The Use of Multiple MP3 Audio Files for Additional Steganography Capacity

R Indrayani1*, H A Nugroho2, R Hidayat2, Dony Ariyus1, Y Pristyanto1

1Faculty of Computer Science, Universitas Amikom Yogyakarta, Yogyakarta, Indonesia
2Department of Electrical Engineering and Information Technology, Universitas Gadjah Mada, Yogyakarta, Indonesia

E-mail: rini.i@amikom.ac.id*

Abstract. Steganography is the science and art of hiding secret messages in such a way so that the existence of messages is undetected by human senses. In this paper, multiple MP3 files are used as cover with the aim of increasing steganographic capacity without limiting the number of used files. Concealing secret messages in the form of files on the MP3 file cover can be performed by using homogenous frame. Homogeneous frames in the MP3 files store the bits of the inserted secret files. Before being inserted into some MP3 cover files, secret files are first encrypted employing AES algorithm with passwords implicated by MD5 processes. Audio quality changes between steganography MP3 files and original MP3 files are measured by using stereo signal strength and Mean Opinion Score (MOS). The test results show that the distribution of steganography load on multiple MP3 files can fulfill the user’s needs for the capacity of steganography without reducing the quality of used MP3 files. This paper proposed a method about the increase of steganography capacity which is capable to insert and retrieve secret files on multiple MP3 audio files using homogenous frame utilization methods without resizing and reducing the quality of used MP3 files.

1. Introduction

Data and information security is an issue that has become one of the particular concerns in current technological developments. The ease of exchanging digital files over the internet allows any file exchange can be done in an instant way, and any files can go through it. However, no one can guarantee that such activity will be safe from various file theft activities such as eavesdropping. Therefore, steganographic science is required to help users hide secret files into other container files to prevent third parties from knowing the existence of those files.

Research on steganography have been tested on various media such as texts, images, audios, and videos. In audio steganography, a secret message is inserted by slightly altering the binary sequence of sound file. In a study, a classical steganographic method namely Least Significant Bit (LSB) was modified as in the study of Abdul et al. [1] which involved each sample of audio data. Data concealment in an audio sample proposed in this technique. The bit range of each audio sample related to the use of audio bits as a cover. The smaller range, the less bits can be inserted.

A research that used McElieic cryptosystem to encrypt the secret message was conducted by Salman [2]. The encrypted message inserted in the ID3V2 tag with BAF method. The album cover part used to keep encryption code. Secret message security using McElieic encryption can withstand
the steganalysis activity, but this method does not provide maximum insertion capacity. Steganography uses HLLAS Technique orienting on Least Significant Bit which utilizes every 4 bits of hidden message characters to be converted to hexadecimal form and calculated using modulo operator with the value of MP3 signal audio represented in hexadecimal form. Mohammed et al. [3] classifies the types of steganography in which researchers once described as Embedding Before Compression, Embedding During Compression, and Embedding After Compression. Embedding After Compression method is considered as the best method for steganography method due to its simple implementation and relatively maximize the capacity, ignoring the fact that it is not superior in compression resistance to MP3. One example of Embedding After Compression is a study by Philip et al. [4] which is the utilization of Huffman trees on steganography. The result of MP3 file compression is used as a medium of secret message insertions. Some properties of Huffman trees that have the same characteristic and length codeword are exploited into a supersymbol. Thus, there is redundancy in the Huffman tree structure that can be utilized to hide information.

Yoeseph et al. [5] incorporated spread spectrum and secret sharing method on multiple MP3 files. This research orientates towards the secret sharing method of Shamir's secret sharing, the first method in steganography on how to share secret messages [6]. This was done for security purposes. Information dissemination gives benefits to the user because the third party will not be easily able to collect secret messages which then gradually become a complete secret message.

Yoeseph et al. [5] incorporated spread spectrum and secret sharing method on multiple MP3 files. This research orientates towards the secret sharing method of Shamir's secret sharing, the first method in steganography on how to share secret messages [6]. This was done for security purposes. Information dissemination gives benefits to the user because the third party will not be easily able to collect secret messages which then gradually become a complete secret message.

