ISSN 2320 – 2602
Volume 4 No.4, April 2015
International Journal of Advances in Computer Science and Technology
Available Online at http://www.warse.org/ijacst/static/pdf/file/ijacst05442015.pdf

A Secure Data Storage and Trustworthy Resource Sharing In Cloud Computing Environment

T. Esther Dyana[1], S. Maheswari[2]
[1]PG Student, Nandha Engineering College (Autonomous), Erode, estherdyana20@gmail.com
[2]Associate Professor, Nandha Engineering College (Autonomous), Erode, maheswarinec@gmail.com

ABSTRACT

Cloud Computing is an internet based computing where virtual shared servers provide software, infrastructure, platform, devices and many other resources and hosting to customers on a pay-as-you-use basis. Cloud computing customers do not own the physical infrastructure rather they rent the usage from a third-party provider. Data owners host their data on cloud servers and users can access the data from cloud servers. Due to the data outsourcing it introduces some new security challenges, which requires an auditing service to check the data integrity and confidentiality in the cloud. The steganography technique is used to encode the data that are hosted by the data owners and the watermarking technique is to decode the data that is to view who are all the authorized user access the owner's data. This would enhance the security of the cloud user and owner.

Key Words: Attacks, Security, Steganography, Watermarking.

1. INTRODUCTION

Cloud storage is an important service of cloud computing, which allows data owners to move data from their local computing systems to the cloud. More and more owners start to store the data in the cloud. However, this new paradigm of data hosting service also introduces new security challenges. Owners would worry that the data could be lost in the cloud. This is because data loss could happen in any infrastructure, no matter what high degree of reliable measures cloud service providers would take. Sometimes, cloud service providers might be dishonest. They could discard the data that have not been accessed or rarely accessed to save the storage space and claim that the data are still correctly stored in the cloud. Therefore, owners need to be convinced that the data are correctly stored in the cloud.

The steganography technique is used to hide secret messages in other messages, such that the secret's very existence is concealed. Generally the sender writes an innocuous message and then conceals a secret message in a text, image, audio or video and then encodes the data that are hosted by the data owners and the watermarking technique is to decode the data that is to view who are all the authorized user access the owner's data. This would enhance the security of the cloud user and owner.

The cloud owners should also check whether their data are accessed by the authorized users. The attackers could also try to gain some information from the data owners. For that instance, whenever the owner wants to store the information on the cloud they should encrypt the data and store. It will protect the data from both the dishonest cloud service provider and the attackers.

2. RELATED WORK

Steganography is an art that involves communication of secret data and an effort to conceal the existence of the embedded information. Some of them used in steganography are image, audio, video or TCP/IP header file. Combining secret image with the carrier image gives the hidden image [1], [2]. To protect customers' confidential data involved in the computations [3] keep both the sensitive input and output of the computation private.

The comparison of different techniques for steganography in images are described in [4], [5] explains each with its advantages and pitfalls and the Blowfish Algorithm [6] used for encrypting purpose. The vendor has to provide some assurance for security of data in the cloud computing [7]. Using homomorphic encryption and secure multiparty computation, cloud servers may perform regularly structured computation on encrypted data, without access to decryption keys [8]. The auditing protocol is to support the data dynamic operations, which is

efficient and provably secure in the random oracle model [9].

Using automatic blocker [11] for privacy preserving public auditing for data storage security in cloud computing and [12] solves the problems in privacy preserving data mining, collision avoidance in communications and distributed database access [13] propose taxonomy of security in cloud computing and [14] propose a different approach for securing data in the cloud using offensive decoy technology and monitor data access in the cloud and detect abnormal data access patterns. Privacy as a Service (PasS) provides a privacy feedback process which informs users of the different privacy operations applied on their data and makes them aware of any potential risks that may jeopardize the confidentiality of their sensitive information [15].

3. LOGGING MECHANISM:

- The Logger Structure
- Log Record Generation
- Dependability of Logs
- JARs Availability
- Log Correctness

3.1 The Logger Structure:

We leverage the programmable capability of JARs to conduct automated logging. A logger component is a nested Java JAR file which stores a user's data items and corresponding log files.

The Figure 1 describes the main responsibility of the outer JAR. It handles validation of entities which want to admittance the data stored in the JAR file. In our perspective, the data owners may not know the accurate CSPs that are going to handle the data. Hence, authentication is specified according to the servers' functionality (which we assume to be known through a lookup service), rather than the server's URL or identity. For example, a policy may state that Server X is allowed to download the data if it is a storage server. As discussed below, the outer JAR may also have the access control functionality to enforce the data owner's requirements, specified as Java policies, on the usage of the data.

