ACEEE Int. J. on Network Security , Vol. 5, No. 1, January 2014

Enhancing Data Storage Security in Cloud Computing Through Steganography

Mrinal Kanti Sarkar[1], Trijit Chatterjee[2]
[1] Department of Computer Science & Engineering
University of Engineering and Management, Jaipur, India
Email: mks08iitkgp@gmail.com
[2]Department of Computer Science & Engineering
MCKV Institute of Engineering, Howrah, India
Email: trijitchatterjee@gmail.com

***Abstract*—in cloud computing data storage is a significant issue because the entire data reside over a set of interconnected resource pools that enables the data to be accessed through virtual machines. It moves the application software's and databases to the large data centers where the management of data is actually done. As the resource pools are situated over various corners of the world, the management of data and services may not be fully trustworthy. So, there are various issues that need to be addressed with respect to the management of data, service of data, privacy of data, security of data etc. But the privacy and security of data is highly challenging. To ensure privacy and security of data-at-rest in cloud computing, we have proposed an effective and a novel approach to ensure data security in cloud computing by means of hiding data within images following is the concept of steganography. The main objective of this paper is to prevent data access from cloud data storage centers by unauthorized users. This scheme perfectly stores data at cloud data storage centers and retrieves data from it when it is needed.**

***Index Terms*—Cloud Computing, Data Storage Security, Steganography.**

I. Introduction

Cloud computing is defined as a model for enabling ubiquitous, convenient, cheapest, on-demand network access to a shared pool of configurable computing resources (e.g. networks, servers, storage devices and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction [1]. In such an environment users need not own the infrastructure for various computing services. In fact they can be accessed their data from any computer from any part of the world. It can allocate or reallocate resources dynamically with an ability to continuously monitor their performance [1]. It is now a challenging way where we can share data, information, and knowledge. The benefits of cloud computing are many. One is reduced cost, since you pay as you go. The ultimate goal is allowing customer to run their everyday IT infrastructure in cloud.

In the cloud computing many services are provided to the client by the cloud. Storing of data is the main features that the cloud service provider provides to the client companies or any other users. They can store their huge amount of data in cloud data storage centers. But many clients are not ready to implement cloud computing technology due to the lack of proper security control policy and weakness in protection of data which leads to a big challenge for the cloud computing providers. The pioneer of cloud computing vendors, Amazon Simple Storage Services (S3) and Amazon Elastic Compute Cloud (EC2) [2] are well known example. Amazon S3 provides a simple web services interface that can be used to store and retrieve any amount of data, at any time, from anywhere on the web. It also allows developer to access the highly scalable, reliable, secure, fast, inexpensive infrastructure that Amazon uses to run its own global network web services.

From the viewpoint of data security which has been an important aspect of quality of services, cloud computing unavoidably poses new challenging security threats for number of reasons. Firstly, we cannot adopt the traditional cryptographic primitives for the purpose of data security in cloud computing as the user' loss their data control. So, we need a data verification strategy but without explicit knowledge of the whole data, it is very hard to verify the correct data. Considering various kinds of data for each user, stored in the cloud and demand of the long term continuous assurance of their data safety, the problem of verifying correctness of data storage in the cloud becomes even more challenging. Secondly, it is not a just third-party data warehouse. The data stored in the cloud may be frequently updated by the user, including insertion, deletion, modification, appending, recovering, etc. So, for this dynamic operation, it needs to be more advanced technology to prevent data loss from the cloud data storage centers. Last but not the least data centers are running in a simultaneously, cooperated and in distributed manner [3]. Every user' data is stored in multiple physical locations randomly. Therefore distributed protocols for storage correctness assurance will be most importance in achieving a robust and secure cloud data storage system in the real word.

In this paper, we propose an effective and flexible data hiding scheme with explicit dynamic data support to ensure the security of data when it is residing in the cloud data storage. We enhanced the security of data to store it into an image. When these images are stored in the cloud data centre, no one can view the original content of the data without any proper identification. Our scheme almost guarantees the security of data when it is residing on the data center of any

Cloud Service Provider (CSP).

