2016 International Conference on Information Technology Systems and Innovation (ICITSI)
Bandung – Bali, October 24 – 27, 2016
ISBN: 978-1-5090-2449-0

Design and Implementation of Stegokripto

Ahmad Almaarif
School of Electrical Engineering and Informatics, Institut Teknologi Bandung, Bandung 40132, Indonesia

Budi Rahardjo
School of Electrical Engineering and Informatics, Institut Teknologi Bandung, Bandung 40132, Indonesia

***Abstract*—This paper is about Stegokripto, an information concealment method based on noise. Basically, Stegokripto is a combination between steganography and cryptography. The message is encrypted with an encryption algorithm, in this case AES and the encrypted messages is inserted into the noise. By using noise as a cover messages, the messages will have two kind of protections; encrypted messages and noise as diversion. This kind of protection will increase the level of security of the messages being transferred. Hence the effort of attack is not only focused on breaking the encrypted messages, but also finding the right messages to be broken.**

Keywords—Steganography; Cryptography; Noise; Advanced Encryption Standard (AES); Stegokripto

I. INTRODUCTION

According to Cisco Visual Networking Index, the global IP traffic in 2016 will reach 1.1 zettabytes and in 2020 the global internet traffic will be equivalent to 95 times of total global internet traffic in 2005 [1]. With this enormous amount of traffic, information security becomes more concerned. Until today, there are many methods proposed by researchers to secure the information. The most common one is cryptography. Cryptography is a method to scramble the messages in order to prevent unauthorized party to read the content of the messages. Another method worth mentioning is steganography. Steganography is a method to secure the information by concealing their existence. Both of these methods have their own benefits. Even though cryptography can prevent the unauthorized party to read the content of messages, the existence of messages is known to the attacker. With this information, the attacker can perform cryptanalysis to the encrypted messages and the messages could be extracted within a certain time with a certain resources. Meanwhile, the information concealed in steganography can be analyzed once the existence of messages known to the attacker.

Instead of replacing cryptography, steganography can acts as a complementary [2]. Johnson and Jajodia [3] stated that if these methods combined, it can results a stronger security method. By referring to this theory, many recent studies have put efforts into combining cryptography and steganography to increase information security [4] [5].

Steganography usually uses images, audio, video and text as its cover messages. This cover messages hides the real messages so the unauthorized party could not aware of the existence of that messages. However, many attempts to find other kind of cover messages have been conducted. Some of them use network protocol [4], 3D Images [8], and ECG Signals [9] as cover messages.

In this presented paper, we use noise to hide information. We propose a design and implementation of a method called Stegokripto, a concept introduced in [6] and [7] as a combined steganography and cryptography method by mixing noise with encrypted data to increase security. For the purpose of experiment, we use AES as encryption method and PRNG as noise source. The remainder of this paper is organized as follow: in section II, we briefly introduced the Stegokripto concept and its design. The implementation and experiment results are described in section III while conclusions are described in section IV.

II. STEGOKRIPTO METHOD

A. System Overview

Steganography hides the existence of actual messages in a cover messages. In this case, we use noise as cover messages. The fundamental idea of Stegokripto is to hide encrypted messages in the stream of noise so the attacker cannot differentiate between them. The process of implementing this concept is shown in Fig. 1. A message is first encrypted with any encryption algorithm. Instead of sending this encrypted message, we embed them into stream of noise generated by noise generator. The combined form of noise and encrypted message is called stego-noise. Stego-noise should be noise-like stream so the attacker could not detect its existence.

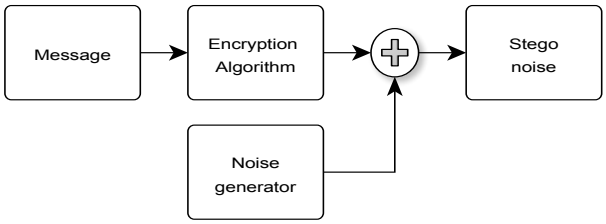

Fig. 1. Concept of Stegokripto

The process of Stegokripto consists of two sub process: data embedding and data extraction. Data embedding process occurs in sender side while data extraction process in receiver side.

B. Data Embedding

The process of data embedding is shown in Fig. 2. Message is first encrypted using encryption algorithm. The encrypted message is divided into sequences of encrypted messages with a certain length. We called each of this sequence as *segment*. We used fixed size length in order to simplify the extraction process in the receiver side.

We use a stego key called *flag* that is placed as header of each segment. Noise generator will generate noise stream which later is mixed with the segment. In order to add randomness into the stream, each segment waits for a random number of times before mixed into the stream of noise.