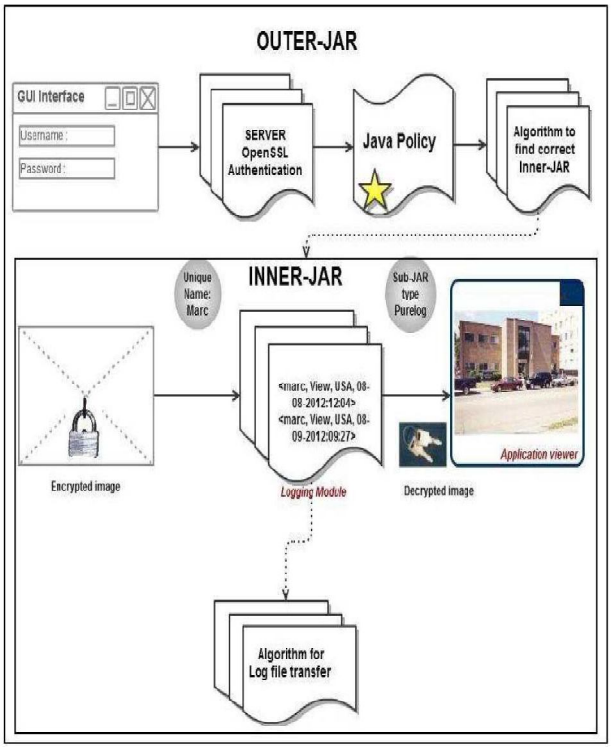

Figure 1 The Logger Structure

A Java policy specifies which permissions are available for a particular piece of code in a Java application environment. The permissions expressed in the Java policy are in terms of File System Permissions. Moreover, the outer JAR is also in charge of selecting the exact inner JAR according to the identity of the entity who requests the data.

Each inner JAR contains the encrypted data, class files to facilitate repossession of log files and display enclosed data in a suitable format, and a log file for each encrypted item. We support two options:

- **Pure Log.** To record every access to the data stored in cloud. The log files are used for pure auditing function that is very helpful for the data owner's for their data integrity and confidentiality.
- **Access Log.** It has two functions: logging actions and enforcing access control. In case an access request is denied, the JAR will record the time when the request is made. If the access request is granted, the JAR will additionally record the access information along with the duration for which the access is allowed.

No secret keys are ever stored in the system. The outer JAR may contain one or more inner JARs, in addition to a class file for authenticating the servers or the users, another class file verdict the correct inner JAR, a third class file which checks the JVM's validity using oblivious hashing. Further, a class file is used for managing the Graphical User Interface (GUI) for user authentication and the Java Policy.

3.2 Log Record Generation

Log records are generated by the logger component. Logging occurs at any access to the data in the JAR, and new log entries are appended sequentially, in order of creation LR=(r1; . . . ; rki). Each record ri is encrypted individually and appended to the log file. To ensure the correctness of the log records, we verify the access time, locations as well as actions. The most critical part is to log the actions on the users' data. In the current system, we support four types of actions, i.e., Act has one of the following four values: view, download, timed access, and Location-based access.

- **View:**

The entity (e.g., the cloud service provider) can only read the data but is not allowed to save a raw copy of it anywhere permanently. Recall that the data are encrypted and stored in the inner JAR. When there is a view-only access request, the inner JAR will decrypt the data on the fly and create a temporary decrypted file. The decrypted file will then be displayed to the entity using the Java application viewer in case the file is displayed to a human user.

- **Download:**

The entity is allowed to save a raw copy of the data and the entity will have no control over this copy either log records concerning access to the copy. If Pure Log is adopted, the user's data will be directly downloadable in a pure form using a link. When an entity clicks this download link, the JAR file associated with the data will decrypt the data and give it to the entity in raw form. In case of Access Logs, the entire JAR file will be given to the entity.

- **Timed access:**

This action is combined with the view-only access, and it indicates that the data are made available only for a certain period of time. The Pure log will just record the access starting time and its duration, while the Access Log will enforce that the access is allowed only within the specified period of time. The duration for which the access is allowed is calculated using the Network Time Protocol. The View access right not combined with the Download.

- **Location-based access:**

The Pure Log will record the location of the entities. The Access Log will verify the location for each of such access. The access is granted and the data are made available only to entities located at locations specified by the data owner.

