Volume 3, Issue 11, November 2013 ISSN: 2277 128X

International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

Secure Services for Efficient Online Data Storage Using Cloud Computing

R.Jeena
Research Scholar, Department of CSE
Veltech Dr.RR & Dr.SR Technical University, India

Dr.S.Saravana Kumar
Professor, Department of IT
Panimalar Institute of Technology, India

Abstract-For some computer owners, finding enough storage space to hold all the data they've acquired is a real challenge. Some people invest in larger hard drives. Others prefer external storage devices like thumb drives or compact discs. Desperate computer owners might delete entire folders worth of old files in order to make space for new information. But some are choosing to rely on a growing trend: cloud storage. On the surface, cloud storage has several advantages over traditional data storage. For example, if you store your data on a cloud storage system, you'll be able to get to that data from any location that has Internet access. You wouldn't need to carry around a physical storage device or use the same computer to save and retrieve your information. With the right storage system, you could even allow other people to access the data, turning a personal project into a collaborative effort. It is no secret that cloud computing is becoming more and more popular today and is ever increasing in popularity with large companies as they share valuable resources in a cost effective way. Due to this increasing demand for more clouds there is an ever growing threat of security becoming a major issue. This paper shall look at ways in which security threats can be a danger to cloud computing and how they can be avoided.

Keywords- Data integrity, dependable distributed storage, error localization, data dynamics, Virtualization

I. INTRODUCTION

Cloud Computing implies that your customer information is exchanged via the internet to qualify for various web services, and involves a serious danger in terms of security. Data on the Internet are highly susceptible, while safer when stored in Home/Office on storage media. It is possible to corrupt irreparably, sensitive information, caused by the death of servers, or the worst of an entire data center. Failure prevention measures at the level of suppliers of services related to Cloud Computing presume interconnection, data encryption and network servers for periodical backups of the same files on multiple machines. While cloud storage sounds like it has something to do with weather fronts and storm systems, it really refers to saving data to an off-site storage system maintained by a third party. Instead of storing information to your computer's hard drive or other local storage device, you save it to a remote database. The Internet provides the connection between your computer and the database. The two biggest concerns about cloud storage are reliability and security. Clients aren't likely to entrust their data to another company without a guarantee that they'll be able to access their information whenever they want and no one else will be able to get at it.

II. SECURITY THREATS

To secure data, most systems use a combination of techniques, including:

- Encryption, which means they use a complex algorithm to encode information. To decode the encrypted files, a user needs the encryption key. While it's possible to crack encrypted information, most hackers don't have access to the amount of computer processing power they would need to decrypt information.
- Authentication processes, which require creating a user name and password.
- Authorization practices -- the client lists the people who are authorized to access information stored on the cloud system. Many corporations have multiple levels of authorization. For example, a front-line employee might have very limited access to data stored on a cloud system, while the head of human resources might have extensive access to files.
- The relative infancy of cloud based computing services, there uncertainty about the level of information security offered by these services. Infrastructure-as-a-service (IaaS) cloud services are largely reliant on virtualization technology, which is seen as providing all the security and process isolation a customer might want. Multi-tenancy and virtualization enable an efficient computing model. Multi-tenancy allows multiple tenants to coexist in the same physical machine sharing its resources (CPU, memory, network...) and, at the same time, creates an isolated environment for each one.
- Virtualization allows multiple operating systems (OS) to run on the same physical device at the same time. This allows several users to execute their applications on the same physical environment, but isolated from each other. This paper will summarize in the area of cloud security, with a focus on virtualization security.

III. LITERATURE SURVEY

Cloud computing has been envisioned as the next generation architecture of the IT enterprise due to its long list of unprecedented advantages in IT: on demand self-service, ubiquitous network access, location-independent resource pooling, rapid resource elasticity, usage-based pricing, and transference of risk. One fundamental aspect of this new computing model is that data is being centralized or outsourced into the cloud. From the data owners' perspective, including both individuals and IT enterprises, storing data remotely in a cloud in a flexible on-demand manner brings appealing benefits: relief of the burden of storage management, universal data access with independent geographical locations, and avoidance of capital expenditure on hardware, software, personnel maintenance, and so on.

