An Improved AES Key Generation in Audio Steganography

D. N. Purnamasari\textsuperscript{1} and A. D. Ramadhani\textsuperscript{2}

\textsuperscript{1} Department of Informatics Engineering, University of Trunojoyo Madura, Bangkalan – Madura, Indonesia
\textsuperscript{2} Department of Electrical Engineering, Politeknik Elektronika Negeri Surabaya, Surabaya - East Java, Indonesia

Abstract. The higher Internet use also affects the level of information security in the digital world. Nowadays, many users send private messages or even secret messages through the Internet. This has an impact on the increasing number of illegal people who try to see and even change the information for personal gain. There is a solution to this problem by inserting a secret message in a file such as images, audio, or video. To increase the level of security, secret messages are encrypted first using cryptographic techniques. In this paper, we propose a method for developing key generation in audio steganography techniques using the AES (Advanced Encryption Standard) algorithm. We use additional parameters to create the AES key, i.e., the user's identity. The user's identity is entered into the key table which is then randomly selected by the system when the user wants to get the key. There is a combination process and hash function to add complexity to the key. The performance of the proposed method is tested based on computing time and audio signal strength. The test results show that the message inserted in the audio file does not affect the file size, but rather affects the audio signal strength. Audio signals that experience additional data are at frequencies of 15 to 20 kHz and for other frequencies only experience additional noise.

Keywords— AES; audio steganography; user identity;

1. Introduction

The widespread use of the internet as a place to exchange data has many advantages such as in terms of time and cost-efficiency. However, due to the exchange of data via the internet, users must also consider the security of the data sent. This is because the data sent is digital data that is easy to retrieve and change by unauthorized parties (commonly called attackers), so the information received will be different from the information sent.

Cryptography is one way to hide or encode a message so that the message will not be readable by others. Cryptography has two processes, namely encryption, and decryption. The encryption process is a process of conversion from the original message (plaintext) to a message that has been coded (ciphertext). While the description process is a process to change or return from ciphertext to the original message (plaintext). Both processes require a key as a parameter of a message [1-3]. Cryptography is classified into 3 based on keys used for encryption and decryption including symmetric cryptography.
that uses the same key [4-7], asymmetric cryptography that uses different keys [8-10], and hashes that do not use keys in encoding data.

Besides cryptography, one technique for hiding messages is steganography [1-19]. Steganography is a technique for hiding messages in multimedia data such as pictures [4-6] [8-9] [13], audio [1-3] [7] [10-11] [14-15], or video [16-19]. The basic principle of steganography is to insert message data bits into digital data bits, but it does not significantly change data bits in digital data. Steganography is fundamentally different from cryptography. Cryptography completely hides messages to encoded messages, while steganography uses larger data to store data that is smaller and more sensitive. As technology develops, there are currently many tools for analyzing steganography so that messages will be easier to detect. So the cryptographic process needs to be done before the message is inserted into digital data. One algorithm that can be combined with steganography is the AES (Advanced Encryption Standard) algorithm. This is because AES is a reliable algorithm in encrypting a message and has been widely adopted in hardware and software [20].

In this paper, we propose a development method to generate the AES key. Before starting the key generation, the system must have identity data from the user which will be entered into a table called the key table. For example, there are two users namely A and B. User A has the name Alice and user B has the name Bob. Then the system will retrieve and combine the two names to get a digest of the user's identity. The results of digest are used as one component at the time of AES key formation so as to increase key security. The proposed method has been implemented using steganography with audio data (*.wav) inserted by encrypted messages. There is a sampling process using the FFT (Fast Fourier Transform) algorithm to convert audio data from the time domain to the frequency domain. The sampled audio data and the ciphertext are then encoded to produce audio steganography. The results of this scheme have been tested to prove the efficiency of the proposed method.

2. Methods

In this paper, we propose a development method for generating AES keys using a key table. The key table contains the user's identity such as name, address, telephone number, e-mail, and others. The proposed scheme is implemented in an audio file (*.wav) using steganography techniques. The system design designs used in this paper are shown in Figure 1.

![Figure 1. Overview System](image_url)
In this system design, we divide the focus of work into three parts including the key generation process, the encryption and decryption process, and the audio steganography process. In this paper, we use several different audios to test the performance of steganography techniques.

In the key generation process, we use the identity of the user who wants to communicate. Previously, users had to submit their credentials to be entered in the key table such as name, address, telephone number, email, and others. Each identity will be given a unique code to facilitate the user when retrieving the key. For example, Alice wants to communicate with Bob then Alice sends a request to the key server to get a Public Key (Pk). On the key server, Alice and Bob's identity will be randomly selected and combined to get a digest of Alice-Bob's identity. Furthermore, the digest that is obtained becomes one of the parameters of AES key generation to increase the uniqueness and security level of the key. Illustration of key generation is shown in Figure 2.