Prihadana et al. [7] also used a spread spectrum method implemented on homogeneous frame of a single MP3 file. The secret message is first encrypted using Elgamal's encryption method. Then the secret message is spread to every available frequency spectrum. However, this method increases the redundancy level, according to the size of secret message. The larger the size of the inserted secret message, the greater the noise it generates.

This study uses multiple MP3 file as a container media of steganography. The use of multiple MP3 files can increase the capacity and security of steganography. In terms of capacity, the user can specify the number of MP3 cover files used in accordance with the size of the secret file. In terms of security, the advantage of using multiple MP3 files is that the secret message is not gathered on a single MP3 cover file, thus makes third parties difficult to collect secret messages spread over several MP3 files.

The system uses MD5 hash function for password randomness and AES encryption to secure confidential messages. The use of encryption and hash functions improves the security aspect for secret messages are encrypted using keys that also experience in randomization. The purposes of this study are to increase steganography capacity in MP3 audio files and improve security aspects using AES encryption and MD5 hash function.

2. Methodology

This study uses multiple MP3 files as container media of steganography. The use of multiple MP3 files can increase the capacity and security of steganography. In terms of capacity, the user can determine the number of MP3 cover files used in accordance with the need of steganography capacity. In terms of security, the advantage of using multiple MP3 files is that the secret message is not gathered in a single MP3 cover file. Therefore, third parties will find it difficult to collect secret messages that are spread over several MP3 files.

The proposed technique consists of five main processes: steganography load sharing, secret message encryption, secret message insertion, secret message extraction, and decryption of secret messages. In the steganography load sharing stage, the percentage of steganography loads is calculated according to the total steganography capacity of all multiple MP3 files and secret message files used. As this study applies homogeneous frames of MP3 files as a medium for secret messages, the total
steganography capacity depends on the availability of homogeneous frames of MP3 covers used. Homogeneous frame is a part of MP3 structures that has a homogeneous value other than the header and side information. If all bytes of an MP3 frame are homogeneous, it means that the frame can be used as a place for data insertion. Examples of homogeneous frames are shown in Figure 1.

At the encryption stage, the secret message is encrypted using 128-bit Advanced Encryption Standard (AES). The study also employs MD5 hash function for password randomness used as a Chiperkey on AES algorithm.

At the insertion stage, a secret message is inserted into some MP3 audio files used. The insertion procedure depends on the number of homogeneous frames owned by the MP3 file used. The next stage is the extraction phase of secret messages, the stage where parts of secret messages are collected from some MP3 files used. The last stage is the process of decryption. The decryption process is the opposite of the encryption process. At this stage, the secret message collected at the extract stage is decrypted using 128-bit AES algorithm and involves MD5 hash function as in the encryption stage. In general, the general system scheme is shown in Figure 2.

---

**Figure 1.** Homogeneous Frame.

At the encryption stage, the secret message is encrypted using 128-bit Advanced Encryption Standard (AES). The study also employs MD5 hash function for password randomness used as a Chiperkey on AES algorithm.

At the insertion stage, a secret message is inserted into some MP3 audio files used. The insertion procedure depends on the number of homogeneous frames owned by the MP3 file used. The next stage is the extraction phase of secret messages, the stage where parts of secret messages are collected from some MP3 files used. The last stage is the process of decryption. The decryption process is the opposite of the encryption process. At this stage, the secret message collected at the extract stage is decrypted using 128-bit AES algorithm and involves MD5 hash function as in the encryption stage. In general, the general system scheme is shown in Figure 2.

---

**Figure 2.** General System Scheme

---

2.1. Steganography Load Sharing

Steganography load sharing is measured by the number of MP3 cover files used, the total Max stego,
and the size of the secret message. Max stego is the steganographic capacity of each single MP3 used. Max Stego is the steganography capacity of each single MP3 used. The secret message can only be inserted only if the size is not greater than the Max stego value measured by byte units. Max stego is measured by using (1).

\[
\text{Max Stego} = \sum (uf \cdot 36)
\]

(1)

The variable of fs is the size of homogenous frames which is used to steganography process. While the value of the deductible 36 is the allocation of headers which should not be changed as it contains the information of the core structure of the MP3 used. According to [9], the size of the homogeneous frame is measured by variable uf as formulated in (2).

\[
uf = \frac{144 \cdot \text{bitrate}}{\text{sample frequency}} + \text{padding}
\]

(2)

Bitrate, sampling rate, and padding are parts of specially allocated MP3 structures at the beginning of the frame. The bitrate is at the 17th bit to the 20th bit. The sampling frequency is in the 21st bit to the 22nd bit. The padding is on the 23rd bit. The bits are decoded according to the bitrate allocation table and the sample frequency in the ISO MPEG-1 Layer 3.

