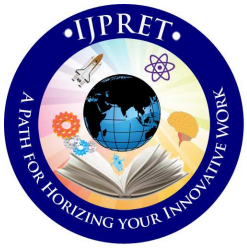
Research Article *ISSN: 2319-507X*
Sandeep Tengale, IJPRET, 2013; Volume 1(8): 379-391 *IJPRET*



INTERNATIONAL JOURNAL OF PURE AND APPLIED RESEARCH IN ENGINEERING AND TECHNOLOGY

A PATH FOR HORIZING YOUR INNOVATIVE WORK

AN APPROACH TO ENHANCE THE PRIVACY IN CLOUD COMPUTING USING ACCOUNTABILITY AND STENOGRAPHY

SANDEEP TENGALE

Department of Information Science and Engineering, BLDE College of Engineering and Technology, Bijapur

Accepted Date:

27/02/2013

Publish Date:

01/04/2013

Keywords

Cloud,

Cloud computing,

Eucalyptus,

Maven.

Corresponding Author

Mr. Sandeep Tengale

Abstract

Cloud computing is a technology which is considered as the next stage in Internet evolution. The “cloud” in cloud computing can be defined as the set of hardware, networks, storage, services, and interfaces that combine to deliver aspects of computing as a service. Cloud services include the delivery of software, infrastructure, and storage over the Internet based on user requirements. There are two aspects which are in need to concern when we are using the cloud service, they are Security and Privacy. So cloud service provider should assure the security and privacy of the data, limitations in security leads to a huge loss to cloud user as well as service provider in terms of reputation. In this paper, we are proposing a method which can be used to control the privacy of the data. We are proposing the two methods, where first one uses the stenography technique to hide the text data in the image and other uses the accounting technique to get data user list. As the user requirement increases the cloud service provider hires the third party server to store and process the user data, which will leads to the violation of the privacy. We have tested both the method on the Eucalyptus test bed using maven and results are presented.

1. Introduction

Cloud computing is a new trends in Internet technology which has revolutionized the Internet in terms of its use and capability. Cloud computing is like a service on demand which provides the service to the end user based on their requirements. It is called as anytime anywhere service which allow the users to access and use their data anytime and anywhere. Cloud computing is about storing the data and required tools on the network resources such as servers and storage devices (cloud) and use it anytime and at anyplace. The different format of data can be stored on the cloud, the different range of tools can be used on the cloud, from word processing tool to complex image processing tool. The end users are not need to have all the tools with him all the time, he just need to have the browser to use the tools stored at the server.

In this paper, we are addressing the issue of the security and the privacy of the data that we are storing on cloud server. In this paper, we are particularly highlighting the privacy aspect of the data. As the cloud users are storing sensitive information and proprietary tools on the cloud, there should be some technique to handle the security issues of our data. The usage of cloud will make our data more transparent to cloud service provider. There are some possibilities that the service provider sell the user data and tools to third party, so the proposed method will address such an issue.

Moreover this, as the user requirement increases gradually the cloud service providers starts running out of the resources or the available server are not capable to handle the requested service type, in such cases they will hire some third party servers which is again a compromising factor to the privacy of the user data. Disturbance of the security will affect the cloud user as well as the service providers. So its better to adopt the innovative approaches of data security to the cloud computing to enhance the privacy of the data. There are different ways to hide the data such as encryption, cryptography etc. In the proposed method, we are using the Stenographic method to hide the user data.

Stenography is a data hiding technique, which will hide one format of data into another format i.e. hiding of text data into image, video or audio without distorting the container. In the proposed method, we are hiding the text data into the image of type JPEG. So that even if the user data is available to the cloud service provider in image format, but he will not able get the sensitive data out of it.

Since the data is stored on the servers which are neither owned nor controlled by some service provider, there is always a fear of losing the control over their data. Hence we are introducing a highly decentralized approach which aims to maintain accountability log in the server in an executable JAR. The data owner can access this JAR by authenticating himself to the service which will list all the users who access the user data making the data usage transparent and traceable. This not only adds access control but also the usage control. Both method proposed here is tested on Eucalyptus test bed and on Heroku cloud using Maven. Eucalyptus is open source software for cloud virtualization which allow user to install required operating system on the cloud and Heroku is an online service which lets the users to test their application of the cloud. The rest of the paper is organized as follows Section 2 is related work on security of cloud computing, section 3 is proposed method, section 4 deals with the implementation details of proposed methods and Section 5 is the conclusion of the paper.