According to our survey, our work is debutant in this field to store data in data storage centers in the form of images. Our contribution summarized as the following aspects: 1) Compared to many of its predecessors, which only store data in raw format, but in our scheme we are storing data into images, and 2) this new scheme support secure and efficient data storage and retrieval operation. The rest of the paper is organized as follows. In Section II, we describe the cloud architecture and security issues. Section III introduces the system architecture, security model, our design goal and notations. Then we provide the detailed description of our scheme in Section IV. Section V gives the security analysis and performance evaluations, followed by Section VI overviews the related work. Finally, we give the concluding remarks of the whole paper and future scope in Section VII.

II. Cloud Architecture and Security Isssues

Cloud computing can be broadly classified in two categories, the service delivery models and the deployment models [1]. The deployment models are: 1) **Private cloud**: a cloud platform is dedicated for specific organizations; all of these services are deployed through a privately owned data center used exclusively by the organization that builds it. These private clouds may deploy proprietary technologies inaccessible to other users of the cloud services, 2) **Public cloud**: a cloud platform available to public user to register and use the available infrastructure. Multiple enterprises can work on the same infrastructure, at the same time. User can dynamically provision recourses through the internet from an off-site service provider, 3) **Hybrid cloud**: a private cloud that can extend to use resources in public clouds. It focuses to create an environment in which an organization provides and manages some resources in-house and has others provided externally. It combines public cloud and private cloud data center principles, it's possible to plan a hybrid cloud deployment from either of these starting points and 4) **Community Cloud:** a community cloud in computing is a collaborative effort in which infrastructure is shared between several organizations from a specific community with common concerns (security, compliance, jurisdiction, etc.), whether managed internally or by a third-party and hosted internally or externally. Public clouds are the most vulnerable deployment model because they are available for public users to host their services, they may be malicious users.

The service delivery model is organized into three layers, Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS).

Infrastructure as a Service (IaaS) is the lowest layer that provides the basic infrastructure as a support. It basically refers to the sharing of the hardware resources for executing services, typically including virtualization technology. The cloud consumer has the provision for processing, storage, networks etc and to deploy and run arbitrary software supported by the operating system run by the virtual machine.

Platform as a Service (PaaS) is the middle layer which provides platform oriented services, besides providing the environment for software execution. It aims to protect data in storage. It delivers platforms, tools and other business services that enable customer to develop, deploy, and mange their own application, without installing any of these platforms and support tools on their local machines.

Software as a Service (SaaS) is the topmost layer which features a complete application offered as a service on demand. It deliver application hosted on the cloud infrastructure as internet based service for end user without requiring installing the application on the customer's computers. It ensures that the complete applications are hosted on the internet and users use them.

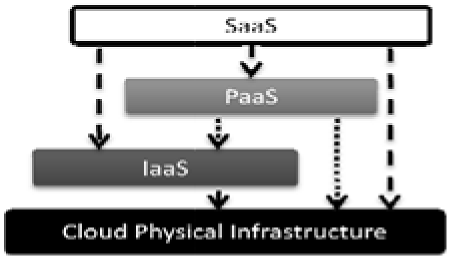


Figure 1: Cloud service delivery model

Each service delivery model has different possible implementations, as in Figure 1, which complicates the development of standard security model for each service delivery model. Moreover, these service delivery models may coexist in one cloud platform the security management process.

It is clear that the security issue has played the most important role in hindering cloud computing acceptance. Without doubt, putting your data, running your software on someone else's hard disk using someone else's CPU appears daunting to many. Well-known security issues such as data loss, phishing, and botnet (running remotely on a collection of machines) pose serious threats to organization's data and software. Moreover, the multi-tenancy model and the pooled computing resources in cloud computing has introduced new security challenges that require novel techniques to tackle with. For example, hackers can use cloud to organize botnet as cloud often provides more reliable infrastructure services at a relatively cheaper price for them to start an attack [4].

