deletion, etc. [2].
The owner creates the text file & saves that text file in the cloud service provider, without keeping a same file in the local computers. It is necessary that the file which is remotely stored on cloud server should not be corrupted. The integrity verification of text file is an opened challenge in many proposed work. This involves in auditing of text file publically and privately. Auditing privately has ability to provide efficiency with higher scheme, while auditing of text file publically supports any one, without keeping any information privately it allows clients to focus only on storage security. In auditing text file publically, the client sends the public key to the trusted third party (TTP). The TTP will verifies clients text file and informs client about text file security which is saved on server. Integrity checking of file which is remotely stored is a big problem, solution for this issue is by using proposed RSASS (RSA storage security) method, this proposed method based on auditing of file which is stored on the remote server using trusted third party (TTP). In the proposed RSASS system focuses mainly on three important protocols for remote data integrity checking: public auditability, data dynamic storage security and data integrity proofs.

1.1 Related Work

Many researchers have used cloud computing for the numerical security and performance concerns. Exhaustive literature review has been carried out and presented as under
In [1] the author focus on providing the security for data which is stored on cloud. Which is important concept of QOS. The author has proposed a suitable scheme with two features, insurance for storage correctness that is integrity checking and localization of errors which are occurred in data. This scheme support for data dynamic operations, this supports any one of the public auditability or data dynamic operations. In [2] the authors have contributed to reveal on cloud data storage security, for the confirmation that data which user stores on the remote cloud is not corrupted or modified, they propose the best distributed scheme with two features, which uses protocol “challenge and response” this provide the method for checking both correctness of data and errors can be easily identified. This only perform partial support for data dynamic operations. In [3] the authors

Licensed Under Creative Commons Attribution CC BY

proposed a method for carrying out, services for storage-outsourcing and networks for resource-sharing, the provable data possession (PDP) model is based on symmetric key cryptography. However, this PDP scheme can applicable for static files. In [4] author studied the problem of outsource the data in the encrypted form. This is based on public key cryptography for encryption of outsourced data. This allows outsourcing of dynamic data with partial support of dynamic data operations. But this does not achieve fully dynamic data operation and answer only partial queries. In [5] the authors have proposed a PDP model. This scheme supports PDP model checking of remote data and large data set can be supported in a distributed storage manner. This work uses the BLS algorithm. BLS based storage security system supports limited block size and gives the result only limited query. In [6] the authors have defined and explore proofs of retrievability (PORs). A POR is a proof of knowledge (POK) cryptography, it provides guarantee for QoS and it shows within time limit file can be retrieved. Updating dynamically prevents sequential nodes & queries of clients can be fixed priory. In [7] the authors have conducted m/n erasure-correcting stored data and for verification uses the Hash function with algebraic functions and properties. However computational complexities of file block will take place in a linear manner at server and client side. In [8] the author proposed dynamic privacy-preserving public auditing protocol. It provides the way to check integrity of data file stored on cloud. This scheme supports only for dynamic data not for static data. In [9] the author have proposed a homomorphic encryption scheme based on the Elliptic curve cryptography. It implements a provable data possession scheme to support dynamic operation on data. The application of proof of retrievability scheme provisioned the client to challenge integrity of the data stored. But storage security is not that much strong they provide security using small key.

2. RSA based Storage Security System

In this section the design and implementation of the proposed system is discussed