2. Related Work

A. **Accountability** is one of the factors that can add more privacy constraint on the user data. This can be achieve by using an JAR(Java Archive) file which stores all the information about the users accessed the particular data from the cloud. The method also makes the JAR inaccessible to unauthorized user. The logging method is implemented to this JAR file only the authorized user can access or edit this file by providing required credentials. The JAR file is just like a log file which stores the information such as IP address of the access user, time of access and the file is a tamper and damage proof. The file is made decentralized to make communication load

less as possible. So as to make the JAR file accessible to only the authentic user, the Identity based encryption is used which uses the public and private key pair. Whenever any user access the data from the server, the log file in the JAR gets updated and it is encrypted with the public key of the user. User can decrypt this log file using his private key whenever it is needed.

B. Stenography

This method uses the novel approach of Stenography, which uses a secret key to get access to the image and later it will perform the stenography. The approach includes manipulating the bit values of the container image without distorting much of the image content. It uses the shift operations such as left and right shift depending upon the size of the image. This container image can be stored or can be sent to different location. Only the person who knows the stenography algorithm used to hide can only able to extract the text from the given image. This technique has used the .GIF and .JPEG image as a container to hide the text. The algorithm is tested with the PSNR (Peaksignal-to-noise ratio), which indicated the quality of the container after application of the Stenography. The PSNR value is high which indicates the good quality of the Stenography.

C. Access control in cloud

There are some methods which deal with how we can control the access to our own data on the third party server. There are many issues which need to be addressed when we are using the data that is stored on the external sever such as security, availability and the data control. There are security issues before the cloud computing gets into market above that we are making data more transparent with the use of cloud technology. The available service like Gmail and Amazon leverages the advertising service to third party services which seeks the user data to provide more correct ads making the data more transparent, to cope with such as access, a novel approach is implemented. Here the trusted monitor in installed on the server which audits the data. Trusted Computing also allows secure bootstrapping of this monitor to run beside (and securely isolated

from) the operating system and applications. The monitor can enforce access control policies and perform monitoring/auditing tasks.

C. Virtualization

There is a method to improve the security in the cloud computing by using the virtualization. Cloud computing already leverages the virtualization for the load balancing with physical machines. The concept of virtualization in turn helps in the easy implementation of the security procedures. Key kernel is a component that is actively managing the key in the system which is prune to get attacked. In order to monitor the attack, the method is monitoring the middleware components so as able to detect any possible modification to kernel data and code, thus guaranteeing that kernel and middleware integrity have not been compromised. The author has included in the ACPS (Advanced Cloud Protection System) in a virtual machine which continuously monitors the cloud environment from the external attack on kernel and the middleware and the whole system is embedded on the Eucalyptus test bed and OpenECP. Suspicious guestactivities canbe noticed bythe Interceptor and recordedbythe Warning Recorder into the Warning Pool, wherethepotentialthreatwillbe evaluated bythe Evaluator component and reported the administrator.

3. Proposed Method

Here the entire method is divided into two modules, one as a Stenography method and another is the accountability method. As the name indicated Stenography is data hiding technique and accountability is the data tracking technique which keeps the account of user who access the data from the cloud server. First we look at the Stenography method used; this method is tested on the Eucalyptus test bed and Heroku cloud. Stenography is a java module which is developed by shifting the image pixel bits.

Stenography used in this case hides the text data in the image; here we are using the JPEG format image to manipulate. Every image is made up of set of pixel and each pixel is composed of set of bits which represents the resolution of the image.

Every pixel of the image in JPEG image is made up of 8 bits which represents the RBG component each. Depending on the text we want to hide, we manipulate these bits by shifting them in the required direction. The care should be taken that the image should not get distracted with this manipulation.

In the proposed technique, we are working on the individual pixels. Every pixel is made up of 8 bits, where least significant bit (LSB) represents lower details of the image where as the most significant bit(MSB) represent the higher details of the image which usually represent the edges and lines in the image. So here we just manipulate the LSB so that distortion of image will not take place.

First we collect the text which has to be hidden in the image. Then we calculate the ASCII value of each letter and store in the form of array. Next calculate the binary value of the given ASCII value and store separately in other variable. The bit shift operations are applied on the pixels of the image to store the text in it. Here are the steps to hide the text into carrier image. The given carrier image is a set of pixel of particular length and width. First store the image into the buffer and bring it to the user space which gives all flexibility to work directly on the pixels. Take the first letter from the text which we want to hide, get its ASCII value and convert into binary format. Shift the bits in every iteration by some number and store it in the last bit of the pixel.