III. Problem Statement

Managing data-at-rest plays a crucial role in cloud computing. The main issue with data-at-rest in cloud computing is loss of control; even an unauthorized user may have access the data in a shared environment. However, now-a-days storage devices are powered by encryption methodologies which restrict unauthorized access to data to share extents. If the encryption and

decryption keys are accessible to malicious users encryption methodologies fails to provide authorized access. Another approach to provide security in data-at-rest is to hide data behind images, following the concept of stenography. This paper aims to provide a better security through steganography.

A. *System Architecture*

Schematic network architecture for our proposed model for cloud data storage is illustrated in Figure 2. In this architecture, we have different network entities which can be identified as follows:

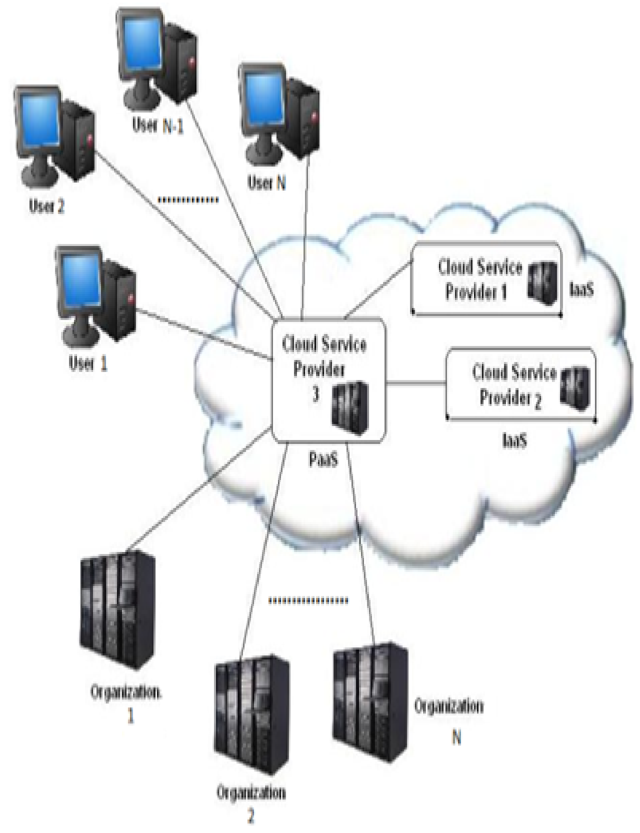


Figure 2: Architecture of cloud data security model

• **User:** Users or organizations who have their data for storage in the cloud and rely on the cloud for data computation.

• **Cloud Service Provider-1(CSP-1):** A cloud service provider which have significant resources of grayscale image of various sizes, a database which maintains records of file names, number of characters present etc.

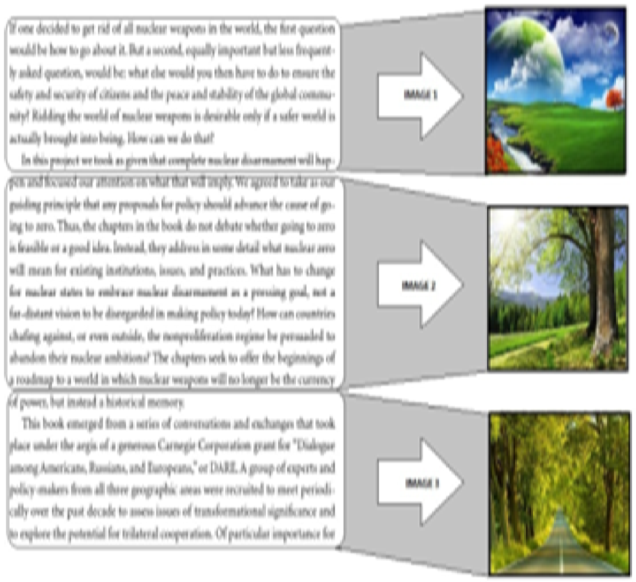
• **Cloud Service Provider-2(CSP-2):** Another cloud service provider, which stores a mechanism or algorithm for data hiding within images and retrieving data from images.

• **Cloud Service Provider 3(CSP-3):** Another Cloud Service which is connected to CSP-1 and CSP-2. All the computations that will be carried out by the user, will take place in CSP-3. According to the requirement, CSP-3 will interact with CSP-1 and CSP-2.