3.3 Dependability of Logs

First, an attacker may try to evade the auditing mechanism by storing the JARs remotely, corrupting the JAR, or trying to prevent them from communicating with the user. Second, the attacker may try to compromise the JRE used to run the JAR files.

3.4 Jars Availability

To protect against attacks perpetrated on offline JARs, the CIA includes a log harmonizer which has two main responsibilities: to deal with copies of JARs and to recover corrupted logs.

Each log harmonizer is in charge of copies of logger components containing the same set of data items. The harmonizer is implemented as a JAR file. It does not contain the user's data items being audited, but consists of class files for both a server and a client processes to allow it to communicate with its logger components. The harmonizer stores error correction information sent from its logger components, as well as the user's IBE decryption key, to decrypt the log records and handle any duplicate records.

3.5 Log Correctness

For the logs to be correctly recorded, it is essential that the JRE of the system on which the logger components are running remain unmodified. To verify the integrity of the logger component, we rely on a two-step process:

1) We repair the JRE before the logger is launched and any kind of access is given, so as to provide guarantees of integrity of the JRE.

2) We insert hash codes, which calculate the hash values of the program traces of the modules being executed by the logger component. This helps us detect modifications of the JRE once the logger component has been launched, and are useful to

verify if the original code flow of execution is altered.

4. STEGANOGRAPHY AND WATERMARKING

Watermarking and steganography are processes in which the digital image is changed in a way that one may see the background image or the text without any kind of dishonesty in the image.

4.1 Steganography

Steganography is changing the image in a way that only the sender and the intended recipient is able to detect the message sent through it. It is invisible, and thus the detection is not easy. It is a better way of sending secret messages than encoded messages or cryptography as it does not attract attention to itself.

There are many ways in which steganography is done. The messages appear as articles, images, lists, or sometimes invisible ink is used to write between the lines. Steganography is achieved by concealing the information in computer files. Sometimes steganographic codes are inside the transport layer like an image file, document file, media files, etc. Due to the large size of the media files, they are considered ideal for steganography.

4.1.1 Features of Steganographic Techniques

Steganographic techniques have various features which are dealt in [2] characterizes their strengths and weaknesses. Features include:

- **Embedding capacity:** The amount of data that can be inserted into the cover-media without deteriorating its integrity.
- **Perceptual transparency:** To avoid suspicion the embedding should occur without significant degradation or loss of perceptual quality of the cover media.
- **Robustness:** The ability of embedded data to remain intact if the stego-image undergoes various transformations such as scaling, rotation, cropping or compression.
- **Tamper resistance:** The difficulty to alter or forge a message once it is embedded in a cover-media, such as replacing a copyright mark with the one claiming legal ownership.
- **Computational complexity:** Computational complexity of steganography technique employed for encoding and decoding is another consideration and should be given importance.

4.1.2 Classifications of Steganographic Techniques

Classifications of steganographic techniques based on the types of cover files and data hiding are shown in Figure 2and Figure 3. Almost all digital file formats can be used for steganography, however only those with a high degree of redundant bits are preferred. The larger size of audio and video files makes them less popular as compared to images. In Spatial domain, cover-image is first decomposed into bits planes and then least significant bit (LSB) [10] of the bits planes are replaced with the secret data bits.

Advantages are high embedding capacity, ease of implementation and imperceptibility of hidden data, higher level of robustness against simple statistical analysis. Unfortunately, it lacks high embedding. In compression domain, secret data is embedded into compression codes of the cover-image which is then sent to the receiver. The major drawback is its vulnerability to various simple statistical analysis methods. Frequency domain embedding techniques, which first transforms the cover-image into its frequency domain, secret data is then embedded in frequency coefficients.

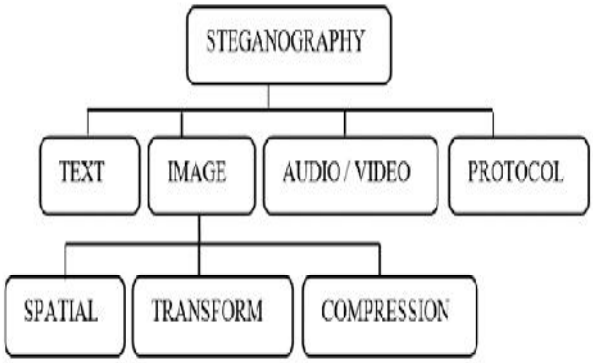

Figure 2 Classifications Of Steganographic Techniques.