It's required to have two iteration values, one to keep track of the number of iterations we are making and another to maintain the offset of the image in the user space. Only the last bit in every pixel is edited by ANDing the pixel value with the 0XFE.

Now once the text is added to the container image the data is hidden, we can use this image to store on the cloud server or it can be sent to third party. Because it is difficult to predict the existence of text in the image, even if it is predicted it's difficult to extract. The next operation is to extract the text form the container image whenever it is required for use.

The extraction of the text form the image is reverse process of hiding of the text. Since

the length of the text data is not known to the system, so first 4 bytes of the image is dedicated to store the length of the text data that has been hidden in the image. Depending on the value of length, first the array of byte of given length is created. In every iteration the pixel of given image which is loaded into the user space will be taken and its LSB will be taken out by ANDing it with 0001. The offset value will keep track of the pixel which is under the process. The LSB value taken will be stored into the array of bytes which we have created initially. Now byte value is again converted back to the character as an extracted text which is hidden in the image. Entire method is tested on Heroku using maven as interpreter. So here we are controlling the privacy of the user by using the Stenography.

Another approach to control the privacy is by using the technique of Accounting. The accounting will keep track of the user who has accessed the data stored on the cloud storage. It uses the public key cryptography along with the data encryption technique to carry the task. This module consists of the two components 1. Logger 2. Log harmonizer

Logger is a component which is closely associated with the user data. Whenever any user tries to access any particular data available on the cloud storage, the logger generates the log data and it will be sent to the log harmonizer. The data in the cloud has some set of permissions such as read, modify, download etc. depending on the operation that is performed on the data the log file is generated and stored o the log harmonizer. Now the log harmonizer acts as the central component which allows the user to look into the log files generated. There are two methods by which we can look into the log i.e Push and Pull method. Push method is one where the cloud service provider themselves can send the log files to the user to keep the user updated about their data usage. These are beneficial for those who want regular updates on their data usage especially organization owners or it is useful in those cases where the user data is in heavy use. Pull method is like on demand service, where the log information will be provided to the users on request.

Whole procedure is implemented using the JAR concept of JAVA. JAR files are going to act like the Logger and the Log harmonizer on the cloud. Every cluster of data available on the cloud server has a closely associated JAR file along with it. So this process is highly distributed and light weighted too. It is possible that either cloud service provider or external party can access or manipulate the JAR file. So this method encrypts the JAR file using the public key of the user. The steps are pictorially showed in figure 1

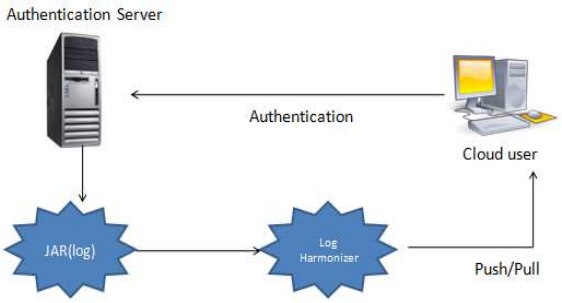


Fig 1: Block for Accountability logger

The procedure starts with the generation of key pairs by the user and authenticates himself to the cloud service provider. This process of decrypting the JAR will take place automatically as the user logs into the service. Now user can apply Pull technique to download this file on to his system to check the information about all the users who has accessed the data. So whenever cloud service provider or the third party tries to access the user data from the cloud server the log file will gets updated. The log file collects the set of information such as ID, ACT, TIME, Loc, Sig where ID represents some identity of the user such as IP address, ACT represents the action that is performed on the data like read, download or modify, TIME represents the time of access of user data, Loc represent the location of the user from where the access is been done, this is indirectly dependent on the IP address of the entity used it and sig gives the signature of log so that it was retrieved in case of corruption. These all information is stored in the log file in the form of the vector like

r_i=<ID,ACT,TIME,Loc,SIG>

As soon as the log record is generated by the logger associated with the data, the record get updated into the JAR filed called Log harmonizer. We can improvise the security of the JAR file by embedding it with an external JAR.

This technique will help to implement different types of security attacks like:-

Copying attack: If any other user such as cloud service provider themselves or attacker tries to access or download our data stored on cloud server, then log file will be updated with the access information.

