Thwarting Audio Steganography Attacks in Cloud Storage Systems

Bo Liu, Erci Xu, Jin Wang, Ziling Wei, Liyang Xu, Baokang Zhao*, Jinshu Su
Department of Computer Science
National University of Defense Technology
Changsha, Hunan, CHINA
{bkzhao,sjs}@nudt.edu.cn

***Abstract*— Nowadays, enterprises and individuals are increasing tending to store their data in the cloud storage systems, yet, these sensitive data will face serious security threats. Currently, cloud storage service providers mainly adopt encryption and authentication to protect sensitive data, and a lot of approaches have been proposed to ensure data security in cloud storage systems.**

Recently, audio steganography has been regarded as serious attacking measures to threaten cloud storage systems. Nevertheless, little research has been focused on thwarting the Audio steganography Attacks in Cloud Storage Systems.

In this paper, we analyze the Audiosteganography Attacks in Cloud Storage Systems, and then, we propose and develop StegAD, a novel scheme for defending Audiosteganography Attacks. StegAD includes two algorithms, i.e., the enhanced-RS algorithm and the SADI algorithm. The enhanced-RS algorithm is adopted to detect the audio steganographied files, and after that, SADI is applied to infer and compensate the possible hiding positions. To evaluate the performance of StegAD, we perform extensive evaluations on a real platform in terms of detecting, audio quality and interfering intensity. Experimental results show that, our proposed StegAD scheme is very efficient in thwarting the Audio steganography Attacks in Cloud Storage Systems.

Keywords- cloud storage; steganography; security

I. INTRODUCTION

Cloud storage, the byproduct of cloud computing, is the online storage which generally managed by third parties. Cloud storage provides us with the immense space for storing. Unlike traditional storage, the cloud is provided as a service [1] [2], therefore users or enterprises no longer need to worry about the hardware management and maintenance. Yet, Moving data to a third-party hosted storage may easily bring serious security concerns over sensitive data. Therefore, many service providers offer defensive mechanism [3][4].

Audio steganographyAudiois very efficient to enable users hiding their secret data within regular media files. As a result, steganography users can transmit the secret information just by sending the steganographied media files which look (sound) common. Therefore, recently, researchers find that audio steganography attacks will be very serious in certain threats on data security in cloud storage[7], which can deceive the current defensive mechanism, and traditional countermeasures (i.e., steganalysis) to steganography attacksis unable to deployed the cloud storage environment due to performance limits[7]. It is very urgent and challenging to develop practical solutions to thwart the audio steganography attacks for cloud storage systems.

In this paper, we perform a careful analysis on the Audio steganography Attacks in Cloud Storage Systems. And thereafter, we design and develop StegAD (Stegnaography Active Defence), a cloud-suitable defensive scheme to tackle with threats of data leakage by using audio steganography on cloud storage. In StegAD, we firstly transplant the famous RS image grayscale steganalysis algorithm to the scan the possible hiding place of audio files under cloud storage environment. Then after acquiring the suspicious files, we use the SADI (Steganography Audio Dynamical Interference) technique to interfere all the possible places in those suspicious files. The interference leads to two results. For those steganographied files the interfering process may consequence in damage to the information hid in (i.e., steganograms) and make it impossible to restore the steganograms. On the other hand, the innocent files, though suffer the interference, won't take devastating damage to audio quality since the interference in possible hiding place is likely only arousing minor distortion. Moreover, after initial interference, the SADI will further eliminate the steganograms and manage to reduce the quality loss in interference by comparing the previous audio value with the one after interference and minimize the difference by changing neighboring audio values. By doing so, the innocent files may suffer from less audio quality loss and it is more difficult for steganogram to restore since the more hiding place has been interfered in the latter compensation.

To evaluate the performance of StegAD, we build a prototype cloud storage system and conduct extensive experiments in terms of detecting steganographied files, interference effects on various steganography techniques and audio quality loss. Experimental results show the efficiency of our proposed StegAD scheme.

The remaining part of this paper proceeds as follows. Section II presents the related work. The preliminaries ofAudio

978-1-4577-1637-9/11/$26.00 ©2011 IEEE

audio steganography techniques are outlined in Section III. The proposed StegAD scheme is described in Section V. We show the experimental results and analysis in Section VI. Finally, Section VII concludes the paper.

