EMITTER International Journal of Engineering Technology
Vol.1, No.1, December 2013
ISSN : 2443-1168

Implementation and Analysis Audio Steganography Used Parity Coding for Symmetric Cryptography Key Delivery

Afany Zeinata Firdaus, Mike Yuliana, Mochamman Zen Samsono Hadi

Department of Telecommunication Engineering
Department of Electrical Engineering
Electronic Engineering Polytechnic Institute of Surabaya
Jl . Raya ITS , Sukolilo , Surabaya 60111
Tel : +62-31-5947280 , Fax : +62-31-5946114
Email : afany_02@yahoo.co.id , mieke@eepis-its.edu , zenhadi@eepis-its.edu

Abstract

In today's era of communication, online data transactions is increasing. Various information even more accessible, both upload and download. Because it takes a capable security system. Blowfish cryptographic equipped with Audio Steganography is one way to secure the data so that the data can not be accessed by unauthorized parties. In this study Audio Steganography technique is implemented using parity coding method that is used to send the key cryptography blowfish in e-commerce applications based on Android. The results obtained for the average computation time on stage insertion (embedding) the secret message is shorter than the average computation time making phase (extracting) the secret message. From the test results can also be seen that the more the number of characters pasted the greater the noise received, where the highest SNR is obtained when a character is inserted as many as 506 characters is equal to 11.9905 dB, while the lowest SNR obtained when a character is inserted as many as 2006 characters at 5,6897 dB .

Keywords: audio steganograph, parity coding, embedding, extractin, cryptography blowfih.

1. INTRODUCTION

In recent communication era like nowadays, online data transactions has increased compared to the previous communication era of just voice and text messaging. To improve the security of data access privacy, it would require a system that is not easy to know the content of the information by third parties that are not desirable. One such system is a cryptographic security Blowfish method. cryptography is defined as the science and art of maintaining the confidentiality of messages by encrypting way to form that can not be understood anymore meaning [3]. Blowfish is a symmetric block

cipher algorithm with three major parameters , namely : word, rounds, and the number of bytes in the secret key [3]. The drawback of this method is the need for the same key in the shuffle (encryption) and return (decryption) of the data. If the key can be hacked by a third party, it will be very easy to take the desired data . Therefore it is necessary security systems to hide the key.

One of the other security systems is an audio steganography where steganographic (steganography) is the science and art of hiding messages within other messages that the existence of the first message is unknown [1]. In hiding the message, there are several criteria that must be met [2], namely:

1. Impercepbility. The existence of the message can not be perceived by the visual.
2. Fidelity. Medium reservoir quality does not change much as a result of the insertion.
3. Recover. The hidden message should be revealed again.

In this study, the cover - object used is audio.wav (cover - audio). Audio steganography technique used is parity coding. Through this method, the original data information will not be easily identified and removed from the parent file (audio cover). The data will be more secure, because if the data is successfully hacked by a third party, then all you see is an audio file.

2. RELATED WORKS

In previous studies conducted by Lisa M. Marvel [4] , it has been made a steganographic method using error control coding, image processing, and spread spectrum. This method is used to hide the secret message in the cover image without increasing the size or the dynamic range of the image. In addition, the original image is not required to take the hidden message, and a requirement that the sender and receiver have a common key to decrypt secret messages in images .

Then, on audio steganography research conducted by Rico Arlando Saragih [6], it has been compared methods parity coding and spread spectrum. From the results of tests performed is seen that the parity coding method, the value of SNR at audio signals cover a longer duration have a better value, while the method of spread spectrum data security is more secure because it uses code spreader that is not known by the other party.

This study is implementation of audio steganography techniques with parity coding method and create applications for cryptographic key delivery in blowfish .

3. ORIGINALITY

Several studies [4] [5], usually use steganography to hide data parity coding secret messages in the media image, but in this study the parity coding method is used to hide the secret message in an audio medium. Parity Steganography Coding System created integrated with e-commerce applications based on Android, and is equipped with blowfish cryptographic method that security of data sent over secure.

4. SYSTEM DESIGN

4.1 Method of Parity Steganography Audio Coding

Steganography technique using parity coding is the process of counting the bits with even parity condition [1]. The result of the calculation is checked, when bit 1 odd number, then the parity value of the bit is 1. If bit 1 an even number, then the parity value of the bit is 0 [2]. A description of the work system Steganography using parity coding method is divided into two parts, the part of the sender of the embedding process (insertion) and the receiver section extraction (removal).