Breaking JAR attack:- Sometimes attacker may try to break the JAR file which consist of the access log information so to manipulate it. But it can be controlled since the JAR is encrypted with the public key of the data owner.

Man in middle attack:- This attack is difficult to take place, because it is only possible at two points one before the access or after the access. Since while access is taking place, it supported by SSL encryption so attacking in middle is difficult. Accessing it before or after the use of JAR is not possible as the JAR is fixed with timestamp, after the timestamp the JAR will get encrypted again with the public key of the user.

4. Implementation

We have implemented this method on the cloud test bed called Eucalyptus and third party service called Heroku and SalesForce. First we tested it on Eucalyptus installed on Ubuntu 11.10 server. On Eucalyptus, an Eucalyptus certified operating system called Euca2oolis installed which supports the virtual machine to run the java application. Here we are using Maven as the virtual machine which supports to run the Java web and simple AWT application.

Stenography and Availability parts are implemented separately and mounted over the cloud and tested. First we have a Stenography module developed AWT which accepts the text data from the user which we want to hide and the image which acts like a container to hide the sensitive data. Bits of every pixel will be shifted based on the text data that we are going to hide. First the text data is converted to its ASCII value and it converted to binary value. The last bit of the binary value will be loaded to the last bit (LSB)of image pixel. Here is the algorithm

private byte[] bit_conversion(int i)

{byte byte3 = (byte)((i & 0xFF000000) >>>24); byte byte2 = (byte)((i & 0x00FF0000) >>> 16); byte byte1 = (byte)((i

& 0x0000FF00) >>> 8); byte byte0 = (byte)((i & 0x000000FF)); return(new byte[]{byte3,byte2,byte1,byte0});}

This new byte is will be ANDed with the image pixel by updating particular pixel in the image which is updated by an offset variable.

int b = (add >>> bit) & 1;

image [offset] = (byte)((image[offset] & 0xFE) | b);

We used 0XFE because we want to keep all 15 bits except the last bit.

Next part is taking data out of the container image, it is exactly the reverse of the hiding process. Based on the offset value which we have stored previously, keep tracing the bits in the pixel to gather all the bits to store it in an temporary variable, which will be the text data that we have hided in the container image.

Result [b] = (byte)((result[b] << 1) | (image[offset] & 1));

Result is the variable where the given text data is stored which is retrieved and displayed. Figure 2 shown the design of the design of the Stenography module.

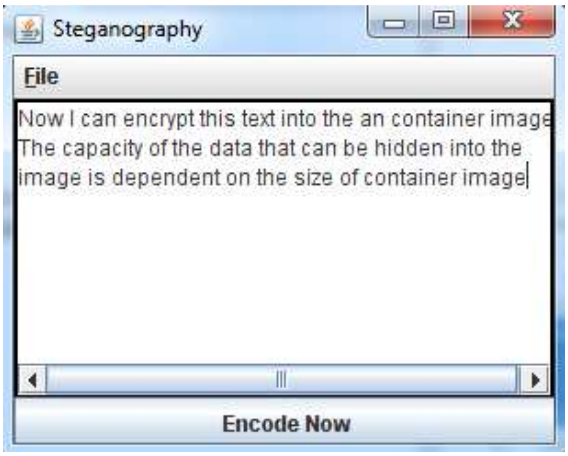


Fig 2: Text to be hiding

Another part is the Accountability, where we developed a Java Archive file and place it on the cloud server. We are using a third party service from SalesForce which carry the encrypted login part, where we are using RSA algorithm to encrypt so as to make the secure login. The data is present on the server for the image files each having the set of permission. The JAR is also present in the encrypted form which gets decrypted when user login. Means after login the JAR is open and accessible. Whatever data that is used by any entity will be getting logged into the JAR file continuously, even the data owner access his own data, that is also getting updated

into the log file. This file again gets encrypted with the public key as soon as user logs off. The log file available can be pushed to the user periodically so that they can get to know about their data access. If the user wants the data at any point of time, then he can download the log file with the pull method. The design of the Accountability module is shown in following figure 3, where we can see the list of files currently residing in the server. The list also includes the owner of the file and date at which the file is uploaded. All the files present in the server are of type .PNG type. Since the Stenography modules produce the output in PNG format only. We can also look at the option for the PULL method to work which downloads the log file to the client system

My files on cloud:

hello.png sandeep 11-11-2012
www.png sandeep 11-11-2012
mytext.png sandeep 11-11-2012
gmail.png sandeep 12-11-2012

Get Log File

Fig 3: List of files on the server

The client system can access the files in the server using from the browser via an application which is capable of opening the hidden text from the container image. Once the file is available on the server, the decrypt application should download the .PNG image from the server and extract the text out of it.