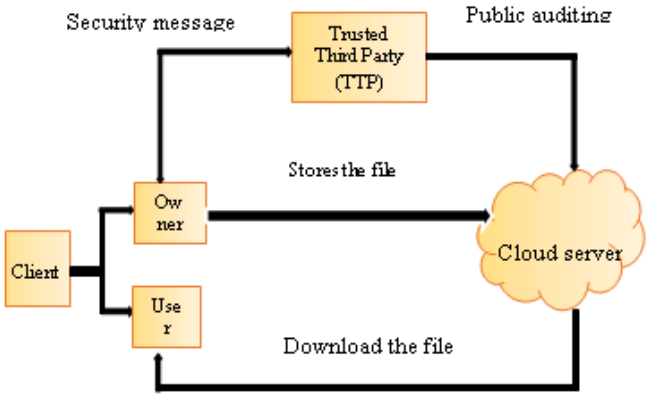


Figure 1: Block Diagram of RSA based Storage security

Block diagram of RSASS is shown in Fig 1. It consist of three entities as client, trusted third party (TTP), cloud service provider (CSP). Client system consist of owner and user, where owner is the person who has text file and loads text file on the cloud storage for security. The client may be a single user or an organization. The user is public who tries to access owner's text file. All of these three entities need to register and then owner will login and will give the file name which he wants to upload and choose the file and upload it on a cloud. TTP is the entity who will login and verify the file uploaded by the owner by decrypting it and upload the file for the conformation like he verified the owner's file. CSP is the entity who will login and tries to verify file uploaded by owner but he can't access it directly so he will get key by hacking it and then tries to make modification but the original data can't modify so he act like actually he verified owner's file and for the confirmation he will upload the file. Now owner will login and check the file alert and get to know by TTP like CSP tries to modify his data. Now for dynamic data storage security TTP and CSP will register and login twice. They will verifies text file of owner by decrypting it and as a confirmation both will upload a file. And then user will login and tries to access the owner's text file but he can't access it. The owner will gives security by RSA based key generation so owner will login and send a key by entering the valid email id and password after that user will check the mail and get the key and enter that key to access the owner's file and verify it then download it. If the downloaded file is same as that of the uploaded file then integrity proofs is achieved.

3. Proposed Methodology

In this proposed methodology, security for the remotely stored data is carried out by the RSASS using RSA based key generation algorithm. The procedure of protocol divided in to following steps: set up phase, Public auditability for correctness assurance of data storage, Dynamic data storage security, Multiple Auditing of files for number of clients, Data integrity verification.

3.1 Setup

In this phase Key Generator KeyGEN ( ) and Hash function method are for securing data file by using cryptography key generation as public and private. The RSA is one of the efficient cryptographic algorithms. The amount of confidentiality, data integrity and privacy is provided by using this. In the key generator method user uses the RSA algorithm and generates the encrypted messages. The receiver, on receiving the messages decrypts them. In secret key cryptographic algorithm has a key, which is used for both encrypting and decrypting the text file. Using this key code that obtain plain text when applied to cipher text. This key can be commonly used by sender and receiver. Its N secret key cryptography has only a key. If the key is disclosed the information security is compromised, hence it does not provide the security to the sender when the receiver receives the message it blames the sender that he only sent. A key which is public, it can be known by everyone, used to encrypting text file and a key, which is private known to only the recipient of the text file, it can be used for decryption. The Hash Function used for information encryption. Hash functions used in many operating systems to encrypt passwords for providing the proof of integrity to a file.

Licensed Under Creative Commons Attribution CC BY

3.2 Public auditability for correctness assurance of data storage

The owner of the file uploads the file on server, TTP has the ability to check correctness of stored file. TTP will register and login then verify the file and upload a file as a confirmation that he verified the owner's text file. If anybody attempted to modify the owner file TTP will audit and sent file alert to owner of the file.

3.3 Dynamic data storage security

The clients are allowed to perform integrity of remotely stored text files while maintaining the correctness assurance of data storage. The proposed work aim to provide both auditing of text file publically and dynamic data storage security by changing the cloud.

3.4 Multiple Auditing of files for number of clients

As cloud servers can be carry out multiple auditing for different clients, given N number of different files with different file sizes from N number of clients, it is more efficient to averaging auditing time into a one short and do auditing on behalf of client. To obtain this result, we extend proposed work to support for multiple auditing. By doing this communication cost is greatly reduce and security is archived with high message authentication.

3.5 Integrity checking of text file

The text file can be checked either by client or TTP. This is done by giving text file to server. Owner will upload a file, TTP and CSP will verify a file as a confirmation they upload a file. Then the client in order to decrypt the files content uses public key and private key and download the file. If the file downloaded is same as that of the uploaded file then we achieved the integrity.

Algorithm

The proposed work can be implemented in java by using Net Bean 6.9.1 software. It uses Cryptographic Algorithm RSA for key generation, encryption, decryption and HASH function for providing integrity proofs.

1. Cryptographic Algorithm for key generation
The RSA Algorithm is used to create a private- public key pair. Both these involved in Set up phase. In this method user uses RSA algorithm and generates the encrypted messages. The receiver, on receiving the messages decrypts them using the RSA algorithm.