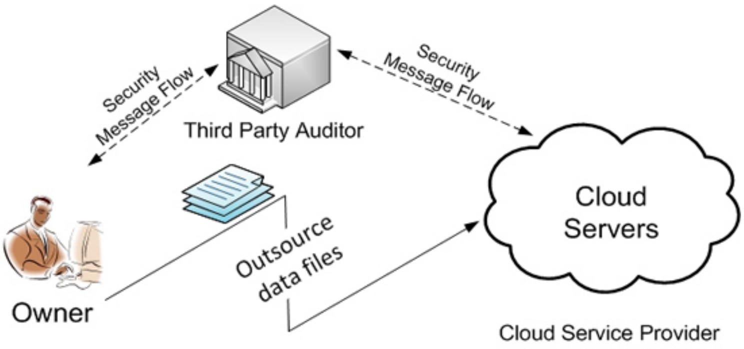

Fig.1 Cloud Data Storage Scenario

several trends are opening up the era of Cloud Computing, which is an Internet-based development and use of computer technology. The ever cheaper and more powerful processors, together with the software as a service (SaaS) computing architecture, are transforming data centers into pools of computing service on a huge scale. 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.

Moving data into the cloud offers great convenience to users since they don't have to care about the complexities of direct hardware management. The pioneer of Cloud Computing vendors, Amazon Simple Storage Service (S3) and Amazon Elastic Compute Cloud (EC2) are both well known examples. While these internet-based online services do provide huge amounts of storage space and customizable computing resources, this computing platform shift, however, is eliminating the responsibility of local machines for data maintenance at the same time. As a result, users are at the mercy of their cloud service providers for the availability and integrity of their data.

On the one hand, although the cloud infrastructures are much more powerful and reliable than personal computing devices, broad range of both internal and external threats for data integrity still exist. Examples of outages and data loss incidents of noteworthy cloud storage services appear from time to time. On the other hand, since users may not retain a local copy of outsourced data, there exist various incentives for cloud service providers (CSP) to behave unfaithfully towards the cloud users regarding the status of their outsourced data. For example, to increase the profit margin by reducing cost, it is possible for CSP to discard rarely accessed data without being detected in a timely fashion. Similarly, CSP may even attempt to hide data loss incidents so as to maintain a reputation. Therefore, although outsourcing data into the cloud is economically attractive for the cost and complexity of long-term large-scale data storage, its lacking of offering strong assurance of data integrity and availability may impede its wide adoption by both enterprise and individual cloud users.

In order to achieve the assurances of cloud data integrity and availability and enforce the quality of cloud storage service, efficient methods that enable on-demand data correctness verification on behalf of cloud users have to be designed. However, the fact that users no longer have physical possession of data in the cloud prohibits the direct adoption of traditional cryptographic primitives for the purpose of data integrity protection. Hence, the verification of cloud storage correctness must be conducted without explicit knowledge of the whole data files. Meanwhile, cloud storage is not just a third party data warehouse.

The data stored in the cloud may not only be accessed but also be frequently updated by the users, including insertion, deletion, modification, appending, etc. Thus, it is also imperative to support the integration of this dynamic feature into the cloud storage correctness assurance, which makes the system design even more challenging. Last but not the least, the deployment of Cloud Computing is powered by data centers running in a simultaneous, cooperated and distributed manner. It is more advantages for individual users to store their data redundantly across multiple physical servers so as to reduce the data integrity and availability threats. Thus, distributed protocols for storage correctness assurance will be of most importance in achieving robust and secure cloud storage systems. However, such important area remains to be fully explored in the literature.