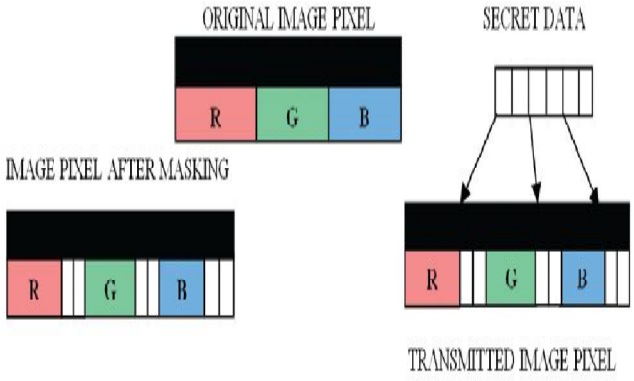

Figure 3 Basic Spatial Domain Data Hiding

An improved algorithm has been adopted in literature [6], the secret image is first encrypted by using BLOWFISH algorithm which has very good performance and is a most powerful technique compared to other Algorithms.

4.1.3 Applications

- Steganography is used in modern printers.
- It has been used allegedly by terrorists.
- It is allegedly used by intelligence services.

4.2 Watermarking

Watermarking is used to verify the identity and authenticity of the owner of a digital image. It is a process in which the information which verifies the owner is embedded into the digital image or signal. These signals could be either videos or pictures or audios. For example, famous artists watermark their pictures and images. If somebody tries to copy the image, the watermark is copied along with the image. Watermarking is of two types; visible watermarking and invisible watermarking.

4.2.1 Visible Watermarking

As the name suggests, visible watermarking refers to the information visible on the image or video or picture. Visible watermarks are typically logos or text. For example, in a TV broadcast, the logo of the broadcaster is visible at the right side of the screen.

4.2.2 Invisible Watermarking

Invisible watermarking refers to adding information in a video or picture or audio as digital data. It is not visible or perceivable, but it can be detected by different means. It may also be a form or type of steganography and is used for widespread use. It can be retrieved easily.

Some of the attacks in watermarking are robustness attack, presentation, interpretation and implementation attacks.

4.2.3 Applications:

- It is used for copyright protection.
- It is used for source tracing.
- Annotation of photographs. Adding identity information in medical images, so that there is no need of external procedures to insure the right binding between identities and pictures.
- Captioning of pictures, useful for professional photographers and news agencies.
- Feature tagging in digital images, so that parts of them can be highlighted and described briefly.
- Indexing of stream data, as audio files, without the need of an additional communication channel.

4.3 ADVANTAGES

- One of the main innovative features of the CIA framework lies in its ability of maintaining lightweight and powerful accountability that combines aspects of access control, usage control and authentication.
- Providing defenses against man in middle attack, dictionary attack, Disassembling Attack, Compromised JVM Attack.
- The ability of invisibility in steganography.
- The ability to embed information in any type of file format.
- Steganography have relatively high data rate.
- It's Suitable for limited and large number of storages.

5. EXPERIMENTAL RESULTS:

Our approach permits the information owner to not solely audit his content however additionally enforces sturdy back-end protection if required. Moreover, one amongst the most options of our work is that it permits owner to audit even those copies of its data that were created while not his information. Within the future, we have a tendency to decide to refine our approach to verify the integrity of the JRE and also the authentication of JARs. This analysis is geared toward providing software system tamper resistance to Java applications. Within the future, we have a tendency to decide to style a comprehensive and a lot of generic object-oriented approach to facilitate autonomous protection of traveling content. And to support a range of security policies, like assortment policies for text files, usage management for executable, and generic responsibility and beginning controls.