B. *Security Model*

Our proposed model aims to secure data-at-rest, not by physically storing files, instead of the data present in a files are somehow stored within some images. This underlining concept is known as steganography which tells- "The art and science of writing hidden messages in such a way that no one apart from the sender and the intended recipient, suspects the existence of the message, a form of security through obscurity".

Figure-3 shows whole file is partitioned into three parts and data within each portion is saved into the corresponding images. The number of images would be varying depending on the size of data files. The partitioning of a file depends on the size of images and the contents of the file.


If one decided to get rid of all nuclear weapons in the world, the first question would be how to go about it. But a second, equally important but less frequently asked question, would be: what else would you then have to do to ensure the safety and security of citizens and the peace and stability of the global community? Ridding the world of nuclear weapons is desirable only if a safer world is actually brought into being. How can we do that?

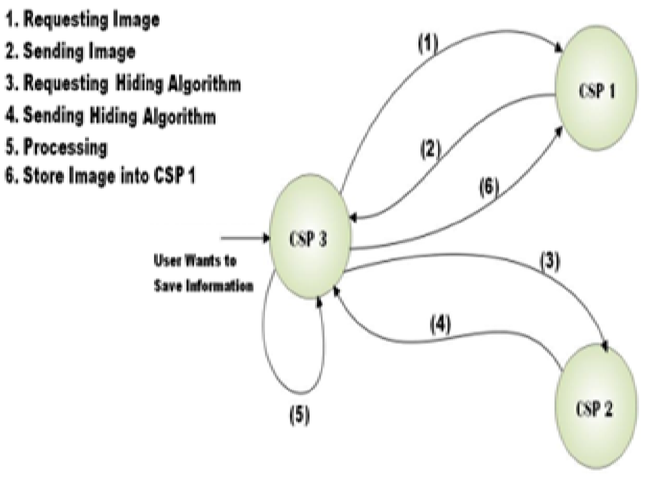
In this project we took as given that complete nuclear disarmament will happen and focused our attention on what that will imply. We agreed to take as our guiding principle that any proposals for policy should advance the cause of going to zero. Thus, the chapters in the book do not debate whether going to zero is feasible or a good idea. Instead, they address in some detail what nuclear zero will mean for existing institutions, issues, and practices. What has to change for nuclear states to embrace nuclear disarmament as a pressing goal, not a far-distant vision to be disregarded in making policy today? How can countries chafing against, or even outside, the nonproliferation regime be persuaded to abandon their nuclear ambitions? The chapters seek to offer the beginnings of a roadmap to a world in which nuclear weapons will no longer be the currency of power, but instead a historical memory.

This book emerged from a series of conversations and exchanges that took place under the aegis of a generous Carnegie Corporation grant for "Dialogue among Americans, Russians, and Europeans," or DARE. A group of experts and policy makers from all three geographic areas were recruited to meet periodically over the past decade to assess issues of transformational significance and to explore the potential for trilateral cooperation. Of particular importance for

Figure 3: Mapping from Data into Images

The security model shown in Figure 4 and Figure 5 give a detailed description of how the system will be worked. The users will perform their computations in CSP-3. Whenever user wants to save their data, the following operation will be happen:

1. CSP-3 requests set of images from CSP-1.
2. CSP-1 acknowledges CSP-3 by providing a set of valid images.
3. CSP-3 requests the data hiding algorithm which is stored in CSP-2.
4. CSP-2 provides the algorithm to CSP-3.
5. According to the algorithm the data are saved within the pixels of the images which are taken from the CSP-1.
6. The images containing the data will be stored in CSP-1.