IV. PROBLEM STATEMENT

A. System Model

Representative network architecture for Cloud Data Storage Scenario is illustrated in Fig. 1. Three different network entities can be identified as follows: User: an entity, who has data to be stored in the cloud and relies on the cloud for data storage and computation, can be either enterprise or individual customers.

- Cloud Server (CS): an entity, which is managed by cloud service provider (CSP) to provide data storage service and has significant storage space and computation resources (we will not differentiate CS and CSP hereafter).
- Third-Party Auditor: an optional TPA, who has expertise and capabilities that users may not have, is trusted to assess and expose risk of cloud storage services on behalf of the users upon request. In cloud data storage, a user stores his data through a CSP into a set of cloud servers, which are running in a simultaneous, cooperated, and distributed manner. Data redundancy can be employed with a technique of erasure correcting code to further tolerate faults or server crash as user's data grow in size and importance. Thereafter, for application purposes, the user interacts with the cloud servers via CSP to access or retrieve his data. In some cases, the user may need to perform block level operations on his data. The most general forms of these operations we are considering are block update, delete, insert, and append.

Note that in this paper, we put more focus on the support of file-oriented cloud applications other than nonfile application data, such as social networking data. In other words, the cloud data we are considering is not expected to be rapidly changing in a relative short period.

B. Design Goals

To ensure the security and dependability for cloud storage under the aforementioned adversary model, we to design efficient mechanisms for dynamic data verification and operation and achieve the following goals:

1. Storage correctness: to ensure users that there are indeed stored appropriately and kept intact the time in the cloud.
2. Fast localization of data error: to effectively the malfunctioning server when data corruption been detected.
3. Dynamic data support: to maintain the same level storage correctness assurance even if users modify, delete, or append their data files in the cloud.
4. Dependability: to enhance data availability against Byzantine failures, malicious data modification server colluding attacks, i.e., minimizing the brought by data errors or server failures.
5. Lightweight: to enable users to perform storage correctness checks with minimum overhead.

V. SYSTEM FORMATION

A. Putting Server Access Restore Point and Time Management

To avoid server failure in every data inclusion or any other activity by unauthorized person or any internal and external attack coming into the CSP address domain, one server access point or restore point is given to the cloud server when the client is doing some delete, modification, and append operations in his will. This is a new technique that we have introduced in our system design. The reason for which we want to do this restore point is that if there is any server failure and crash then, this restore access point helps a lot to recover everything that we have lost before that crash time itself.

This process is proceeded by fixing one access point to server database (it is done with the help of CSP, because we don't have any copy of our outsourced data) when we finally finish our data exchange in cloud server. Also since we don't have any physical possession of our data in cloud server we can‟t have any separate login access using cryptographic key [8] for that. It is a major drawback to our cloud server to maintain our individual or group data. In the days to come, it can be rectified with the help of CSP. Here one time management is fixed in cloud database management with the help of CSP's access or automatically in server failure when user stores, deletes, modifies and appends the data and this time management is used to know the restore access point easily by CSP and clients when servers meet such failure.

B. File Distribution Preparation

1. File Encode

It is well known that erasure-correcting code may be used to tolerate multiple failures in distributed storage systems. In cloud data storage, we rely on this technique to disperse the data file F redundantly across a set of n = m + k distributed servers. An (m, k) Reed-Solomon erasure-correcting code is used to create k redundancy parity vectors from m data vectors in such a way that the original m data vectors can be reconstructed from any m out of the m + k data and parity vectors. By placing each of the m + k vectors on a different server, the original data file can survive the failure of any k of the m + k servers without any data loss, with a space overhead of m + k. For support of efficient sequential I/O to the original file, our file layout is systematic, i.e., the unmodified m data file vectors together with k parity vectors is distributed across m + k different servers.