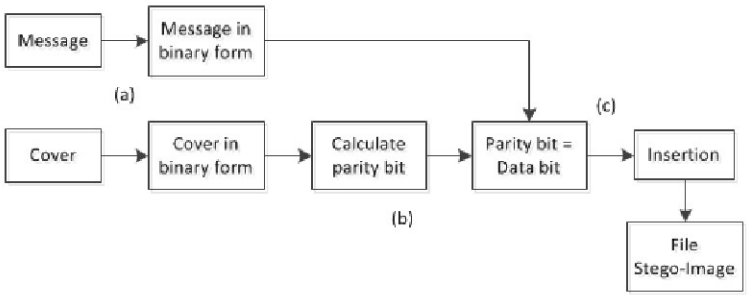

Figure 1. Process flow diagram insertion (embedding) [8]

Figure 1 is a process of embedding a secret message into the cover. Here is the working procedure of parity coding method [2]:

a) File messages and Cover-Audio converted in binary form.
b) Cover-Audio files sorted and counted its corresponding RGB is even parity, in order to combine with the message file.
c) If the sum of the parity bits are not equal to one bit of the message it needs to change the value of the LSB of the RGB (if one is replaced with 0, and vice versa), the insertion can be done because both have the same bits, and generates the stego-audio files.

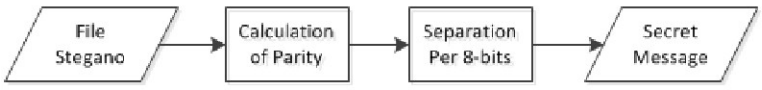

Figure 2. Flowchart making process (extraction) [8]

Extraction is done is the opposite of the data hiding process. The process is carried out initially is the parity calculation of stego-file. Then, the result is divided by 8-bits to be translated into the character, so the secret message will be retrieved.

4.2 Method of Cryptography Blowfish

Blowfish methods included in the class of Symmetric Cryptosystem and made to be used on computers that have a large microposesor (32-bit or more with large data caches) [3]. A description of the system using the Blowfish cryptographic work is divided into three parts, the parts of the

sender and encryption key generation process, then part of the receiver decryption.
Here is the key generation step :
a) Initialize the P - array and 4 S - Box
b) XOR P1 with the first 32 bits of the key, XOR P2 with the second 32 bits of the key, and so on up to P18
c) Encrypt the all - zero string with the Blowfish algorithm
d) Replace P1 and P2 with the output of step 3
e) Encrypt the output of step 3 with blowfish
f) Replace P3 and P4 with the output of step 5

Blowfish algorithm has 16 iterations and 64 bit data input " X ". The steps in the encryption process is as follows :
a) For X into 2 Xl and Xr
b) Iteration i = 1 to 16 do :
Xl = Xl Xor P (i)
Xr = F (Xl) Xor Xr
Exchange Xl and Xr
c) Iteration to 16, change Xl and Xr and do :
Xr = Xr Xor P (17)
Xl = Xl xor P (18)
d) Combine back Xl and Xr

Decryption process is similar to the encryption process, only sub - keys are used in reverse order, namely P (1) to P (18) , P (2) to P (17) and so on. The pseudocode of this process is

```
i= 1
loop from I to 16
      Ri = Li-1 XOR P19-i
      Li = F(Ri) XOR Ri-1
end loop
      L17 = R16 XOR P1
      R17 = L16 XOR P2
```

4.3 Equipment and Materials

In this research, the design of the support device that includes :
1. Hardware (Hardware)
   - Asus A43SV - VX072D
   - Smartphone Samsung Galaxy Y S5360 , Android 2.3.3 Gingerbread OS
2. Software (Software)
   - Netbeans version 7.1.1
   - Apache Tomcat version 7.0.21
   - phpMyAdmin version 3.4.5
   - Eclipse Indigo

4.4 System Design

In this section described the overall integration of the entire system to the data processing . The workflow of the system as shown in Figure 3 are :

- From the user side, there is the Android application that acts display a list of products, handle and store the member registration process for securing data Steganography Blowfish cryptographic keys used to encrypt user data (username and password), which is then sent to the Network.
- Before sending user data (username and password), we send first Blowfish cryptographic key for decryption on the server side in the form of the stego - audio.
- The data in the form of username and user passwords , encryption and cryptography blowfish Blowfish key cryptography in the form of the stego - audio toward the network (Network).
- Once the data is up on the server, the stego - audio files are processed to remove infomasi a secret decryption key for the Blowfish cryptography, and the key was used to decrypt blowfish cryptographic containing user data (username and password). Then the data is used for authentication.
- Once successful, a confirmation will be sent to the successful login the user side.