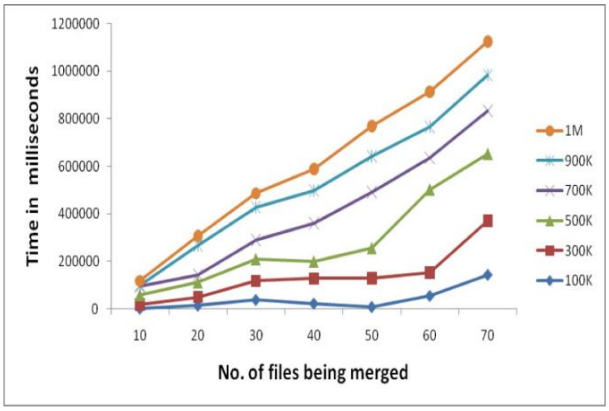

Figure 4 Average time to merge files

6. CONCLUSION AND FUTURE WORK

In this study allows the data owner to not only audit his content but also enforce strong back-end protection if needed. Moreover, one of the main features of our work is that it enables the data owner to audit even those copies of its data that were made without his knowledge. In the future, we plan to refine our approach to verify the integrity of the JRE and the authentication of JARs. This research is aimed at providing software tamper resistance to Java applications. In the long term, we plan to design a comprehensive and more generic object-oriented approach to facilitate autonomous protection of traveling content. We would like to support a variety of security policies, like indexing policies for text files, usage control for executables, and generic accountability and provenance controls.

REFERENCES

1. Arvind Kumar, Km. Pooja. (2010), 'Steganography- A Data Hiding Technique', International Journal of Computer Applications', Vol.9, No.7,ISSN: 0975 – 8887.
2. Babloo Saha and Shuchi Sharma. (2012), 'Steganographic Techniques of Data Hiding using Digital Images', Defence Science Journal, Vol.62, No.1, pp.11-18.
3. Cong Wang, Kui Ren, Jia Wang, Qian Wang. (2013), 'Harnessing the Cloud for Securely Outsourcing Large –Scale Systems of Linear Equation', IEEE Transactions On Parallel And Distributed Systems, Vol.24, No.6.
4. Falesh M. Shelke, Ashwini A. Dongre, Pravin D. Soni. (2014), 'Comparison of different techniques for Steganography in images', International Journal of Application or Innovation in Engineering & Management, Vol.3, Issue 2, ISSN: 2319-4847.
5. Hardikkumar V. Desai. (2012), 'Steganography, Cryptography, Watermarking: A Comparative Study', Journal of Global Research in Computer Science, Vol.3, No.12, ISSN: 2229-371X.
6. Hemlata Sharma, Mithlesh Arya and Dinesh Goyal. (2013), 'Secure Image Hiding Algorithm using Cryptography and Steganography' IOSR Journal of Computer Engineering, Vol.13, Issue 5, ISSN: 2278-8727, pp. 1-6.
7. G. Jai Arul Jose, C. Sajeev. (2011), 'Implementation of Data Security in Cloud Computing', International Journal Of P2p Network Trends And Technology, ISSN: 2249-2615.
8. John C. Mitchell, Rahul Sharma, Deian Stefan, and Joe Zimmerman.(2012), 'Information-flow control for Programming on Encrypted Data', IEEE Computer Society, IEEE 25th Computer Security Foundations Symposium.
9. Kan Yang, Xiaohua Jia.(2013), 'An Efficient and Secure Dynamic Auditing Protocol for data Storage in Cloud Computing', IEEE Transactions on Parallel and Distributed Systems, Vol.24, No.9.
10. Kanika Anand, Er. Rekha Sharma. (2014), 'Comparison of LSB and MSB Based Image Steganography', International Journal of Advanced Research in Computer Science and Software Engineering, Vol.4, Issue 8, ISSN: 2277-128X.
11. K.Kiran Kumar, K.Padmaja, P.Radha Krishna.(2012), 'Automatic protocol Blocker for Privacy-preserving public Auditing in Cloud computing', IEEE Transactions On Cloud Computing, IJCST Vol. 3, Issue 1, Spl. 5, ISSN : 2229-4333.
12. Larry A. Dunning, and Ray Kresman.(2013), 'Privacy Preserving Data Sharing with Anonymous ID Assignment', IEEE Transactions On Information Forensics And Security, Vol. 8, No. 2.
13. Nelson Gonzalez, Charles Miers, Tereza Carvalho, Mats Naslund and Makan Pourzandi.(2012), 'A Quantitative Analysis of Current Security Concerns for Cloud Computing', Gonzalez Et Al. Journal Of Cloud Computing: Advances, Systems And Applications, **1**:11.

14. Salvatore J. Stolfo, Malek Ben Salem, Angelos D. Keromytis.(2012), 'Fog Computing: Mitigating Insider Data Theft Attacks in the Cloud', IEEE Symposium On Security And Privacy Works.
15. Wassim Itani Ayman Kayssi Ali Chehab.(2009), 'Privacy as a Service: Privacy-Aware Data Storage and processing in Cloud Computing Architectures' Eighth IEEE International Conference On Dependable Autonomic And Secure Computing, pp. 711-716.