![Figure 2. Illustration of Key Generation](image)

In the encryption and decryption process, we use the AES-256 algorithm to encrypt messages in the form of data strings. Please note that the AES-256 algorithm has a 256-bit cipher key and 14 rounds, each round consisting of AddRoundKey, SubBytes, ShiftRows, and MixColumns. Before the original message is encrypted, the message will be converted into 128-bit data blocks. The results of the encryption process in the form of ciphertext containing the original message, while the results of the decryption process in the form of the original message.

In the process of audio steganography, we insert the encryption results in the form of ciphertext in the audio file. There is a sampling process using the FFT (Fast Fourier Transform) algorithm to convert audio data from the time domain to the frequency domain. The system will only take a portion of the audio data then look for its basic amplitude and then insert the ciphertext. So there are parts of the audio file that don't get the encoded bits.

3. Experiment and results

In this paper, there are two tests that have been carried out namely the performance of the proposed method based on computational time and signal strength. The method has been implemented simulating a client-server scenario. The purpose of this study is to analyze the capabilities of the proposed method.

The performance testing of the proposed method is carried out with a client-server scenario. The client is responsible for sending secret messages in the form of data strings that are inserted in the audio file. Before the message is sent, the client is also tasked to encrypt the message so that the information
inserted in the audio file is ciphertext. While the server is responsible for getting secret messages from the audio decode and decrypt ciphertext cover.

In this test, we observe the file size and computational time needed for the system to run the entire system. The test was carried out 10 times with 10 different audio files. Table 1 shows the results of testing file size and computational time for each process.

| Cover Audio (*.wav) | Original Size (MB) | Audio Stego Size (MB) | Encrypt Time (s) | Decrypt Time (s) | Encode Time (s) | Decode Time (s) |
|---------------------|--------------------|-----------------------|------------------|------------------|----------------|-----------------|
| Audio 1             | 6.98               | 6.98                  | 0.031            | 0.0              | 3.058          | 1.919           |
| Audio 2             | 4.27               | 4.27                  | 0.031            | 0.0              | 2.902          | 1.797           |
| Audio 3             | 4.52               | 4.52                  | 0.031            | 0.0              | 2.886          | 1.729           |
| Audio 4             | 3.81               | 3.81                  | 0.031            | 0.0              | 2.770          | 1.601           |
| Audio 5             | 3.70               | 3.70                  | 0.031            | 0.0              | 2.761          | 1.608           |
| Audio 6             | 3.32               | 3.32                  | 0.031            | 0.0              | 2.671          | 1.583           |
| Audio 7             | 3.04               | 3.04                  | 0.031            | 0.0              | 2.650          | 1.560           |
| Audio 8             | 5.73               | 5.73                  | 0.031            | 0.0              | 2.976          | 1.779           |
| Audio 9             | 10.2               | 10.2                  | 0.031            | 0.0              | 3.338          | 2.231           |
| Audio 10            | 5.91               | 5.91                  | 0.031            | 0.0              | 2.988          | 1.803           |

Based on Table 1, the file size in the original audio file and after inserting the message has no difference. This states that the addition of ciphertext data that has been done does not affect the file size. Then we need to observe the computational time required by the system to run each process. The encryption time needed by the system is 0.031 seconds for different audio file sizes and the decryption time needed by the system is 0.0 seconds. This can be said to be a reliable system because the time required for the encryption and decryption process is very fast that is less than 1 second. As for the encoding process, the system requires an average of 2.9 seconds and 1.76 seconds for the decoding process. The computation time is obtained by inserting messages in audio files with sizes of 3 to 10 MB.

In addition to computing time, we also observe the signal strength of audio files in the frequency domain. In this test we use 6 audio files consisting of 3 original audio files and 3 others are the result of steganography. The results of testing the audio signal strength in the frequency domain are shown in Figure 3.
Comparison of audio files before and after encoding has an impact on the audio signal in it. It can be seen in Figure 3, before being encoded, a signal with a frequency of 15 to 20 kHz has a signal strength of around -150 dB / kHz. While the audio file that has been encoded, the signal with a frequency of 15 to 20 kHz has increased to reach -100 dB / kHz. This states that the inserted ciphertext file is placed at a frequency of 15 to 20 kHz so that it affects the shape and strength of the signal at that frequency. Whereas at other frequencies, the audio signal does not have a significant change or there is only noise that does not change the shape of the signal.

4. Conclusion
We propose a development method to generate the AES key with the encryption results inserted in the audio file. The purpose of this paper is to determine the performance of the proposed method. The proposed method has been tested based on computing time and audio signal strength. The test results show that the message inserted in the audio file does not affect the file size, but rather affects the audio signal strength. Audio signals that experience additional data are at frequencies of 15 to 20 kHz and for other frequencies only experience additional noise.