Let F = (F1, F2, ….. Fm) and $F_i = (F1i, F2i, \ldots, Fli)^T$ Here, T (shorthand for transpose) denotes that each F_i is represented as a column vector, and *l* denotes data vector size in blocks. All these blocks are elements of $GF(2^p)$. The systematic layout with parity vectors is achieved with the information dispersal matrix A, derived from an *m × (m + k)* Vander monde matrix.

$$\begin{pmatrix} 1 & 1 & . & . & 1 & 1 & . & . & 1 \\ \beta_1 & \beta_2 & \cdot & \cdot & \beta_m & \beta_{m+1} & \cdot & \cdot & \beta_n \\ . & . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . & . \\ \beta_1^{m-1} & \beta_2^{m-1} & \cdot & \cdot & \beta_m^{m-1} & \beta_{m+1}^{m-1} & \cdot & \cdot & \beta_n^{m-1} \end{pmatrix}$$

where $\beta_{j\ (j \in \{1,,,n)\})}$ are distinct elements randomly picked from $GF(2^{Þ})$. After a sequence of elementary

row transformations, the desired matrix A can be derived as

$$A = (I|P) = \begin{pmatrix} 1 & 0 & \dots & & 0 & p11 & p12 & \dots & p1k \\ 0 & 1 & \dots & & 0 & p21 & p22 & \dots & p2k \\ . & . & \dots & . & . & & . & \dots & . \\ . & . & \dots & . & . & & . & \dots & . \\ 0 & 0 & \dots & 1 & & p_{m1} & p_{m2} & \dots & p_{mn} \end{pmatrix}$$

Where, I is an m x m identity matrix and P is the secret parity generation matrix with size m x k. Note that A is derived from a Vander monde matrix, thus it has the property that any m out of the m + k columns form an invertible matrix. By using F and P, the user obtains the encoded file R = F X P, Where R is a matrix of order m x k that is derived by multiplying F and P.

$G = (F_1, F_2, \ldots.. F_m, R_1, R_2, \ldots R_k)$ Where, F_i denotes data blocks, $1 \leq i \leq m$

R_j denotes j^{th} column of matrix R, $1 \leq j \leq k$

2. Token Precomputation

In order to achieve assurance of data storage correctness and data error localization simultaneously, our scheme entirely relies on the precomputed verification tokens. The main idea is as follows: Before file distribution the user precomputes a certain number of short verification tokens on individual vector, $G^{(j)}$ $(j \in \{1\ldots n)\})$ where, j = 1 to n, each token covering a random subset of data blocks. Later, when the user wants to make sure the storage correctness for the data in the cloud, he challenges the cloud servers with a set of randomly generated block indices. Upon receiving challenge, each cloud server computes a short "signature" over the specified blocks and returns them to the user. The values of these signatures should match the corresponding tokens precomputed by the user. Meanwhile, as all servers operate over the same subset of the indices, the requested response values for integrity check must also be a valid codeword determined by the secret matrix P.

Suppose the user wants to challenge the CS for „t‟ times to ensure the correctness of data storage, then, he must compute t verification tokens for each $G^{(j)}$ $(j \in \{1\ldots n)\})$, using a PRF *f(.)* , a PRP Ø(.), a challenge key k_{chal}, and a master permutation key K_{PRP}. Specifically, to generate the i^{th} token for the server j, the user acts as follows:

1. Derive a random challenge value αi of $GF^{(}2^{p)}$ by $\alpha_{i=}\ f_{\ kchal}\ ^{(i)}$ and a permutation key $kprp^{(i)}$ based on $K_{PRP.}$
2. Compute the set of r randomly-chosen indices $\{I_q \in [1,,,,,,, l]|1 \leq q \leq r\}$, where $I_q = Ø_{k\ pr\ p}{}^{(i)\ (q)}$
3. Calculate the token as