Once the system determines the secret message size and the number of max stego of each MP3 file used, the percentage of the steganography load can be calculated using (3).

\[
B = \frac{s}{S_{\text{total}}} \times 100\% (Y)
\]

(3)

Where,

- \(s\) = max stego
- \(S_{\text{total}}\) = total max stego of all MP3 cover files used
- \(B\) = percentage of max stego capacity
- \(Y\) = secret message size

Steganography load sharing illustrated on Figure 3 shows each MP3 file used will be filled by separated secret messages. For example, there is a user using multiple MP3 files whose each steganography load sharing are 50%, 25%, and 25%. System will divide the secret messages into 3 parts based on their respective percentages.

| Part 1 | Part 2 | Part 3 |
|--------|--------|--------|
| (50%)  | (25%)  | (25%)  |

**Figure 3.** Example of Steganography Load Sharing

2.2. Encryption

AES encryption has 10 rounds consisting of addroundkey, bytesub, shiftrows, and mixcolumn processes [10]. Before entering the first round, passwords are first processed using MD5 hash function. Then the password of hash result is processed by AES key scheduling along with the plaintext. The type of AES encryption used is 128 bit AES encryption that requires a key number of 16 bytes of characters. Since MD5 is a hash function with 16 bytes of output[11], it allows the user to choose a steganographic password, it means that the user does not have to enter a password of 16 characters, Because the MD5 already fit the characters of password to be 16 character. The Encryption process is shown in Figure 4.
Figure 4 shows SubBytes, ShiftRows, MixColumns and AddRoundKey processes. All of these processes result in strong diffusion and confusion. In AddRoundKey operation, XOR operations are performed between State blocks with Chiperkey blocks. In SubByte process, the substitution transformation changes the byte value based on the original value and the value that is in the S-box [12]. That process is an intrabyte transformation. Permutation generated from ShiftRows process exchanges the bytes without changing the bit values in bytes. This transformation is a byte-exchange transformation (byte exchange). In MixColumn, an interbyte transform will be performed to change the bit value in bytes. The goal is to generate diffusion at the bit level.

2.3. Embedding
The embedding process is a process where a secret message is inserted into homogeneous frames of MP3 covers used as a container medium.

In MP3 files that do not use CRC, generally the main data location starts after the 36th byte to the next frame, so byte replacement method in steganography will be done from the 37th byte to touch the next frame. It is expected that by replacing the contents of the main data which is homogeneous, the
sound contained in the frame will not change. The homogeneous frame check is by checking the byte after the header (36 bytes). Then the system will check whether the 37th byte (calculated from FF FB or FF FA) has the same value as the 38th byte. If not, then checking will be shifted to the next byte.

The insertion of multiple MP3 files is done in accordance with the results of the steganography load sharing calculation. The secret message is inserted on each homogeneous frame of every single MP3 file according to the allocated capacity. The insertion begins by inserting a secret key code identifier that will be useful when retrieving a secret message. The insertion of the lock code then followed by the insertion of the secret message. After the insertion process, the homogeneous frame’s value will changed. Bytes of the homogeneous frame will be transformed into bytes of encrypted secret messages. As the result, homogeneous frames are transformed into heterogeneous frames. An example of the result of inserting a secret message on a homogeneous frame process is shown in Figure 5.

| FF | FB | 90 | 64 | 00 | 0F | F0 | 00 | 06 | 08 | 0D | 20 | 55 | 55 |
|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 55 | 55 | 55 | 55 | 55 | 55 | 55 | 55 | 55 | 55 | 55 | 55 | 55 | 55 |

![Figure 5. Result of Embedding Process](image)

2.4. Extraction
The extraction process begins by selecting multiple MP3 files to be extracted. User must correctly choose all MP3 files to be extracted, because if the user does not select the entire MP3 cover completely, then the secret message cannot be completely returned as well. After the user selects the MP3 file that will be extracted, the system will check the file whether or not it contains secret data. The checking process involves a lock code. It is used as a secret message identifier. If the system found a lock code, then the system will begin collecting secret messages that follow the lock code.