II. RELATED WORK

Moving data to the cloud is certainly convenient for the enterprises no longer need to worry about storage hardware and its maintenance. However, there is no environment which is completely free of security threats. [3]Currently, main threats of cloud storage include data leakage, user identification, data snooping and et al. These threats are not just theoretical hypothesis. [5]Prevalent software called S3 ripper, which aims at Amazon S3 bucket cloud storage service, can easily list the date, sizes and even the entire content of the Amazon cloud storage file when given the file name. Other than hackers and malwares, the malicious insiders and data snooping may also lead to data loss or leakage eventually.

Therefore, in [4] [6], various methods such as limited access and encryption are proposed to prevent the data leakage. In [4], Windows Azure cloud service uses gatekeeper pattern to limit brokered access to storage, which limits the attack surface on sensitive data via using intermediate, privileged roles. These roles are deployed on separate VMs, so that breaking into one web role does not give special access onto the other.

In [6], Natuni design and develop their encrypt tool Nasuni Filer on OpenPGP for cloud storage service, a framework which combines various algorithm in a system. In Nasuni Filer, the software uses AES-256 cipher and offers a RSA OpenPGP for each customer. With highly intense encryption, malicious hackers are not likely to get access to those files on cloud service.

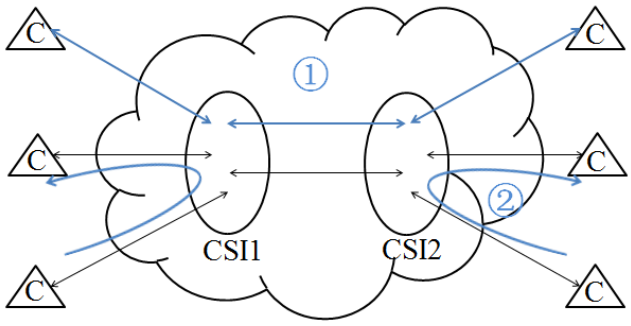

Figure 1. Hidden communication scenarios for cloud computing environment (CSI – Cloud Service Instance,C – Cloud Clients/Customers，CSI1 –Private Cloud, CSI2 –Public Cloud)

Yet, in[7],a novel threat, steganography on cloud storage, may be provided as a huge threat for data exfiltration. Steganography is a mechanism that uses common media files as cover to hide secret information and to transfer them. Therefore steganography technique can be used to leak data in cloud storage service where all behaviors are seemly legal. To be specific, malicious insiders replace the less significant parts in the media files with secret information. With a little distortion in less significant places, the steganographied media file does not look (sound) differently to original ones. Therefore the cover files are expected to be transmitted freely in cloud storage environment. With no cracking on encrypted files and hacking on access, the steganography technique deceives the current defensive mechanism of cloud security service subtly and manages to leak the data

As it is shown in the figure-1, the process ① indicates the steganography information transmitted through different cloud storage system, and the process ② indicates the steganography transmitted within one cloud storage.

As a countermeasure to steganography, steganalysis mainly relies on two techniques, monitoring users' behavior of using suspicious applications and using algorithms to find out files of using steganography. For example, famous steganalysis tool StegAlyzerRTS[8] is able to detect fingerprints of over 960 steganography applications and signatures of over 55 steganography applications. Further, algorithms such as RS algorithm use statistical measures to scan the files least significant bits for data signature of steganography [9]. However, such steganalysis may not work well under the cloud storage environment. Firstly, with regard to huge amounts of users, cloud service cannot afford the payload of monitoring everyone's behavior of using steganography. Plus, despite the time consumption, latest steganography techniques are equipped with techniques including secret sharing [10], pseudo-random embedding [11] and et al., steganalysis may receive less ideal results and more innocent files may be involved. Therefore, simply relying on stegnalysis is not safe.