$$v_i{}^{(j)\,=}\ \sum_{q=1}^{r} q\alpha_i + G^{(j)}\,[I_q],$$

Where $G^{(j)}\,[I_q] = g_{Iq}{}^{(j)}$

Note that $v_i{}^{(j)}{}_{,}$ which is an element of $GF(2^p)$ with small size, is the response the user expects to receive from server j when he challenges it on the specified data blocks. After token generation, the user has the choice of either keeping the precomputed tokens locally or storing them in encrypted form on the cloud servers. In our case here, the user stores them locally to obviate the need for encryption and lower the bandwidth overhead during dynamic data operation which will be discussed shortly. The details of token generation are shown in the following algorithm.

```
ALGORITHM: Token_Precomputation (G)
{
   /*
    INPUT:
    G – Encoded File.
    $K_{chal}$ – A challenge key.
    $K_{prp}$ – Key for a master permutation. n – Number of distributed servers.
    t – Number of tokens. FUNCTIONS:
    f () – Pseudorandom Function (PRF)
    Ø () – Pseudorandom Permutation (PRP) OUTPUT:
    $V_i$ – Tokens.
   */
    for    j=1 to n do  // for each vector
    {
         for i=1 to t do    //for each round
    {
        $\alpha_i = f_{kchal}^{(i)}$ and $kprp^{(i)}$
                        r
     $v_i^{(j)} = \Sigma q\alpha_i + G^{(j)} [Ø_{k\ pr\ p}^{(i)(q)}]$
      q=1
      }
    }
  }
```

Once all tokens are computed, the final step before file distribution is to blind each parity block $g_i^{(j)}$ in $(G^{(m+1)},\ldots\ldots, G^{(n)})$ by

$g_i^{(j)} \leftarrow g_i^{(j)} + f_{kj}(s_{ij})$, $i \in \{1\ldots\ldots l\}$,

Where k_j is the secret key for parity vector $G^{(j)}$, ($j\in \{m+1,\ldots\ldots, n\}$). This is for protection of the secret matrix P. After blinding the parity information, the user disperses all the n encoded vectors $G^{(j)}$ across the cloud servers $S_1,S_2, \ldots ,S_n$.

For verifying Correctness and localizing errors the algorithm CHALLENGE () that is proposed in [8] can be used.

3. File Renovation

The user can reconstruct the original file by downloading the data vectors from the first m servers, assuming that they return the correct response values. That our verification scheme is based on random spot-checking, so the storage correctness assurance is a probabilistic one.

VI. DATA BLOCK DYNAMIC PROCESS

In our proposed system we can ensure our data will be protected by doing the following operations such as append, deletion, and update. These are some operations performed by a user in the cloud environment.

A. Append Operation in Server Block

In this operation we assume that a user having some GB of memory space allotted by a service provider for the user application purposes. CSP will not restrict a user to access for particular applications and user can access any kind of applications also, the service provider is to provide all kind of access to a user. The user can include some data with existing data, change some data or remove some data because of this we can give assurance to the data integrity[3]. It is very efficient method in our proposed design compared to any other such type.

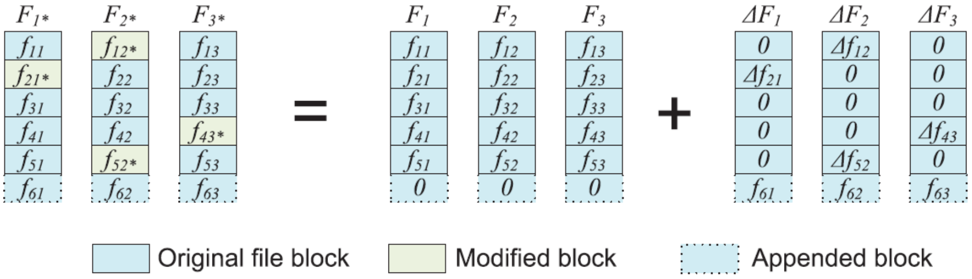

Fig. 2: Representation of Data Block

B. Deletion Operation in Server Block

In this deletion operation, the user has to compare his data with the existing servers‟ data. After this process, user may perform the deletion operation which is shown in Figure 2. If there is number of servers for data selection while deleting the data that are stored in the particular storage server is considered for this. The required data can be deleted [8].