1. Requesting Image
2. Sending Image
3. Requesting Hiding Algorithm
4. Sending Hiding Algorithm
5. Processing
6. Store Image into CSP 1
(1)
(2)
(6)
(3)
(4)
(5)
CSP 1
CSP 3
CSP 2
User Wants to Save Information

Figure 4: Processing model to Store Data

Later whenever user wants to view or retrieve his/her data the following mechanisms will be followed:

1. CSP-3 requests set of images which are associated with the file that users want to view/retrieve.
2. CSP-1 acknowledges to CSP-3 by providing the associated set of images.
3. CSP-3 requests the data retrieval algorithm which is stored in CSP-2.
4. CSP-2 provides the algorithm to CSP-3.
5. CSP-3 processes the algorithm on the images which are provided by CSP-1, and store the data after retrieval into a temporary file. This file is displayed to the user and after any operation on this data; it will be deleted from it.

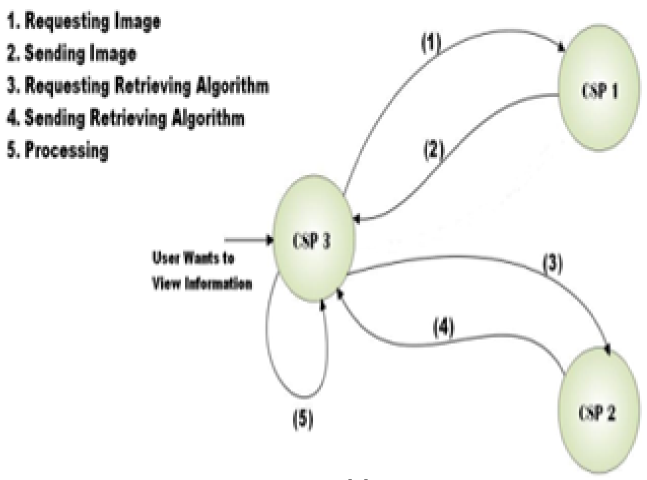

1. Requesting Image
2. Sending Image
3. Requesting Retrieving Algorithm
4. Sending Retrieving Algorithm
5. Processing
(1)
(2)
(3)
(4)
(5)
CSP 1
CSP 2
CSP 3
User Wants to View Information

Figure 5: Processing model to Retrieve Data

Before user log-out from CSP-3, the temporary file will be automatically de-allocated from the system.

C. Design Goal

To ensure the security for cloud data storage from unauthorized users, we have designed efficient mechanism for data-at-rest in cloud data storage centers. As our used techniques highlights hiding data within digital image, this steganography technique exploits the weakness of Human Visual System (HVS). HVS cannot detect the variation in luminance of grayscale vectors at higher frequency side of the visual spectrum. An image is a collection of pixels (Picture Elements), where each pixel of grayscale image is composed of 8 bits. If we change the last bit, the color information may be varying within +1 to -1. This change of the intensity will not be perceived by human eye. Now data consist of characters representing ASCII values. Our idea is to store each character into the last bit of consecutive 8 pixels. So by storing data within images, which are located in remote cloud servers, we can achieve the following goals.

• Correctness: Data can be stored correctly within images.

• Availability: An authorized user can retrieve the information from an image when required.

• Protection: It is fully protected from unauthorized user because, perceiving an image does not provide any idea of the original information.

D. Notation and Preliminaries

• **F**: The data file to be stored in the cloud. We are assuming that the users initially perform their computation in a text file. This file will be de-allocated after storing data into images.

• **F_{Temp}**: It is a temporary file which is used during data retrieval operation and will be de-allocated automatically after use.

• **C_{Count}**: Number of characters present in a file.

• **$Pixel_{Count}$**: Number of pixel present in an image.

• **A_8**: Accumulate 8 bits from the last bits of each pixel.

• **File_Name:** The name of the file where user had performed computation.

• **Img_Index:** The address of a valid image returned by **ImgSearch ().**

• **Img_Index1:** A temporary image address.

• **Img_Database:** It maintains collections of images of random sizes which is available in CSP-1, here each images consists the following attributes:

a. Name of image

b. Valid bit, which represents an image, is available for storing data or not.

c. Address field, which represents the connectivity among the various images to store different consecutive parts of a file.

• **File_Database:** It is a file which signifies on which set of images our data is stored, which is also available in CSP-1. An obvious misconception may arise that the file database contains the actual information, but it is not true. It actually maintains following attributes:

a. Name of file

b. C_{Count} of file.

c. Address field, which represents the address of images associated with each file.