III. PRELIMINARIES

A. *Introduction to steganography*

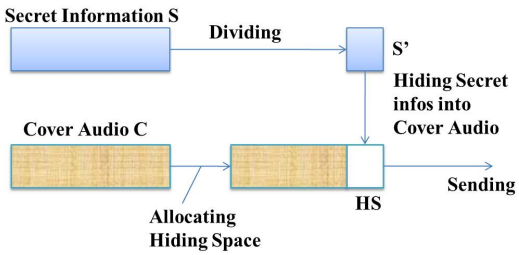

Figure 2. The Audio information Embedding Process

When using steganography, three factors are concerned, the cover file format, the hiding space and the steganography scheme. As the process shown in Figure 2, the steganography tools will firstly analyze the cover file format for suitable hiding space HS to allocate secret information. Then the steganography tool will split the secret information S into relatively small size blocks S`. Further, the tools will replace the original content in hiding space with those small blocks. As a result, the secret information has been embedded in the cover

file. In order to leak secret information, therefore, it only needs to transfer the steaganographied cover file.

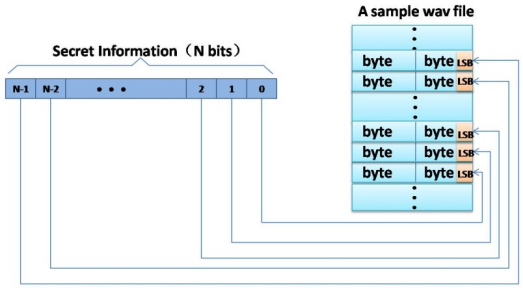

Figure 3. The information embedding process in LSB steganography

To specific, in this example shown in Figure 3, we use the G.711 (a wav file format) which consists of 160 voice vectors in each packet as a cover file. The suitable hiding space for G.711 format is the last the 4 bits of each vector. Finally, the algorithm will choose the scheme which decides where and when to embed. A simplest way, as the picture shows, is to embed information in the LSB (Least Significant Bit) alternatively. Latest steganography tool usually uses more complicated algorithm including pseudo-random embedding and Shamir secret sharing threshold scheme which enables the embedding processing in a more complicated way and leaving less steganography signatures.

IV. THE PROPOSED STEGAD SCHEME

With enterprises paying more and more attention on cloud storage, the security of cloud storage needs to be further improved. Fronting with the threat of steganography, it is necessary to develop proper defensive scheme to secure the data from leaking by steganography. Whereas traditional steganalysis tool is not able to fulfill job due to the limits on computing resources and consuming too much time. Therefore, in this paper, we propose a new scheme StegAD (Steganalysis Active Defense), to take an active role on detecting and eliminating information leaking by using audio steganography.

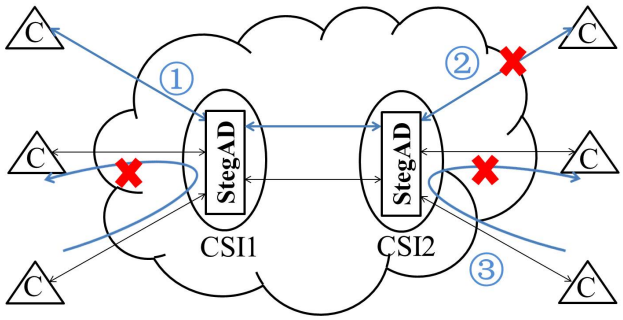

Figure 4. The StegAD Interface between CSIs

In StegAD, we firstly apply classical image signature detecting algorithm RS to hybrid cloud platform and use it to scanning for suspicious audio files. Further, we design and develop an interference algorithm, SADI (Steganographied Audio Dynamical Interference) to interfere the information in possible hiding place of the suspicious files. After replacing, SADI, based on comparing the original data and that afterward, manage to minimize the distortion in the interfering process.

A. *The enhanced-RS detecting algorithm*

Media data is highly time-relevant or space-relevant. In other words, the values, in an audio or image file, will remain rather steadily than change radically from time to time or pixel to pixel. However, when use steganography to embed information, the embedding process will disturb such consistency. Based on these abnormal changes, statistics approaches will help us find out which file is suspicious of using steganography. RS algorithm is using this characteristic on detecting steganography within pictures by scanning the grayscale changes between neighboring pixels[12][13]. Here, we use the algorithm definition and simulates the process on detecting audio files within the hybrid cloud storage.

1) *Audio detection with the enhanced RS algorithm*

Unlike image, audio files are stored in one dimension, namely time. With the time-consistency in audio files, it is possible use the time to simulate the RS process on grayscale. By adapting the concepts, we use the following definition to define the process:

Definition 1 ADM (Audio Data Matrix): a matrix in rows of audio data sampling binary values.