2.5. Decryption
The opposite of AES encryption process called decryption process [13]. The cipher transform can be reversed and implemented in the opposite direction to produce an easily understandable inverse cipher for AES algorithm[14]. The byte transformations used in the inverse cipher are InvShiftRows, InvSubBytes, InvMixColumns, and AddRoundKey. Figure 6 shows AES decryption scheme.
3. Result and Analysis
The implementation result of the proposed multiple files usage methods are tested from various aspects namely Recovery, Robustness, Security, Bit Modification Rate, and signal strength changes.

3.1. Percentage of Steganography Load Sharing

| Secret Message (byte) | MP3 Cover | Size of MP3 Cover (byte) | Max Stego (byte) | Used Space (byte) | Used Space Percentage (%) |
|-----------------------|-----------|--------------------------|------------------|-------------------|---------------------------|
| 9479                  | Unknown.MP3 | 500000                   | 382              | 53                | 0.56                      |
|                       | Best.MP3   | 220000                   | 121              | 17                | 0.18                      |
|                       | Kudituki.MP3 | 320000                 | 11808            | 3458              | 79.35                     |
|                       | davichi    | 961000                   | 3394             | 9409              | 99.26                     |
| 4358                  | nokia.MP3  | 222300                   | 121              | 35                | 0.81                      |
|                       | romantic.MP3 | 222000               | 2955             | 865               | 19.85                     |
|                       | guitar.MP3 | 430000                  | 3696             | 2102              | 22.18                     |
| 9479                  | nikhil.MP3 | 120000                  | 5347             | 3042              | 32.09                     |
|                       | flowersun.MP3 | 640000            | 7620             | 4335              | 45.73                     |

Figure 6. Decryption Process.
The percentage of steganography load sharing is the percentage of MP3 cover space used for steganography on secret message size, calculated using equation 3 and shown in Table 1. Table 1 shows that used space percentage is depends on max stego. The big max stego value has affect to the big used percentage value.

3.2. Recovery Testing
The recovery aspect in steganographic science is that the inserted secret data must be fully recovered without any change in structure and size[15]. Then, at this stage, testing will be done several times to ensure that the secret data inserted in the MP3 can be recovered in a full state. The recovery aspect test is shown in Table 2.

| MP3 Cover                | Secret Message | Size Before Embedded (byte) | Size After Extraction (byte) |
|--------------------------|----------------|------------------------------|-----------------------------|
| Armada – Pemilik hati.MP3| Citra_uji_1.jpg| 7310                         | 7310                        |
| 01 Lirih – Chrisye.MP3   | Citra_uji_2.png| 2010                         | 2010                        |
| Adera – Lebih Indah.MP3  | Citra_uji_3.bmp| 20                           | 20                          |
| Renungan.MP3             | Teks_uji_1.txt| 324                          | 324                         |
| Depapepe – Time.MP3      | Metodologi.docx| 1020                        | 1020                        |

In Table 2, it appears that the secret message on all test data can be re-extracted with the same size. That is, this steganography method meets the recovery aspect of data security.

3.3. Security Testing
Security aspects of steganography can be tested by utilizing the science of steganalysis. In this security testing, Stegsecret steganalysis application is used to test whether or not secret messages are detected. Test results shown in Table 3 indicates that hidden secret messages with the proposed technique are not detected by the application Stegsecret.

| MP3 Stegged  | MP3 Size (Byte) | Max Stego (Byte) | Secret File | Secret File Size | Inspection Status |
|--------------|-----------------|------------------|-------------|------------------|-------------------|
| Unknown.MP3  | 500000          | 382              | Uji1.txt    | 120              | Not Detected      |
| Best.MP3     | 220000          | 121              | Uji2.txt    | 150              | Not Detected      |
| Kudituki.MP3 | 320000          | 67597            | Uji3.txt    | 5000             | Not Detected      |
| Davichi.MP3  | 961000          | 11808            | Uji4.txt    | 1001             | Not Detected      |
| Nokia.MP3    | 222300          | 121              | Uji5.txt    | 104              | Not Detected      |