A.Algorithm 1: RSA
The steps for implementation of RSA algorithm are given below
Step 1: Begin
Step 2. Get two integers, p and q from the user.
Step 3. Check if p and q are prime. If prime, continue the process, else exit the code.
Step 4. Calculate (p-1)*(q-1) and name it as Φ(n).
Step 5. Calculate n=p*q.
Step 6. Select e; such that, e is relative prime to Φ(n) and e< Φ(n), gcd (e, Φ(n)) = 1
Step 7. Select d; such that, e=1 mod Φ(n) or d.e mod Φ(n) =1
Step 8.We get the public and private key such as
Public Key: [e, n]
Private Key: [d, n]
Step 9: End.

B. Encryption:
The Encryption function is
$C = M^e \bmod n$.

C. Decryption
The Decryption function is
$D=M^d \bmod n$.

2. HASH Function for Data integrity using MAC (message authentication code)

The Hash Function for encrypting the information uses one of the mathematical transformation. This algorithm for encryption and decryption of text file does not use keys. Hash functions used in many operating systems to encrypt passwords for providing the proof of integrity to a file.

Algorithm: Hash function

The steps for implementation of HASH function algorithm are given below
Step 1: Begin
Step 2: Cryptographic hash function h (k, m) with two inputs:
Step 3: Message integrity with MAC
Step 4: HMAC
Step 5: HMAC Provides a secure constructions
Step 6: Hash Chain
Step 7: Validation Chain
Step 8: Hash Tree
Step 9: Hash Tree Authentication
Step10: End.

4. Performance Analysis

In RSASS, The RSASS generates large key under different file sizes. Moreover, generating the keys provide much security there is not necessary for encryption & decryption. The storage space for key generation, Hash function can be used for integrity checking of text file. File is used instead of file Block .Fig 2 gives the server computation time with file size, Fig 3 gives average auditing time over number of client and Fig 4 gives data integrity checking in a constant order of RSASS compared with S-PDP method.

Licensed Under Creative Commons Attribution CC BY

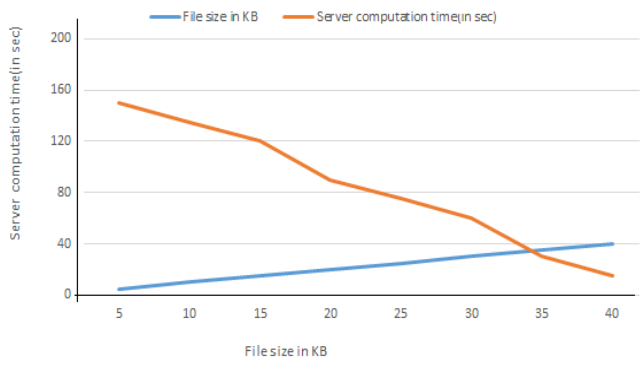


Figure 2: Server computation time over file size

Fig 2 shows, variation of server computation time head with increase in file size. It can be observed that as the size of file increases, the server time decreases. As a proof we obtained that the proposed wok which is based on RSA scheme server computation time reduces even as the file size increases. RSA based approach is yielding more performance.

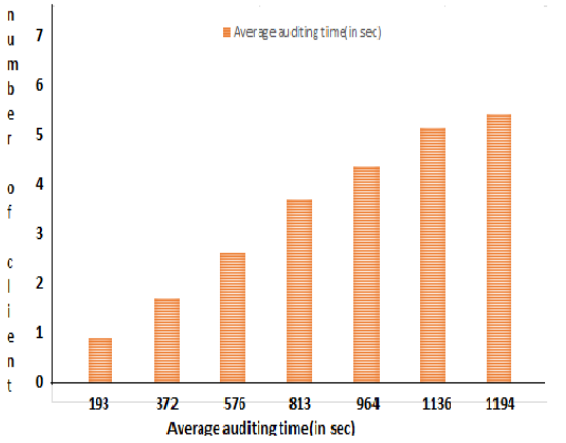


Figure 3: Average auditing time for number of client.

Fig 3 shows, number of clients in the proposed work and average auditing time per client. As the one auditing at one time is giving better result when compared with multiple auditing approach.

