Efficient data security using Dynamic Token check for Cloud Storage Systems

P.Srinivas * , G. Rajesh Kumar #

**Department of Computer Science (CSE) , Swarnandhra College of Engineering and Technology(SCET)*

Abstract :

Cloud computing is the delivery of computing as a service rather than a product. It provides shared resources, software, and information to computers and other devices over a network. The increasing network bandwidth and reliable yet flexible network connections make it even possible that users can now subscribe high quality services from data and software that reside solely on remote data centers. We can store and retrieve the data as we like using cloud computing. To maintain the data securely in distributed environment i.e., on clouds we propose an effective and flexible distributed scheme with Token Generation algorithm for data files checking as a secure and dependable cloud storage service. A new scheme was introduced to encrypt with the user specified key parameters to make the resource more robust. We derive a new algorithm which is very light weight and easy to compute. Here we stores the encrypted blocks into cloud and perform token checking on this encrypted blocks which gives more security to data. We verify the data effectively in case of any block modifications of files before storing to Clouds by token acknowledgment. The proposed scheme is highly efficient and resilient against attacks like Byzantine server failures, malicious data modification attack. Two way verification of file blocks which results more robust and ensure that data will not be modified before reaching to clouds.

Keywords : Trusted Party Auditor (TPA) ,Cloud service provider (CSP) , Token key generation algorithm, Homomorphic mechanism, FileToken

1.INTRODUCTION

Cloud computing is the term used to share the resources globally with less cost .we can also called as 'IT ON DEMAND'. It provides three types of services I,e., Infrastructure as a service(IaaS) , Platform as a service(PaaS) and Software as a service(SaaS). End users access the cloud based applications through the web browsers with internet connection. Moving data to clouds makes more convenient and reduce to manage hardware complexities. Data stored at clouds are maintained by Cloud service providers (CSP) with various incentives for different levels of services. However it eliminates the responsibility of local machines to maintain data, there is a chance to lost data or it effects from external or internal attacks. To maintain the data integrity and data availability many people proposed several algorithms and methods that enable on demand data correctness and verification. So Cloud servers are not only used to store data like a ware house , it also provides frequent updates on data by the users with different operations like insert, delete , update and append. Lastly the deployment of cloud computing is powered by the data centers running in cooperated and distributed manner Recently, the importance of ensuring the remote

data integrity has been highlighted by the following research works under different systems and security models . Researchers also proposed distributed protocols to ensure storage correctness across multiple servers.In this paper we mainly focus on dynamic generation of tokens for verification of blocks and token precomputation before distributing the files in to cloud, the rest of the operations are taken from the same paper(towards secure and dependable storage services in cloud computing) . we use homomorphic token for challenge responses with few parameters . It will easily trace the misbehaving servers which are having corrupted data blocks. We need not concern about distribution of blocks we can distribute the blocks redundantly to many servers or disperse the blocks to different servers with server persistent schemes. We also provide the extension of proposed scheme to support third – party auditing, where user can safely delegate the integrity checking task to trusted third party providers

2. PROBLEM STATEMENT

2.1 System Model

The cloud storage system architecture consists of following network entities

User: An entity, who performs data storage and retrieval operations without knowing the internal issues.

Cloud Server (CS): An entity, which provides data storage space and resources required for computations, cloud servers are managed by cloud service providers.

Third Party Auditor (TPA): An optional Entity, but here we use TPA as Trusted party and to perform some computations instead of users.

Working:

In cloud data storage system, user can upload or stores the data into cloud or use services from the cloud(Here we focused on file storage and retrieval operations). user stores data into set of cloud servers which are running in a distributed and cooperated manner. Data redundant techniques can be employed using erasure correcting code to protect from faults or server crashes. Users can perform manipulations on stored data like insert update and append through blocks. Block level updations and deletions are allowed with token checking. If user has not having enough resources to compute tokens or required hardware support then he can easily delegate the work to a third party auditor called as TPA. He is responsible to generate homomorphic token and stores the token persistently and securely for further verification. In our scheme we assume that TPA is secure and he is responsible to protect from threats, users will pay some incentives to TPA for maintenance.

2.2. Adversary Model

Adversary model was introduced to explore some of threats associated in this model. As we know that the data is not present at users place because data is stored at cloud servers. It may lead to some security threats mainly two, internal attacks and external attacks. Internal attacks comes from the cloud servers itself, these servers may be malicious and lead to byzantine failures and hide some data loss issues. Secondly external attacks are from outsiders who are compromised the data from cloud service providers without its permission. Outsider attacks may lead to modification of data or deleting the users and so on which are completely masked from cloud service providers. All though TPA can also possibly hack the data for itself interested and it is also a

case for inside attacks, but we ensure that TPA's are trusted party servers.

Therefore, we consider the adversary in our model to capture all types of attacks both internal and external threats. Once the server is compromised, the data is polluted with fraudulent data and users cannot get the original data from the clouds

2.3 Design Goals

Our main goal is to ensure the data integrity and security .In this paper, we aim to design 1)precomputation token key generation algorithm which is simple , elegant and secure method and less overhead due to few parameters that has to be chosen.