• **ImgSearch ():** It searches for a valid image for storing data & returns the address of the valid image if available.

• **MdfImage ():** It maps data into images. It is available in CSP 2.

RdfImage (): It retrieves original data from an image and it is available in CSP-2.

IV. Enhancing Cloud Data Storage

In cloud computing user store and process their data in the cloud, no longer posses the data locally. Thus security of data that are stored on the distributed cloud server must be guaranteed. The major security concern regarding cloud computing is that, if data is residing in remote servers when it is available in raw format, and then it can be easily accessible and manipulated by unauthorized users. So, our main scheme for ensuring data security in cloud storage signifies to transform data in such a way, such that it can't be traced by unauthorized users. Now, we are outlining the methodologies which we have implemented.

A. Creation of Image Database

We have created a database consisting of images of different sizes which is located in CSP-1. Whenever a user wants to store data, a set of images will be selected from this database and it will be transferred to CSP-3 and all operation will be occurred in CSP-3.

B. Maintenance of File Database

An obvious misconception may be arisen that the file database contains the actual information, but it is not true. It is a file which signifies that on which set of images our data is stored.

C. Hiding Data within Images

This portion deals with the pre-requisite requirements for the steganographic operations which includes a table where attributes like name of file to be saved, total characters presents, address of images etc are present. This process runs in CSP-3, whenever user wants to store a file.

Algorithm 1: Hiding Data within Images

```
1: procedure
2:     Select F;
3:     Compute CCount from F;
4:     Load Img_Index = ImgSearch (Img_Database);
5:     Store File_Database (File_Name,
       CCount, Img_Index);
6:     MdfImage (Img_Database [Img_Index]);
7: end procedure.
```

D. Searching of Image

Here, we have shown how an image can be selected for performing steganographic operations. It will search a valid image and eventually returns the address of the same.

Algorithm 2: ImgSearch ()

```
1: procedure
2:     Open Image_Database;
3:     for Img_Database (i), i?1, n do
4:         if (Img_Database (i).valid==1)
5:             return i;
6:         end if
7:     end for
8: end procedure
```

E. Mapping Data from a File to Image

This mechanism helps us to store raw data within images. This algorithm also dynamically selects a new image using algorithm 2, whenever an image is overflowed with data.

Algorithm 3: MdfImage ()

```
1: procedure
2:     Read Img_Database [Img_Index];
3:     Compute PixelCount for Img_Database
       [Img_Index];
4:     Open F;
5:     while (Read Characters until EOF) do
6:         if (PixelCount < CCount)
7:             Compute ASCII value;
8:             Store each bits of ASCII into consecutive
               8 pixels of the Img_Database
               [Img_Index];
9:         else
10:            Load Img_Index
               1=ImgSearch (Img_Database);
11:            Img_Database [Img_Index 1].valid=
               Img_Index 1;

12:            Img_Database [Img_Index 1].valid=0;
13:        end if
14:    end while
15: end procedure
```

F. Retrieving Data from Image

When an authorized user wants to view data which are already stored in remote cloud servers within images, the following mechanism helps us to retrieve data into human readable format.

Algorithm 4: RdfImage ()

```
1: procedure
2:     Read F;
3:     for File_Database (i), i?1, m do
4:         if (F exits)
5:             I=Addres of image associated with F;
6:         end if
7:     end for
8:     Open Img_Database;
9:     Read Image_Database [I];
10:    Open an FTemp;
11:    while (Until all characters are within in FTemp) do
12:        k=Convert A8 to Character
13:        write k into FTemp;
14:    end while
15: end procedure
```

V. Sequirity Analisis And Performance Evaluation

In this section, we analyze our proposed model in terms of security and efficiency. Our security analysis focuses on the system architecture and its security model defined in section II. We also evaluate the efficiency of our scheme via implementing the file database and image database.

A. Security Strength Against CSP-1

As in CSP-1, we are storing only set of files which contains only the address of valid images (i.e. those which have some information). We have also stored here the set of images which can store the information when the user wants to store the data. So, when the images contain the information, an unauthorized user cannot perceive anything by viewing these images.