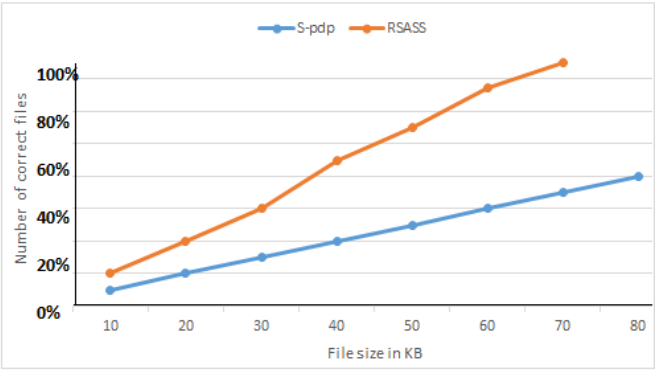


Figure 4: Data integrity of data files comparison

Fig 4 shows, the file data integrity of S-PDP method is less when compared with the proposed method which is based on RSA. RSA based proposed work is obtaining better performance in terms of integrity checking.

5. Conclusion

The proposed system provides an RSA based storage security by generating the key using RSA algorithm which supports text files with different file size, offers strong security in storing the file data in cloud and also provides the storage security dynamically by changing the cloud. This system provides auditing of text file publically which is stored in the server using often integrity verification of stored data file, dynamic storage security and dynamic operation like insertion and deletion of file and provides the integrity proofs. This system can be used in large public databases such as medical archives, astronomy, digital libraries, etc.

By using the Meta cloud it is possible to mitigate or move user from one cloud to next cloud providing dynamic storage security. The Meta cloud already exist it can be used in many technologies, yet lack integration. In this we can achieve the integration proofs.

References

[1] Qian Wang, Cong Wang, Kui Ren, Wenjing Lou, "Enabling Public Auditability and Data Dynamics for Storage Security in Cloud Computing", IEEE Transaction on parrellel and distributed systems, May 2011.
[2] C.Wang, Q.Wang, K.Ren, W.Lou, "Ensuring Data Storage Security in Cloud Computing", Proc. 17th Int'l Workshop Quality of Service (IWQos'09), 2009.
[3] C.Erway, A.Kupcu, C.Papamanthou, R.Tamassia, "Dynamic Provable Data Possession", Proc. 16th ACM Conf. Computer and Comm. Security (CCS' 09), 2009.
[4] G.Ateniese, R.D.Pietro, L.V.Mancini, G.Tsudik, "Scalable and Efficient Provable Data Possession", in Proc. of SecureComm'08. New York, NY, USA: ACM, 2008, pp. 1–10.
[5] H. Shacham and B. Waters, "Compact Proofs of Retrievability," Proc. 14th Int'l Conf. Theory and Application of Cryptology and Information Security: Advances in Cryptology (ASIACRYPT '08), pp. 90-107, 2008.
[6] A. Juels and B.S. Kaliski Jr., "Pors: Proofs of Retrievability for Large Files," Proc. 14th ACM Conf. Computer and Comm. Security (CCS '07), pp. 584-597, 2007.
[7] T. S. J. Schwarz and E. L. Miller. "Store, forget, and check: Using algebraic signatures to check remotely administered storage", in Proc. of ICDCS'06, Lisboa, Portugal, 2006, pp. 12–12.
[8] Mrs. R. Navajothi, Mr. S. Jean Adrien Fenelon" an efficient, dynamic, privacy preserving public auditing method on untrusted cloud storage" IEEE 2014.
[9] Tamal Kanti Chakraborty, Anil Dhami, Prakhar Bansal and Tripti Singh" Enhanced Public Auditability & Secure Data Storage in Cloud Computing" IEEE 2012.
[10] Pritee Parwekar, Mayuri Saxena, Prakash Kumar, Sakshi Saxena Computer Science and Engineering "Public Auditing: Cloud Data Storage" IEEE 2014.
[11] K.Gayathri, P.Umamaheswari, P.Senthilkumar "Enabling Efficiency in Data Dynamics for Storage

Licensed Under Creative Commons Attribution CC BY

Security in Cloud Computing" IJARCCE Vol. 2, Issue 12, December 2013.

[12] M.A. Shah, R. Swaminathan, and M. Baker, "Privacy-Preserving Audit and Extraction of Digital Contents," Cryptology ePrint Archive, Report 2008/186, 2008.

[13] M.A. Shah, M. Baker, J.C. Mogul, and R. Swaminathan, "Auditing to Keep Online Storage Services Honest," Proc. 11th USENIX Workshop Hot Topics in Operating Systems (HotOS '07), PP. 1-6, 2007.

[14] M.Venkatesh, M.R.Sumalatha, Mr.C.SelvaKumar. "Improving Public Auditability, Data Possession in Data Storage Security for Cloud Computing.",ICRTIT-2012,IEEE,2012.

Licensed Under Creative Commons Attribution CC BY