C. Update Operation in Server Block

In cloud data storage, a user may need to modify some data block(s) stored in the cloud, from its current value fij to a new one, fij + fij. This operation is referred as data update. Figure 2 gives the high level logical representation of data block update. Due to the linear property of Reed-Solomon code, a user can perform the update operation and generate the updated parity blocks by using fij alone, without involving any other unchanged blocks.

VII. EFFICIENCY ANALYSIS

A. *Efficiency of File Encoding*

The content of file F is encoded and stored in G by adding the first m columns of F to G and the next „k‟ columns are multiplied with the matrix P and appended in G.

Time taken to multiply $F_{m+1\cdots n}$ with P is calculated as M (m).

M (m) – Number of multiplications needed to multiply $F_{m+1\cdots n}$ by P.

M (m) $\in$ O (m^2) for small values of k.

M (m) $\in$ O (m^3) for k which is equal to m.

B. Efficiency of Token Precomputation

To distribute the file in cloud environment vectors are generated and tokens are generated for each vector. We propose a method to increase the efficiency of the algorithm that is proposed in [8] for token precomputation.

VIII. CONCLUSION

In this paper, we analyze the problem of data security in cloud data storage, which is essentially a distributed storage system. To achieve the assurances of cloud data integrity and availability and enforce the quality of dependable cloud storage service for users, we propose an enhanced effective and flexible scheme with unambiguous dynamic data support, including block update, delete, and append. We rely on erasure-correcting code in the file distribution preparation to provide redundancy parity vectors and guarantee the data reliability. By utilizing the homomorphic token with distributed verification of erasure coded data, our scheme achieves the integration of storage correctness insurance and data error localization.

ACKNOWLEDGMENTS

The authors would like to thank the editor and anonymous reviewers for their valuable suggestions that significantly improved the quality of this paper.

REFERENCES

[1] G.Ateniese, R.D. Pietro, L.V. Mancini, and G. Tsudik, “Scalable and Efficient Provable Data Possession,” Proc. Fourth Int Conf.Security and in Comm. Networks (SecureComm "08), pp. 1-10, 2008.

[2] L. Carter and M. Wegman, “Universal Hash Functions,” J. Computer and System Sciences, vol. 18, no. 2, pp. 143-154, 1979.

[3] C. Erway, A. Kupcu, C. Papamanthou, and R. Tamassia, “Dynamic Provable Data Possession,” Proc. 16th ACM Conf. Computer and Comm. Security (CCS "09), pp. 213-222, 2009.

[4] J. Hendricks, G. Ganger, and M. Reiter, “Verifying Distributed Erasure-Coded Data,” Proc. 26th ACM Symp. Principles of Distributed Computing, pp. 139-146, 2007.

[5] T. Schwarz and E.L. Miller, “Store, Forget, and Check: Using Algebraic Signatures to Check Remotely Administered Storage,” Proc. IEEE Int"l Conf. Distributed Computing Systems (ICDCS "06), pp. 12-12, 2006.

[6] 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.

[7] Q.Wang, C. Wang, J. Li, K. Ren, and W. Lou, “Enabling Public Verifiability and Data Dynamics for Storage Security in Cloud Computing,” Proc. 14th European Conf. Research in Computer Security (ESORICS 09), pp. 355-370, 2009.

[8] C.Wang Q.Wang, K.Ren, Ning Cao, and W.Lou, “Toward Secure and Dependable Storage Services in Cloud Computing” IEEE Services Computing, Vol. 5, No. 2, April-June 2012.

[9] Q. Wang, K. Ren, W. Lou, and Y. Zhang, “Dependable and secure sensor data storage with dynamic integrity assurance,” in *Proc. Of IEEE INFOCOM’09*, Rio de Janeiro, Brazil, Appril 2009, pp. 954– 962.