B. Security Strength Against CSP-2

In CSP-2, we are storing only the retrieving and hiding mechanism which we will need at the time of viewing and storing the data from CSP-3. So there is nothing to fear from an unauthorized user.

C. Security Strength Against CSP-3

In our proposed model, we have chosen this CSP-3 for computational purpose for any operation i.e. storing data within images and retrieving data from images. After performing the operation all the files are automatically de-allocated from this service provider.

D. Performance Evaluation

We implemented our approach using the concept of steganography. Our experiment is conducted using MATLAB R2012b on a system with Intel(R) Pentium (CPU) running at 2.60 GHz, 2048 MB of RAM, a 7200RPM Western Digital 500 GB Serial ATA drive with an 8MB buffer. It has been found that our implemented system successfully stores data within images and retrieves data from images.

VI. Related Work

Shacham et al. [5] built on this model and constructed a random linear function based homomorphic authenticator which enables unlimited number of queries and requires less communication overhead. Bowers et al. [7] proposed an improved framework for POR protocols that generalizes both Juels and Shacham's work. Later in their subsequent work, Bowers et al. [6] extended POR model to distributed systems. However, all these schemes are focusing on static data. The effectiveness of their schemes rests primarily on the preprocessing steps that the user conducts before outsourcing the data file F. Any change to the contents of F, even few bits, must propagate through the error-correcting code, thus introducing significant computation and communication complexity. Cong Wang et al. [3] use homomorphic token with distributed verification of erasure-coded data towards ensuring data storage security and locating the server being attacked. It support dynamic operation on data blocks such as update, delete and append without data corruption and loss. However, the issues with fine-grained data error location remain to be addressed.

In other related work, Shantanu pal et al. [8] ensures the identification of adversary or the attacking party and helping us find a far off place for an attacking party from its target and hence ensuring a more secure environment for the other VMs. If the adversary gets to know the location of the other VMs, it may try to attack them. This may harm the other VMs in between. Flavio Lombardi et al. [9] show that behavior of cloud components can be monitored by logging and periodic checking of executable system file. But system performance gets marginally degraded and small performance penalty is encountered. Filho et al. [9] proposed to verify data integrity using RSA-based hash to demonstrate escheatable data possession in peer to peer file sharing networks. However, their proposal requires exponentiation over the entire data file, which is clearly impractical for the server whenever the file is large. Shah et al. [10] proposed allowing a TPA to keep online storage honest by first encrypting the data then sending a number of pre computed symmetric-keyed hashes over the encrypted data to the auditor. However, their scheme only works for encrypted files and auditors must maintain long-term state. Schwarz et al. [11] proposed to ensure file integrity across multiple distributed servers, using erasure-coding and block-level file integrity checks.

Ateniese et al. [12] defined the "provable data possession" (PDP) model for ensuring possession of file on untrusted storages. Their scheme utilized public key based homomorphic tags for auditing the data file, thus providing public verifiability. In their subsequent work, Ateniese et al. [13] described a PDP scheme that uses only symmetric key cryptography. This method has lower-overhead than their previous scheme and allows for block updates, deletions and appends to the stored file, which has also been supported in our work. However, their scheme focuses on single server scenario and does not address small data corruptions, leaving both the distributed scenario and data error recovery issue unexplored. Curtmola et al. [14] aimed to ensure data possession of multiple replicas across the distributed storage system. They extended the PDP scheme to cover multiple replicas without encoding each replica separately, providing guarantees that multiple copies of data are actually maintained.

However, we have proposed a new scheme to provide the better security in the world of cloud computing.

VII. Concluion

In this paper, we have investigated the problem of security in cloud computing, which is essentially a distributed storage system. To ensure the security of user' data in cloud storage, we proposed an effective and efficient steganographic strategy for enhancing security on data-at-rest. So, when these images are stored in the cloud data centre, no one can view the original content of the data without any proper identification. Through detailed security and performance analysis, we have seen that our scheme almost guarantees the security of data when it is residing on the data center of any Cloud Service Provider (CSP).