2) Challenge verification scheme was designed in easy and efficient way to prevent data from byzantine server failures and data dependability detection or detect data errors on blocks

3) Cloud servers ensure that the file was saved successfully without block modifications. This can be achieved by two way token checking.

Architecture of cloud storage system:

Fig: Cloud Storage System Architecture

3.ENSURING DATA STORAGE OVER CLOUD

In cloud data storage system, users store their data remotely i,e., on clouds, so that the correctness and availability of data files must be guaranteed to be identical. Our aim is to detect the servers which behaves differently and may leads to internal and external threats. In this paper, we explore the technique used to detect the modified blocks easily with very less overhead using homomorphic token precomputation technique ,later we can use erasure coded technique to acquire the desired blocks from different servers

3.1Challenge Token Pre-Computation

To achieve data storage correctness and data integrity, we use an algorithm which takes a few parameters and compute the token.

Token generation algorithm works as follows:

(Here we assume TPA will participate in key generation)

Let **F** be the filename and **fL** be the length of the file and **V** be the secret matrix which contains special characters in randomized order.

Compute the key with the following parameters:

Algorithm **Pre-TokenGeneration**

Procedure

Choose parameters F , fL and secret vector V

Choose number of blocks to be taken (normally fixed block size)

X= F + fL + V

Compute key

for i=1 to n

fileToken =fileToken +($\sum_{i=1}^{n}$ split(X_i))

Compute short signatures for each block of the file by considering Token and file block data using bit permutations (Token +block data) and store these values in client for dynamic checking

---

End procedure

Before file is distributed to the cloud, TPA will generate token key with required parameters passed by user. once the token key has been generated, TPA will send the file by dividing the file into equal sized blocks and generate a small token signature for each block along with initial key **filetoken** .This fileToken was generated based on mathematical calculations with hash based technique, It is fully randomized we are not explore the operations present in it and just given the function split(X). Before sending the block it stores the computed signatures obtained from bit permutations on both fileToken and block data .The resultant token was stored in its database or at clients place. Each block is send along with short signature and each block is treated as encrypted block.

Cloud will perform the same operation and checks whether the given block is same or not when computed and checks with the signature. If it matches the same, cloud server store each block and acknowledges the newly generated signature to TPA.TPA verifies the signature with the existing signature, if it matches TPA will send next block otherwise it assumes that block was not saved successfully or it may effect to attacks and resend the same block

3.2 File Retrieval and Misleading block checking using Token Computation

In this paper we focus on this issue related to retrieval of a file in efficient manner. The tokens of each block which we were generated using precomputation algorithm has been stored in the database. Now we are using homomorphic technique to retrieve entire file or required blocks dynamically. Once user has been sent the requested file to TPA. TPA monitors whether he is authenticated user or not for accessing the file. TPA maintains the file details and tokens (if TPA is not present user will have the details) but not an entire file, TPA requests the file by passing the pre-computed token stored in the database for each block. If this token is same as it is present in cloud server, cloud server will send the requested blocks.

We can easily check whether the file blocks were damaged or not by computing tokens dynamically as follows:

When TPA challenges or requests a block with block indices, cloud server receives this input and it computes the token of that particular block and sends the short signature to TPA. Upon receiving the signature TPA verifies it with the existing token signature. The result of two tokens are same means the block remains same without any effect, otherwise TPA assumes block was modified and it generates a message to cloud server to perform block recovery operation using distributed schemes and erasure coded techniques

4. RELATED DISCUSSIONS

In this paper, we mainly identify the algorithm for token generation in a different way with user parameters. Here this algorithm also results in a good performance and we just cover the topics in brief. Many algorithms and many approaches are there to perform computations. But our scheme is provided with very less over head. If there is any change in the block, the block will not give the same token value. Here we are encrypting two times firstly with user parameters and Token key is called as fileToken for entire file and finally follows bit permutation approach. Homomorphic technique ensures that

we need not decrypt the key for data checking instead we can compare directly with encrypted token. The fileToken is always present for each file of each user so there will be absolutely no collisions among different users of the same file. We were not mentioned split function here but this function gives the key in very less size with effective mathematical calculations. The secret matrix which we were considered contains the set of special symbols and this vector is randomized for every file.

5.CONCLUSION

- In this paper, we ensure that the data which was sent to the cloud servers(CSP) are acknowledged by generating the token dynamically .
- We will combine the data file with file tokens to send the file from cloud client to TPA.
- Block storage on clouds will give better performance and we can easily distribute the blocks to different cloud servers for more data availability .
- Block modifications can be easily done simply by calling each block with indices along with the token

6.REFERENCES

1.) Towards Secure and dependable storage service in cloud computing by cong wang , Qian wang, Kui Ren , Ning Cao, Winjing Lou

2.) **"**Ensuring Data Storage security in cloud computing" by C Wang , Q Wang

3.) "Auditing to keep online storage service honest " by M.A. Shah , R Swaminathan.

4.) Amazon online shopping web services

url link : http://aws.amazon.com