Definition 2 Neighboring Vectors Pair $< x_i, x_{i+1} >$: the pair of values of vertically neighboring binary values.

Then, according to the definition of the ADM and NVP, we define the following concepts:

Set R: Set consists of the NVP where the odd vector is bigger than the even vector where num_R stands for the number of elements in Set R.

$$R=\{<x_i,x_{i+1}>|(x_i \bmod 2=1, x_{i+1} \bmod 2=0, y_i>y_{i+1})|(x_i \bmod 2=0, x_{i+1} \bmod 2=1, y_i<y_{i+1})\}$$

Set S: Set consists of the NVP where the odd vector is bigger than the even vector where num_S stands for the number of elements in Set S.

$$S=\{<x_i,x_{i+1}>|(x_i \bmod 2=1, x_{i+1} \bmod 2=0, y_i<y_{i+1})|(x_i \bmod 2=0, x_{i+1} \bmod 2=1, y_i>y_{i+1})\}$$

Set U: Set consists of the NVP where vectors are both odds or even.

$$U=\{<x_i,x_{i+1}>|(x_i-x_{i+1}) \bmod 2=0\}$$

Ratio K: Define K as num_R to num_S ratio.

Generally speaking, common audio waves are randomly distributed. After analog to digital convert, the values of audio files also should be balanced where the number of odd value may equal to that of even value. That is to say, within a common audio file, $num_R \approx num_S$, namely $K \approx 1.0$.

However, when using steganography to embed information, by referring to LSB based statistical detecting algorithm, the K (i.e., num_R to num_S ratio) will increase with the steganography embedding. Therefore, by analyze the statistical data of ratio K, we are able to tell the whether the audio file is suspicious of steganography.

2) The numeric results for the enhanced RS algorithms

By applying RS algorithm on audio files, we choose a 16 bit wav audio clip as the test sample. When embedding in LSB, we use statistical measures to calculate the num_R and num_S to calculate the ratio K. The result is demonstrated in Figure 5.

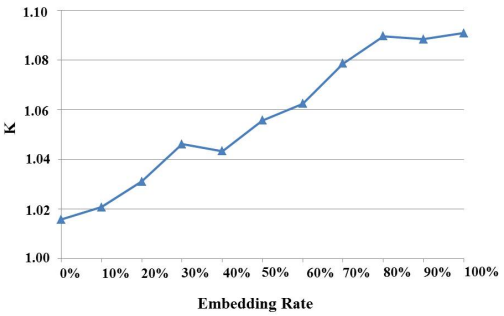

Figure 5. The K values with different Embedding quantity

B. The SADI algorithm

Though the RS algorithm manages to scanning files without requiring much computing resources, there exists one shortcoming that is several innocent files will be marked as suspicious. Thus the following protecting process must only destroy the steganographied files.

In StegAD, after acquiring suspicious files, we apply a scheme which is different from traditional ways. For suspicious files, rather than keep scanning for more data signature, we eliminate the leaking by deliberately replace information in common hiding places such as LSB with random noise. However, the attacking technique does help us to reduce the threats of steganography in hybrid cloud storage, but current steganography provide us with techniques, instead of simply put information in LSB, including pseudo-random sequence, multiple insertion and Shamir secret sharing. With those techniques, malicious users can embed information in a pseudo-random sequence, place more information and restore the information by parts of it.

That is to say, the interference has to be on multiple hiding places or more significant places without causing too much damage on innocent files. Therefore, we come up with the SADI in which we firstly using random noise to replace information including more significant one. Then we compare the previously unchanged information with that afterward and change the neighboring place to make up for the distortion. By doing so, steganography and innocent files will have different consequences. For steganographied files, the change in hiding place will likely to destroy the information hid in. At the same time, the innocent files also suffer from distortion during applying the deliberate interference. However, according to the principles of steganography, major distortion has been minimized by the algorithm and minor distortion will not obvious and annoying noises. We use the following principles to describe the SADI process for wav format.