3.4. Testing of Audio Signal Strength
Testing the strength of the audio signal is done by comparing the value of the audio signal strength called left peak signal and right peak signal before and after the secret file inserted. The assessment of left peak signal and right peak signal is tested on stereo type audio files, shown in Table 4.
Table 4. Testing of Audio Signal Strength.

| MP3 Size (byte) | Max Stego (byte) | Stego Secret Message Size (byte) | Left peak signal $P_0$ (dB) | Left peak signal $P_1$ (dB) | Right peak signal $P_0$ (dB) | Right peak signal $P_1$ (dB) |
|-----------------|-----------------|---------------------------------|---------------------------|---------------------------|---------------------------|---------------------------|
| 4138196         | 29445           | 13904                           | -10.58                    | -10.58                    | -11.21                    | -11.21                    |
| 3602389         | 34359           | 16                              | -13.04                    | -13.04                    | -13.00                    | -13.00                    |
| 3172498         | 102392          | 17440                           | -14.83                    | -14.83                    | -14.11                    | -14.11                    |
| 1360089         | 114523          | 56976                           | -8.97                     | -8.97                     | -8.85                     | -8.85                     |
| 3666156         | 548810          | 386656                          | -14.83                    | -14.83                    | -8.85                     | -8.85                     |

3.5. Bit Modification Rate

Bit Modification Rate is a benchmark of bit changes that occur in MP3 cover audio files before and after the secret message is inserted. The bit changes that occurred were calculated using (4) and shown by Table 5.

$$M = \left( \frac{k}{N} \right) \times 100\%$$  \hspace{1cm} (4)

Where, $k =$ number of MP3 cover bits that have changed
$N =$ total number of MP3 cover bits

Table 5. Bit Modification Rate.

| Size of Secret Message (bit) | N (bit) | k (bit) | M (%) |
|------------------------------|--------|--------|-------|
| 320                          | 2678272| 154    | 0.0057|
| 800                          | 1805096| 370    | 0.0205|
| 1600                         | 2582976| 844    | 0.0029|
| 2000                         | 942912 | 1103   | 0.0326|
| 4000                         | 420856 | 2085   | 0.4954|

3.6. Mean Opinion Score

Subjective assessment of MP3 stego audio quality is conducted by using MOS (Mean Opinion Score). The subjective values were taken based on listeners’ assessment of the MP3 stego after the original audio MP3 was played. The MOS rating can be seen in Table 6. The assessment involved 30 respondents, 5 original MP3 audio files that had not been subjected to steganography, and 5 MP3 stego audios. Data of respondents’ assessment results can be seen in Table 7. MOS values in Table 7 show that qualitatively, MP3 stego audio quality is the same as the original one.

Table 6. MOS Rate.

| MOS Value | Opinion Score | Explanation |
|-----------|---------------|-------------|
| 5         | Excellent     | MP3 stego quality is much better than the original MP3 quality |
| 4         | Good          | The quality of MP3 stego is better than the original MP3 |
| 3         | Fair          | The quality of MP3 stego is the same as the original MP3 |
| 2         | Poor          | The MP3 stego quality is worse than the original MP3 quality |
| 1         | Bad           | The MP3 stego quality is much worse than the original MP3 quality |

Table 7. Results of Qualitative Assessment of Respondents.

| Original MP3     | MP3 stego | Total Opinion Score | MOS Value |
|------------------|-----------|---------------------|-----------|
| Unknown.MP3      | Unknown_STEGO.MP3 | 90                   | 3         |
| Best.MP3         | Best_STEGO.MP3   | 90                   | 3         |
| Kudituki.MP3     | Kudituki_STEGO.MP3| 90                   | 3         |
| Davichi.MP3      | Davichi_STEGO.MP3| 90                   | 3         |
| Nokia.MP3        | Nokia_STEGO.MP3  | 89                   | 2.967     |
4. Conclusion
Optimization of steganography capacity with homogeneous frame utilization techniques and security optimization using AES encryption and MD5 hash function are successfully applied to multiple MP3 audio files. The capacity that can be accommodated by an MP3 depends on the number of frames containing homogeneous bytes, rather than depending on the size of the MP3. Therefore, not all MP3 can be used as a container of message. Only MP3 with frames containing homogeneous bytes can be used as MP3 message containers (MP3 Cover). However, the use of multiple MP3 files can increase steganography capacity because the system adds max stego of every MP3 cover file used. The advantage of this method is the aspects of capacity and security optimization can go hand in hand. The use of multiple files can increase the steganography capacity according to the user’s needs and improve security because third parties have to collect all MP3 covers completely. Addition of MD5 hash function and AES decryption add the security aspect since third parties must decrypt if they want to get the secret messages they desire.