The concept we have discussed here, will help to build a strong architecture for security in the field of cloud computation. This kind of structure of security will also be able to improve customer satisfaction to a great extent and we will attract more investor in this cloud computation concept for industrial as well as future research farms. Security in a very large scale cross cloud environment is an active issue. This present scheme is able to handle only a limited number of security threats in a fairly small environment. We need further simulations to verify the performance. In the future, we will extend our research by providing security through steganography in RGB images. Also, if the raw data is encrypted and the steganographic issues are employed then the protection will be a bit enhanced. The protections can also be enhanced if we can change the pixel positions after steganography. Till now we are working on it to get better performance.

Acknowledgment

We are extremely thankful to Cong Wang, Qian Wang, Kui Ren and Wenjing Lou for their contribution in "Ensuring Data Storage Security in Cloud Computing", 17th Interna

tional workshop on Quality of service, USA, pp1-9, 2009, IBSN: 978-42443875-4. From that we get inspiration towards implementation of this paper.

References

[1] Peter Mell, Timothy Grance, "The NIST Definition of Cloud Computing", Jan, 2011.http://docs.ismgcorp.com /files/ external/Draft-SP-800-145_cloud-definition.pdf.

[2] Amazon.com, "Amazon Web Services (AWS)", Online at hppt://aws.amazon.com, 2008.

[3] Con Wang, Qian Wang, Kui Ren, and Wenjng Lou, "Ensuring Data Storage Security in Cloud Computing", 17th International workshop on Quality of service, USA, pp1-9, 2009, IBSN: 978-42443875-4.

[4] B.P Rimal, Choi Eunmi, I.Lumb, "A Taxonomy and Survey of Cloud Computing System", Intl. Joint Conference on INC, IMS and IDC, 2009, pp.44-51, Seoul, Aug, 2009. DOI: 10.1109/NCM.2009.218.

[5] H. Shacham and B. Waters, "Compact Proofs of Retrievability", Proc. of Asiacrypt '08, Dec. 2008.

[6] K. D. Bowers, A. Juels, and A. Oprea, "HAIL: A High-Availability and Integrity Layer for Cloud Storage," Cryptology ePrint Archive, Report 2008/489, 2008, http:// eprint.iacr.org/.

[7] A. Juels and J. Burton S. Kaliski, "PORs: Proofs of Retrievability for Large Files," Proc. of CCS '07, pp. 584–597, 2007.

[8] Shantanu Pal, Sunirmal Khatua, Nabendu Chaki, Sugata Sanyal, "A New Trusted and Collaborative Agent Based Approach for Ensuring Cloud Security", Annals of Faculty Engineering Hunedoara International Journal of Engineering (Archived copy), scheduled for publication in vol. 10, issue 1, January 2012. ISSN: 1584-2665.

[9] Flavio Lombardi, Roberto Di Pietro, "Secure Virtualization for Cloud Computing ", Journal of Network and Computer Application, vol. 34, issue 4, pp 1113-1122, July 2011, Academic Press td London, UK.

[10] R. Curtmola, O. Khan, R. Burns, and G. Ateniese, "MR-PDP: Multiple-Replica Provable Data Possession," Proc. of ICDCS '08, pp. 411–420, 2008.

[11] S. J. Schwarz and E. L. Miller, "Store, Forget, and Check: Using Algebraic Signatures to Check Remotely Administered Storage," Proc. of ICDCS '06, pp. 12–12, 2006.

[12] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song, "Provable Data Possession at Untrusted Stores," Proc. Of CCS '07, pp. 598–609, 2007.

[13] G. Ateniese, R. D. Pietro, L. V. Mancini, and G. Tsudik, "Scalable and Efficient Provable Data Possession," Proc. of Secure Comm. '08, pp. 1–10, 2008.

[14] R. Curtmola, O. Khan, R. Burns, and G. Ateniese, "MR-PDP: Multiple-Replica Provable Data Possession," Proc. of ICDCS '08, pp. 411–420, 2008.