- Firstly, we define the original file as C, the interfered file as C` and the file after repaired as C``. C` indicates the file is suspicious and has been interfered, so the distortion is between C and C`. We use the following procedures to offset the distortion.
- When the random inference happened in LSB, the distortion is no bigger than 1. So, such case needs no repairing.
- When the interference happened in the second or the third bit from the end, the SADI will calculate the Manhattan distance of each repairing result and choose the smallest one as the repairing approach.

ALOGRITHM 1. THE SADI ALGORITHM

Algorithm SADI for wav audio

Input: C, C', b_i

Output: C''

Procedure:

Var min, j, temp: integer;

1. **if** i is the position z
2. **then** $C'' \leftarrow C$
3. **else if** i is the position y or x
4. temp$\leftarrow C$; $b_{i-1} \leftarrow 0$; $b_{i+1} \leftarrow 0$; min$\leftarrow 8$;
5. **for** j$\leftarrow$0 to 3
6. $b_{i-1} \leftarrow$j/2; $b_{i+1} \leftarrow$j%2; calculate C'';
7. **if** min $< |C''-C|$
8. **then** min$\leftarrow |C''-C|$; temp$\leftarrow C$;
9. **end if**
10. **end for**
11. **else if** i is the position w
12. $b_{i+1} \leftarrow\sim b_{i+1}$; calculate C''; $\leftarrow \min\{C'', C'\}$;
13. **end if**
14. **return** C'';

When the interference happened in the fourth bit from the end, since it is the most significant bit in all possible hiding places in a wav file, the SADI will only choose the latter bit to implementing the repairing after calculating the Manhattan distance.

V. Experimental Results And Analysis

We conducted extensive experiments to evaluate the performance of StegAD.

In the first experiment we focus on the performance of modified RS detecting algorithm. In this experiment, we use OpenPuff v3.10, a famous steganography tool, to simulate the steganographied process [14]. Then we use the modified RS to scan both the original files and those steganographied ones. After calculating ratio K, the results indicate that when the threshold value is above 1.01 all steganographied files in this test can be found out.

In the second experiment, we test the interfering intensity of SADI on steganographied files by try to use SADI to unable the Shamir secret sharing restoring. Firstly, we introduce a CRR (Correct Replacement Ratio) to indicate how many steganograms have been interfered and replaced by random noise. Then we calculate CRR in different situation which proves in SADI when CRR is bigger than 20%, the common Shamir secret sharing is no longer effective.

In the third experiment, we assess the audio quality after using SADI. Firstly, we estimate the audio quality of 16 original sample files by using the ITU-R qualified audio testing software PEAQ [15]. Then we use the SADI technique to interfere with those 16 files and make acoustic compensation. The results reveal that the after acoustic interference and compensation the noise audio quality, according the ODG (Objective Difference Grade) standard, is perceptible but not annoying.

A. *RS-quick detection capacity*

In StegAD, the process will initially use modified algorithm to detect data signature and select the suspicious ones. Therefore the success rate of detecting steganographied files largely affects the success in defending data leakage through steganography. By the setting certain threshold value, the modified will mark those files that own a ratio K higher than threshold. Therefore we conduct the following experiments to set a proper threshold value to select the suspicious files while keep less innocent files involved.

Firstly, we choose 16 audio samples according to ITU-R standard. Then we use the OpenPuff to embed steganograms into those audio samples. After that, we use the modified RS algorithm to scan both the original and steganographied files and then calculate the ratio K. The results are illustrated in Figure 6.

From the result, we notice that some original files and steganographied files have similar K value. In order to set a proper threshold value to mark out all steganographied files with less innocent audio files involved, we introduce the following concept to help us to decide on choosing the threshold.

FAR (False Alarm Rate): the ratio indicates innocent files are mistaken as suspicious of steganography.

MAR (Missing Alarm Rate): the ratio indicates steganographied file has not been recognized.

CAR (Correct Alarm Rate): the ratio indicates the steganographied file is correctly recognized.

As shown in Figure 7, we choose threshold between 1.01 and 1.02, the FAR, MAR and CAR own different figures. When applying we focus on the performance on detecting steganography, therefore the MAR is expected be as small as possible. In the figure, MAR is 0% when the threshold value is 1.01. Even though with higher FAR indicating more innocent files might get involved, it is better to choose 1.01 as threshold value due to security concerns. In other words, when set to 1.01, modified algorithm is able to detect almost all steganographied files.