References
[1] A. Hakeem, N. U. Amin, M. Shah, Z. Khan, and A. Qadi, 2014, “Threshold Based LSB Audio Steganography”, Int’l Conf. on Chemical Engineering & Advanced Computational Technologies, pp. 88–92.
[2] Salman, 2014, “Steganography Application Program Using the Id3V2 in the Mp3 Audio File on Mobile Phone,” J. Comput. Sci., vol. 10, no. 7, pp. 1249–1252.
[3] M. S. Atoum, S. Ibrahimn, G. Sulong, A. Zeki, and A. Abubakar, 2014, “Exploring the challenges of MP3 audio steganography,” Proc. - 2013 Int. Conf. Adv. Comput. Sci. Appl. Technol. ACSAT 2013, pp. 156–161.
[4] P. C. Ritchey and V. J. Rego, 2012, “Hiding secret messages in huffman trees,” Proc. 2012 8th Int. Conf. Intell. Inf. Hiding Multimed. Signal Process. IHIM-MSP 2012, pp. 71–74.
[5] N. M. Yoeseeph, F. A. Purnomo, B. K. Riasti, M. A. Safiie, and T. N. Hidayat, 2016, “Steganography on multiple MP3 files using spread spectrum and Shamir’s secret sharing,” J. Phys. Conf. Ser., vol. 776, no. 1.
[6] P. Feldman, 1987, “A practical scheme for non-interactive verifiable secret sharing,” 28th Annu. Symp. Found. Comput. Sci. (sfcs 1987), pp. 427–437.
[7] P. E. Kresnha and A. Mukaromah, 2014, “A Robust Method of Encryption and Steganography Using ElGamal and Spread Spectrum Technique Based on MP3 Audio File,” Proceeding Conf. Appl. Electromagn. Technol., pp. 11–15.
[8] H. Ghasemzadeh, 2019, “Multi-layer architecture for efficient steganalysis of UnderMp3Cover in multi-encoder scenario,” IEEE Trans. Inf. Forensics Secur., vol. 14, no. 1, pp. 186–195.
[9] R. Indrayani, H. A. Nugroho, R. Hidayat, and I. Pratama, 2016, “Increasing the Security of MP3 Steganography Using AES Encryption and MD5 Hash Function,” Proc. 2016 International Conference on Science and Technology-Computer, pp. 2–5.
[10] A. Conci, A. L. Brazil, S. Bacellar, L. Ferreira, and T. Machenri, 2015, “AES Cryptography in Color Image Steganography by Genetic Algorithms” Proc. International Conference of Computer Systems and Applications (AICCSA).
[11] B. P. Gajendra, V. K. Singh, and M. Sujeet, 2016, “Achieving Cloud Security using Third Party Auditor , MD5 and Identity-Based Encryption,” International Conference on Computing, Communication and Automation, pp. 1304–1309.
[12] P. Ceminari, A. Arelovich, and D. Federico, 2017, “AES block cipher implementations with AMBA-AHB interface,” Conference on PhD Research in Microelectronics and Electronics Latin America, pp. 1–4.
[13] A. Msolli an and A. Helali, 2016, “Image encryption with the AES algorithm in wireless sensor network,” International Conference on Advanced Technologies for Signal and Image Processing, pp. 41–45.
[14] B. Sung, K. Kim, and K. Shin, 2018, “An AES-GCM authenticated encryption crypto-core for IoT security,” International Conference on Electronics, Information, and Communication,
[15] Lindawati and R. Siburian, 2018, “Steganography implementation on android smartphone using the LSB (least significant bit) to MP3 and WAV audio,” Proc. - ICWT 2017 3rd Int. Conf. Wirel. Telemat. 2017, vol. 2017–July, pp. 170–174.