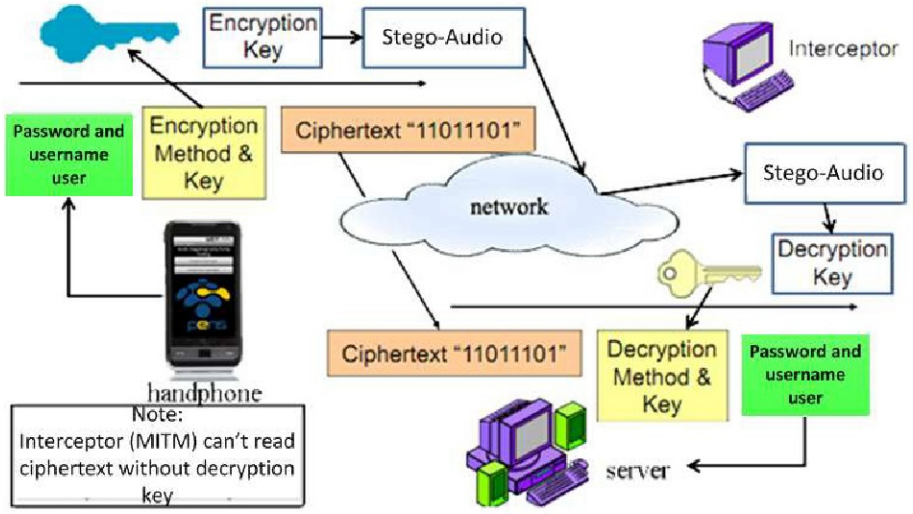

Figure 3. integration System

5. EXPERIMENT AND ANALYSIS

The primary outcome in this study was a comparison between the original image (cover image) with an image that has been filled message (stego-images), as shown in Figure 4.

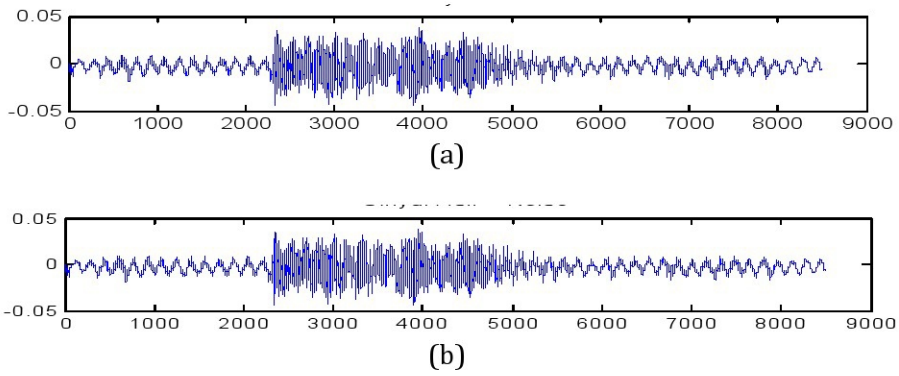

Figure 4. (a) the original audio file (audio cover), (b) The audio files have filled the secret message (stego-audio)

Table 1. Comparison of the average time for the process

Process	Average time (ms)
CLIENT SIDE	
Cryptography	
Key random generation process	0,56
Encryption process	7,47
Steganography	
File wav reading process	0,39
Process of conversion from wave to	11,53
Form integer bit and content separation	
Conversion process from integer array to byte array	0,79
Parity clculation process	10,41
Conversion process of secret string to byte	0,054
Adjustment Process	1,53
Rearrange Process of wave file	0,3
SERVER SIDE	
Steganography	
Reading process of file wave	0,28
Conversion process from wave to	12,27
Form integer byte and content byte	
Conversion Process from integer array to byte array	0,81
Parity calculation process	11,52
Message taken process	160,6
Cryptography	
Descrypt process	0,025

From Figure 4 can be compared between the original audio (cover-audio) with audio that has been filled message (stego-audio), and visible by naked eye that there is no difference between them.

Comparison of average computational time with the insertion of a secret message secret message retrieval computation time can be seen in Table 1. Where the results of the collection of processes that occur, the insertion process requires less time than the process of making a secret message, this happens because when extracting need more time to check the bits one by one.

Testing the maximum number of characters entered also conducted to determine the effect of the addition of characters that have exceeded the maximum number of characters that can be entered. Audio file types used are "a.wav" who has a file size of 16,6 kb and a bit rate of 64 kbps. From the results in Table 2 shows that the wav file with a file size of 16,6 kb can accommodate a maximum of 2122 characters. It can be seen that when given an input of 3006 characters, which reads only 2122 characters.