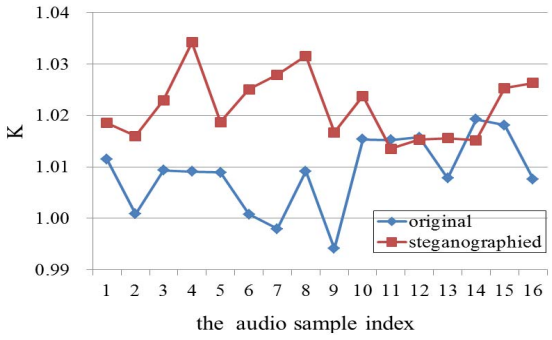

Figure 6. The K values with different audio samples

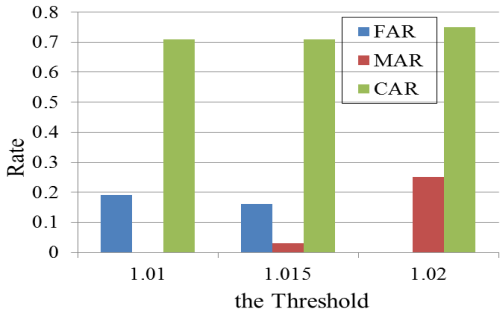

Figure 7. the FAR, MAR and CAR values with different thresholds

B. *Interfering intensity*

After set proper threshold value, StegAD will mark out the suspicious files. Then use the SADI to interfere the possible hiding space of those suspicious files. In this experiment, we will test the interfering intensity by trying to restore steganograms after SADI.

We introduce a parameter CRR (Correct Replacement Ratio) indicating that how much steganogram has been successfully replaced by random noise and E refers to the proportions of information embedded in audio. By calculating the CRR in different E, we are able to tell how many steganograms have been successfully replaced. In the figure 8, when embedding ratio E≥50%, the CRR is more than 20%. As the classical Shamir secret sharing usually needs more than 56.25% of information to restore, it is rather difficult to restore where nearly half the information has been devastated.

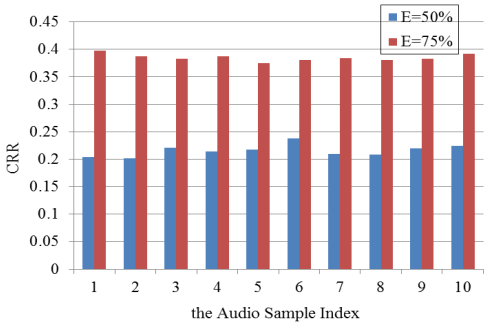

Figure 8. The CRR results with different audio samples

C. *Audio Quality*

From the previous experiments, the SADI interference is able to replace enough steganograms so that receiver can hardly retrieve the data in steganographied files. However, the interference will cause audio quality due to the random noise replacement. Therefore in this experiment we will measure the audio quality loss by ITU-R standard software and indexes.

In general, the Objective Difference Grade (ODG) provides a numerical indication of the perceived quality of received media after compression and transmission. The ODG value is expressed from -4 to 0, where -4 stands for the lowest quality and 0 stands for highest voice quality. ITU comes up with PEAQ algorithm [15] to estimate the codec quality quickly.

TABLE I.
THE K VALUES OF DIFFERENT EMBEDDING ALGORITHMS AND DIFFERENT EMBEDDING QUANTITY

ODG	Quality	Impairment
0.0	Excellent	Imperceptible
-1.0	Good	Perceptible, but not annoying
-2.0	Fair	Slightly annoying
-3.0	Poor	Annoying
-4.0	Bad	Very annoying

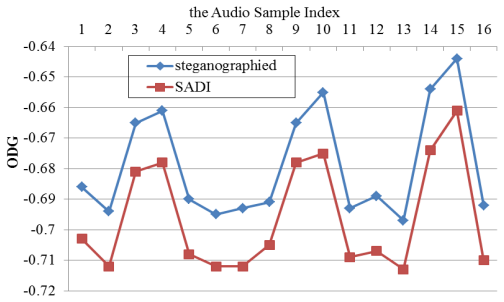

Figure 9. The ODG results with different audio samples

Standard unmodified wav audio will be 0 in ODG. We use the PEAQ to test original audio files, the steganographied files and SADIed files. After using SADI, though experiencing audio quality loss, the average quality loss is within 0.5 and SADIed files have, according to ODG standard, good quality.