Future works. Our future work is developing input data and algorithms that can be inserted in various files.

5. References
[1] N. Kaur, S. Behal, “Audio Steganography Techniques-A Survey”, Journal of Engineering Research and Application, vol.4, no.6, pp. 94-100, 2014.
[2] A. Binny, M. Koilakuntla, “Hiding Secret Information Using LSB Based Audio Steganography”, International Conference on Soft Computing and Machine Intelligence, pp. 56-59, 2014.
[3] I.U.W. Mulyono, A. Susanto, T. Anggraeny, C. A. Sari, “Encryption of Text Message on Audio Steganography Using Combination Vigenere Cipher and LSB (Least Significant Bit)”, Kinetik: Game Technology, Information System, Computer Network, Computing, Electronics, and Control, vol.4, no.1. pp. 63-74, 2019.
[4] U. Sheth and S. Saxena, "Image steganography using AES encryption and least significant nibble," 2016 International Conference on Communication and Signal Processing (ICCSP), Melmaruvathur, 2016, pp. 0876-0879.
[5] A. Conci, A. L. Brazil, S. B. L. Ferreira, and T. MacHenry, "AES cryptography in color image steganography by genetic algorithms," 2015 IEEE/ACS 12th International Conference of Computer Systems and Applications (AICCSA), Marrakech, 2015, pp. 1-8.
[6] P. P. Bandekar and G. C. Suguna, "LSB Based Text and Image Steganography Using AES Algorithm," 2018 3rd International Conference on Communication and Electronics Systems (ICCES), Coimbatore, India, 2018, pp. 782-788.
[7] R. Indracyani, H. A. Nugroho, R. Hidayat, I. Pratama, “Increasing the security of MP3 Steganography using AES Encryption and MD5 hash Function”, International Conference on Science and Technology-Computer (ICST), 2016.
[8] M. Jain, R. C. Choudhary, and A. Kumar, "Secure medical image steganography with RSA cryptography using decision tree," 2016 2nd International Conference on Contemporary Computing and Informatics (IC3I), Noida, 2016, pp. 291-295.
[9] S. Mittal, S. Arora, and R. Jain, "PData security using RSA encryption combined with image steganography," 2016 1st India International Conference on Information Processing (IICIP), Delhi, 2016, pp. 1-5.
[10] A. Gambhir and S. Khara, "Integrating RSA cryptography & audio steganography," 2016 International Conference on Computing, Communication and Automation (ICCCA), Noida, 2016, pp. 481-484.
[11] M.S. Atoum, S. Ibrahim, G. Sulong, A. Zeki, A. Abubakar, “Exploring the challenges of MP3 audio steganography”, Int. Conf. Adv. Comput. Sci. Appl. Technol. ACSAT 2013, pp. 156–161, 2014.

[12] S. Saraireh, “A secure data Communication System Using Cryptography and Steganography”, International Journal of Computer & Communications (IJCNC), vol.5, no.3, 2013.

[13] U. Sheth, S. Saxena, “Image Steganography using AES Encryption and Least Significant Nibble”, International Conference on Communication and Signal Processing, 2016.

[14] M.S. Atoum, S. Ibrahimn, G. SUlong, A. Zeki, A. Abubakar, “Exploring the Challenges of MP3 Audio Steganography”, International Conference on Advanced Computer Science Applications and Technologies, 2013.

[15] C.T. Jian, C.C. Wen, N.Hidayah, I. Rahmi, “Audio Steganography with Embedded Text”, IOP Conf. Series: Materials Science and Engineering, 2017.

[16] T.V.S.G. Prasad, S. Varadarajan, “High Payload and Secured Video Steganography Using Compression and AES Crypting System”, International Journal of Applied Engineering Research, vol.10, no. 11, pp. 27835-27844, 2015.

[17] P.A.S.L.E.Ningsih, G.M.A.Sasmita, N.M.I.M.Mandenni, “MP4 Video Steganography Using Least Significant Bit (LSB) Substitution and Advanced Encryption Standard (AES)”, Journal of Theorical and Applied Information technology, vol.95, no.21, 2017.

[18] R. Balaji and G. Naveen, "Secure data transmission using video Steganography," 2011 IEEE International Conference on Electro/Information Technology, Mankato, MN, 2011, pp. 1-5.

[19] P. Yadav, N. Mishra, and S. Sharma, "A secure video steganography with encryption based on LSB technique," 2013 IEEE International Conference on Computational Intelligence and Computing Research, Enathi, 2013, pp. 1-5.

[20] M. Pitchaiah, P. Daniel, Praveen, “Implementation of Advanced Encryption Algorithm”, International Journal of Scientific & Engineering Research, vol.3, 2012.