Table 2. The results of testing the maximum character

The amount of input characters	The amount of output characters	Result
106	100	Success
506	500	Success
1006	1000	Success
1506	1500	Success
2006	2000	Success
3006	2122	Failed

Furthermore, calculation of the Signal to Noise Ratio (SNR). Where SNR is used to determine the ratio between the cover audio and noise, noise in question is the energy difference between the original audio signal (cover-audio) with audio pitch-stego (stego-audio). SNR value calculation based on the number of characters entered.

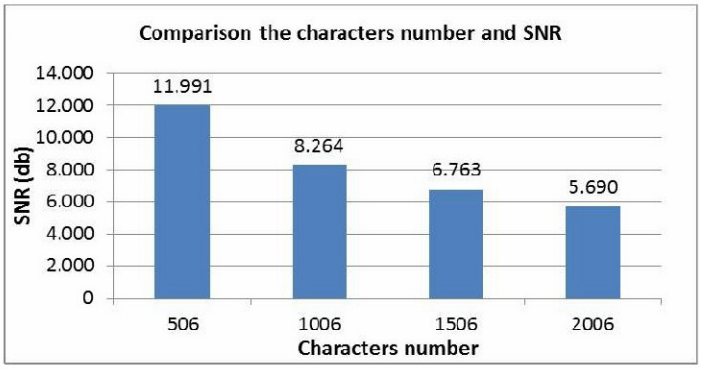

Figure 5. SNR test result charts

From the data in Figure 5, it can be seen that the lowest SNR value when the number of characters entered in 2006 as a character and the highest SNR value when 506 characters. So, the more the number of characters entered, the smaller the value of its SNR. For the record, the greater the value of the SNR, the smaller value of the stego audio noise.

Testing with the addition of noise is also performed to check the robustness of audio stego-noise ratio. This test menggunakani a.wav which has a file size of 16.6 kb and a bit rate of 128 kbps. This test uses a.wav with a secret message inserted is "# # # coba1234!!!". The test results are shown in Table 3 below:

Tabel 3. The test results on the addition of noise a.wav

Noise Energy (dB)	Result
8.5×10^{-5}	Failed
8.5×10^{-7}	Failed
8.5×10^{-9}	Success
8.5×10^{-10}	Success

From Table 3 shows that the maximum noise energy that can be added to be able to take back the secret messages of the audio file is at 8.5 x 10-9 dB.

6. CONCLUSION

Based on the results of the implementation and analysis of the system obtained in this study some conclusions as follows :

1) File exctracting results form the stego-audio files showed no difference when compared with the cover-audio files.
2) When embedding , the average time required to perform the process is 33,03 ms. Meanwhile, when extracting, the average time required to perform the process was 185,51 ms
3) The original audio file (audio cover) before embedding and after embedding process produces an audio file (stego-audio) are exactly the same and has not changed either in sound quality (bit-rate) as well as the size of the file.
4) There is a correlation between the maximum number of characters in embedd the stego audio sound quality. The more characters yng entered, the greater the noisenya.
5) WAVE files with a size of 16,6 Mb with a bit rate of 128 Kbps can accommodate a maximum of 2122 characters.
6) The amount of noise that can be retained by stego audio with size 16,6 Kb with 128 Kbps bit-rate is 8.5 x 10-9 dB .

REFERENCES

[1] Brahim Wijaya, **Implementation And Analysis Of Steganography Using Parity Coding In E-Commerce**, Proyek Akhir Politeknik Elektronika Negeri Surabaya, 2012.

[2] Herianto, **Pembangunan Perangkat Lunak Steganografi Audio MP3 dengan Teknik Parity Coding pada Perangkat Mobile Phone**, Proyek Akhir Institut Teknologi Bandung, 2008.
[3] Mike Drogalis, Dylan Reeder, **The Blowfish Cipher**, Oktober 2012.
[4] Marvel, L.M., C.G. Boncelet, Jr., dan C.T. Retter, **Spread Spectrum Image Steganography**, Image Processing, IEEE Transactions , Vol. 8 , Issue: 8 , 1998.
[5] Putri Alatas, **Implementasi Teknik Steganografi Dengan Metode LSB Pada Citra Digital**, Proyek akhir Universitas Gunadarma Jakarta, 2009.
[6] Riko Arlando Saragih, **Metode parity Coding Versus Metode Spread Spectrum pada Audio Steganography**, Proyek Akhir Universitas Kristen Maranatha Bandung, Bandung, Juni 2006.