The following is the main design guideline and assumption of our approach:

1) Cryptographic key exhanged mechanism are assumed known to sender and receiver. This is to avoid to explain the key exchanged protocol used in this system. This Stegokripto approach is intended to be compatible with any encryption algorithms available.

2) Flags are assumed known to all users. Flag is one of the most important component of this system. It differentitates between noise and encrypted message. Despite its importance, this design is a proof of concept and further research is needed to choose the most secure way to exchange or generate flags.

3) Noise generated by noise generator are assumed to have high similarities with encrypted message. As one of the main security parameter of this method is undetectability, it strength lies in randomness of stego-noise generated by the system. We assumed that encrypted message is random enough to be like-noise so the attacker does not suspicious to the stego-noise we generate.

4) Encryption algorithm is assumed known to all users. This assumption is to limit the explanation to the broader concept of Stegokripto. As stated before, this concept is intended to be compatible with any encryption algoritms.

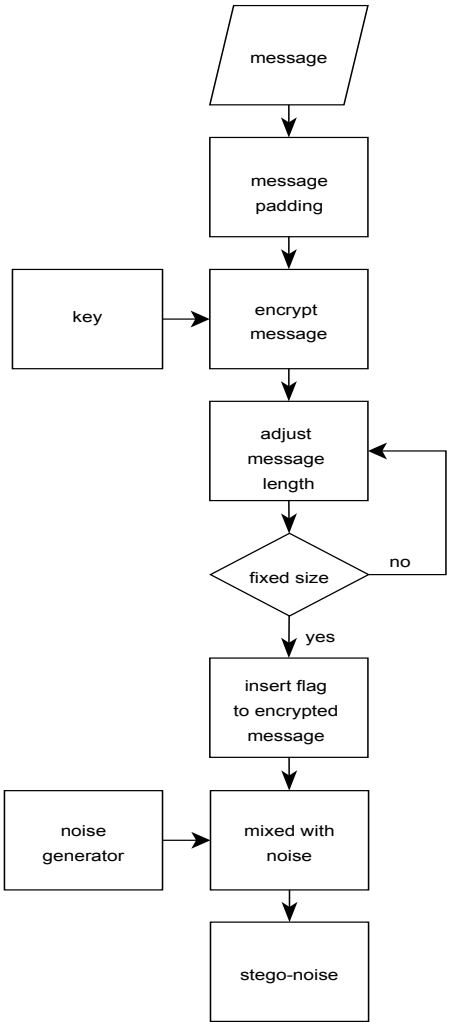

Fig. 2. Data embedding in Stegokripto

C. Data Extraction

Receiver extracts data by separating noise and encrypted message as shown in Fig. 3. This process is conducted by detecting the flag in the stream of noise. Once the flag is detected, receiver will acquire a certain bits of noise, assuming that they are the encrypted message. The decryption algorithm will be run in order to extract the message. There are some considerations in this process as stated below.

1) The unintended recurrence of flag in noise. There is a possibility of unintended presence of flag in noise. This could happen if the possibility of flag recurrence is high, e.g., the flag size is too small.

2) The appropriate length of flag. As mentioned above, if the flag size is too small, the possibility of recurrence in noise is high. But if the flag size is too large, there is high possibility to detect the flag for a large sequence of flag is present in

every segment. In order to generate the appropriate flag, further research is necessary.

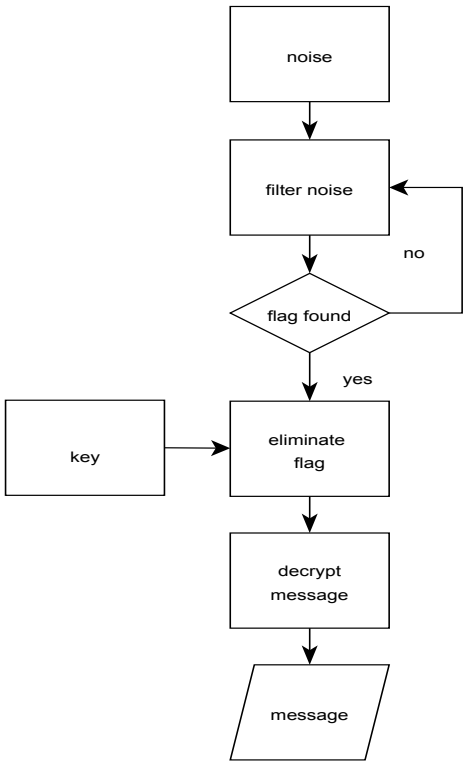

Fig. 3. Data extraction in Stegokripto

III. IMPLEMENTATION

The implementation of Stegokripto is conducted by using RabbitMQ as middleware. This experiment created using Python and PyCrypto as cryptography library. As the testing environment, we used Windows operating system with 3 GB RAM and processor Intel Core 2 Duo @2.2 GHz.