VI. CONCLUSION AND FUTURE WORK

In this paper, we analyzed the Audio steganography Attacks in Cloud Storage Systems, and thereafter, we proposed and developed StegAD, a novel scheme for defending Audio steganography Attacks. StegAD consists of the enhanced-RS algorithm and the SADI algorithm. The enhanced-RS algorithm to detect the audio steganographied files, and after that, SADI was applied to interfere in possible hiding place with random noise and later compensate the damage. To evaluate the performance of StegAD, To evaluate the performance of StegAD, we built a prototype cloud storage system and conduct extensive experiments in terms of detecting steganographied files, interference effects on various steganography techniques and audio quality loss. Experimental results showed the efficiency of our proposed StegAD scheme.

ACKNOWLEDGEMENT

The work described in this paper is partially supported by the grants of the National Basic Research Program of China (973 project) under Grant No.2009CB320503, 2012CB315906; the National 863 Development Plan of China under Grant No. 2009AA01A334, 2009AA01A346, 2009AA01Z423; and the project of National Science Foundation of China under grant No. 61070199, 61003301, 60903223, 60903224; and supported by Program for Changjiang Scholars and Innovative Research Team in University of the Ministry of Education("Network Technology"), the Innovative Research Team in University of Hunan Province("Network Technology") ; and the University student Innovation project of Hunan Province(“Research and Prototype Design on New VoIP Streaming steganography Mechanisms”).

REFERENCES

[1] Wikipedia Encyclopedia, “Cloud storage,” Internet, August 2011, Available: http://en.wikipedia.org/wiki/Cloud_storage. September 2011

[2] J. Strickland, “How Cloud Storage Works,” Internet, Available: http://computer.howstuffworks.com/cloud-computing/cloud-storage.htm. September 2011

[3] NASUAI, “Understanding Security in Cloud Storage”, A NASUNI WHITE PAPER , 2010

[4] A. Marshall, M. Howard, G. Bugher, B. Harden, “Security Best Practices For Developing Windows Azure Applications,” Microsoft, June 2010

[5] Black Hat Team, “[GET] Amazon S3 Ripper,” Internet, October 2009, Available: http://www.blackhatteam.com/f51/get-amazon-s3-ripper-9770.html, September 2011

[6] NASUAI, “Storage Services Overview,” Internet, Available: http://www.nasuni.com/product/product-overview/, September 2011

[7] W. Mazurczyk, and K. Szczypiorski, “Is Cloud Computing Steganography-proof,” Internet, Avaiable: http://arxiv.org/ftp/arxiv/papers/1107/1107.4077.pdf, August 2011

[8] SARC, “Steganography Analyzer Real-Time Scanner (StegAlyzerRTS),” Internet, Available: http://www.sarc-wv.com/products/stegalyzerrts/, September 2011

[9] Andrew D. Ker, “Quantitative evaluation of pairs and RS steganalysis,” Proceedings of the SPIE, Volume 5306, pp. 83-97, 2004

[10] Mustafa Ulutas, Vasif V. Nabiyev, and Guzin Ulutas, “Improvements in Geometry-Based Secret Image Sharing Approach with Steganography,” Mathematical Problems in Engineering, Hindawi, 2009

[11] Anil Kumar, and Navin Rajpal, “Application of T-code, turbo codes and pseudo-random sequence for steganography,” Joumal of Computer Science, pp.148-153, 2006

[12] Fridrich, J., Goljan, M., Du, R. “Reliable detection of LSB steganography in color and grayscale images,” In: *Proceedings of the 2001 workshop on multimedia and security*: new challenges, Ottawa, Ontario, Canada, ACM, October 2001

[13] G. T. Juan You, “Statistics detection algorithm based on audio lsb steganography,” *Computer Engineering of CHINA*, vol. 35(24), pp. 176–177, 2009

[14] EmbeddedSw, “OpenPuff Yet not another steganography SW,” Internet, Available: http://embeddedsw.net/OpenPuff_Steganography_Home.html, September 2011

[15] ITU-R Recommendation BS.1387-1, “Method for objective measurements of perceived audio quality,” [S]. Jan 2001