Fig 4: Container Image

This application will extract the text from image using decode option. This decode option will get data from pixel with every iteration of the offset and fetch the last bit from every pixel. The fetched pixel is arranged in a byte variable which later going to form an ACII value. Those ASCII values are interpreted and display on the

client machine as the user data. The decoded text is shown in the figure 5.

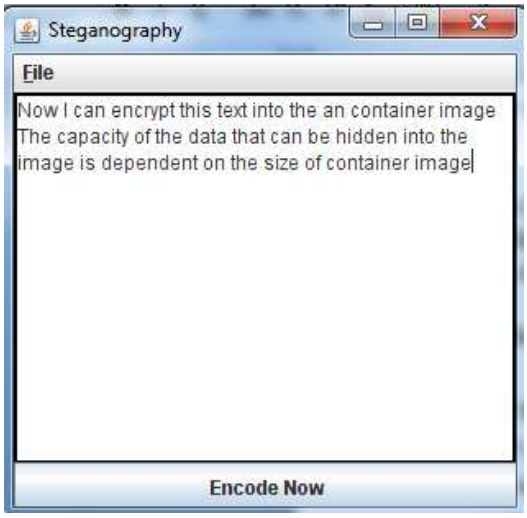


Fig 5: Decoding the text from Container image

The only drawback of this system is its scalability issue. We need to store large sized images onto the server which consumes much space. Since scalability itself is a major issue in all client-server related service. Because storage requirements increases with time. Storing a new image for small amount of text data is an expensive task. The remedy to this type of stenography is the store the text data in the media like video, which has high capacity to store the text data. But stenography is best method to control the privacy in the data, in extension to that we have added accountability factor which keeps track of the illegal access of our data in the cloud platform.

5. Conclusion

Cloud is used by larger firms and an organization to store the sensitive data, storing the sensitive data on the unknown cloud server (which again hires the third party server to meet the demands) is threat to privacy of the data. So we can improve the privacy protection of the data using the oldest text hiding technique called stenography because of which the data is secure even if the data is available, the attacker will not able to make much sense form the data. Again to enhance the privacy of the service, the accountability service is also been added which will keep track of the entire user data on the cloud server, which will present the data owner the access log who have accessed the data. Using both the techniques we can keep track of customer data, enhancing the privacy of the service to higher level.

References

1. Smitha Sundareswaran, Anna C. Squicciarini, "Ensuring Distributed Account ability for Data Sharing in the Cloud", IEEE Transactions on Dependable and Secure Computing, Vol. 9, No. 4, July/August 2012.

2. Chao-Tung Yang, Wen-Chung Shih, Guan-Han Chen, Shih-Chi Yu, "Implementation of a Cloud Computing Environment for Hiding Huge Amounts of Data", International Symposium on Parallel and Distributed Processing with Applications.

3. Chittaranjan Hota, Sunil Sanka, "Capability-based Cryptographic Data Access Control in Cloud Computing", Int. J. Advanced Networking and Applications Volume: 03; Issue: 03; Pages: 1152-1161 (2011).

4. Richard Chow, Philippe Golle, Markus Jakobsson, "Controlling Data in the Cloud:

5. Outsourcing Computation without Outsourcing Control", *CCSW'09*, November 13, 2009, Chicago, Illinois, USA.

6. Rosziati Ibrahim and Teoh Suk Kuan, "Steganography Algorithm to Hide Secret Message insidean Image", Computer Technology and Application Jouranal, (2011) pp-102-108.

7. Flavio Lombardi, Roberto DiPietro, "Secure virtualization for cloud computing" ,Journal of Network and Computer Applications, November 2009.

8. Rolf Blom, An optimal class of symmetric key generation systems, *Proc.* EUROCRYPT 84 workshop on Advances in cryptology: theory and application of cryptographic techniques, Springer Verlag, NY, USA, 1985, pp. 335-338.

9. G. Miklau, and D. Suciu, Controlling access to published data using cryptography, *Proc.* 29th VLDB, Germany, Sept 2003, pp. 898-909.