We use AES-128 as encryption algorithm. Key exchange mechanism is processed using assumption mentioned earlier. In this paper, we focus on the functionality of this concept. There are some scenarios that are used in this implementation:

A. Functional Testing in Normal Condition

Normal condition is described as normal conversation and sentences that is usually used in real life. We use Leipzig Corpora as the input to test the functionality of the program. Using this corpus, we found that every sentence can be decrypted by the receiver normally. Below is the sample of sentence we used.

"*1. Are you concerned about the direction this country is headed?*"

The program encrypted the sentence and then separated it into segments with "ffff" as flag. The results of this process is

ffff241d2c0c31e35e3b56046f39d102bce5
ffff130985979a5501875958048c7449d2b4
ffff0535183e12092c106310cfdb349cb93d
ffffe05d5150d88234fdea90e39c89da43d8
ffff23fd0264b745b6cf7aa6a21d58f05fb1

Noise generator will mixed the encrypted message with noise. For the randomness purpose, the program waits for a random seconds to send the next segment into the noise. From our experiment, the receiver will receive all segments sequentially.

B. Functional Testing in Extreme Condition

We define extreme condition as the condition that rarely happens during conversation. We test some example for this condition; e.g. message with one character, message with 10KB, 32KB, and 64KB. The results are the same with the normal condition.

C. Performance Testing

Python datetime function is being used in this experiment. We measure the difference between the arrival time (t_1) and the sending time (t_0) of each segment by using this formula.

$$\Delta t = t_1 - t_0$$

According to this testing, the mean of time for every segment to arrive at receiver side could be less than one second. Despite that, for a very large message the timing will be an issue for every message will be fragmented into some segment and each segment needs time to arrive at the destination. Another reason is because every segment need to be mixed with noise so the delay time caused by noise should also be concerned.

D. Security Issues

Theoretically, the security of this algorithm has been explained in [6], [7]. Some security issues have also been described in previous research. Further research is necessary.

IV. CONCLUSION

In this paper we proposed the design and implementation of Stegokripto. This concept encourages the use of noise as additional security. By concealing information into noise, the attacker will need extra cost to attack information.

The concept of Stegokripto is implemented by using *flag* as a key for receiver to filter the real message from noise and applying random time in delivering every segment into the noise. This implementation has been tested by using functional testing and performance testing. In the future, further research about the flag choices as stego-key is necessary, together with the research about the security of the implementation of this concept.

REFERENCES

[1] CISCO. “Cisco Visual Networking Index: Forecast and Methodology 2015-2020”.[Online]. Available at: http://www.cisco.com/c/en/us/solutions/collateral/service-provider/ip-ngn-ip-next-generation-network/white_paper_c11-481360.html. [Accessed at: 9-June-2016]

[2] Al-ani, Z.K., Zaidan A.A., Zaidan, B.B., and Alanazi, H.O., “Overview : Main Fundamentals for Steganography”, Journal of Computing, 2010, vol. 2, pp. 158–165.

[3] Johnson, N.F. dan Jajodia, S., “Exploring Steganography: Seeing the Unseen”. Computer Magazine, 1998, vol. 31, pp. 26–34.

[4] Ciobanu, R., Tirsa, M., Lupu, R., Stan, S., Andreica, M.I., "SCONeP: Steganography and Cryptography Approach for UDP and ICMP”, RoEduNet International Conference 10th Edition: Networking in Education and Research, 2011, pp.1–6.

[5] Marwaha, P., dan Marwaha, P., “ Visual Cryptographic Steganography in Images”. International Conference on Computing, Communication and Networking Technologies, 2010

[6] Rahardjo, B., Kuspriyanto, Muchtadi-Alamsyah, I., Paryasto, M.W., “Information Concealment Through Noise Addition”, International Conference on Graph Theory and Information Security, 2015, vol.74, pp. 178-180.

[7] Rahardjo, B., Muchtadi-Alamsyah, I., Paryasto, M.W., “Noise-based Stego-ECC”, International Conference on Advances Science and Contemporary Engineering, 2014.

[8] Chao, M., Lin, C., Yu, C., Lee, T., “ A High Capacity 3D Steganography Algorithm”, IEEE Transactions on Visualization and Computer Graphics, 2009, vol. 15, pp.274–284.

[9] Jero, S.E. dan Ramu, P., Curvelets-based ECG Steganography for Data Security. Electronic Letters, 2016, vol. 52, pp. 4–5.