EPJ Web of Conferences **68**, 00013 (2014)
DOI: 10.1051/epjconf/20146800013

Noise-based Stego-ECC

Budi Rahardjo[1], Intan Muchtadi-Alamsyah[2] and Marisa Paryasto[3]

[1]School of Electrical Engineering and Informatics, Institut Teknologi Bandung, Jl. Ganesha No. 10 Bandung 40132 – Indonesia
[2]Faculty of Mathematics and Natural Sciences, Institut Teknologi Bandung, Jl. Ganesha No. 10 Bandung 40132 – Indonesia
[3]ICT Research Group, Telkom Economics and Business School, Telkom University, Jl. Telekomunikasi No. 1, Bandung

Abstract. A novel method of inserting noise into stream of ciphered text is proposed. The goal of inserting noise is to increase the level of uncertainty, thus making it harder for an attacker to detect data and noise. This form of steganography is implemented using Elliptic Curve Cryptography (ECC). The process of embedding the noise to the message in the encryption process and removing the noise from the message in the decryption process is proposed in this work by modifying ElGamal to allow auto detection of data and noise.

1 Introduction

The objective of coding theory is to resurrect a message from an accidentally noisy environment, meanwhile in cryptography the noise is deliberately superimposed into message so as to make it difficult for an enemy to recover information contained in the message [8]. The addition of noise adds to the level of uncertainty, thus making it harder for an attacker to recover the actual message. Rosing [5] suggested that the ability to hide the same raw data with a lot of random garbage is very useful for cryptography, and the more bits of garbage, the more difficult for adversaries to decrypt it. In a sense, this is a mix of steganography and cryptography. It has been proven that a message can be transmitted securely if it is coded and decoded by a sequence of random bits (key) whose length is equal to that of the message [7]. This is the reason of why inserting noise can add the level of security of a sequence of messages.

There are some issues related to this idea, such as (1) how to differentiate data and noise, and (2) can it be done automatically. The simplest way to differentiate data and noise is to send information about them in a separate stream. This can be done, for example, by sending the list at the beginning or end of a transmission. However, this approach is too cumbersome. There should be a way to do this automatically on the fly. In this paper we modified ElGamal to allow automatic detection, by defining empty message as noise. This will be elaborated in the next section.

Other works related are [1], [3], [6].

2 Theory

In steganography, (stream of) message is embedded into (stream of) cover message. The cover could be an image, text, or even noise. The result is a modified cover message. An attacker or stegoanalyst should not be able to see the actual (embedded) message. If the cover message is an image, the image should not be degraded too much that an attacker could suspect there is a message within the image.

In our case, it is the reverse. We have a stream of ciphered message and noise is added into the stream of ciphered message. However, from a steganography theoretical point of view, we can see this as a stream of noise embedded with ciphered message. The problem would be to make the stego message looks like noise. We will show that our proposed method has this property.

2.1. Crypto + Stego

The process of implementing cryptography and steganography is shown in Figure.1. A message is embbeded as a point $P\ (x, y)$ in an elliptic curve E and then combined with a noise generated by a pseudorandom noise generator that result a sequence of message and noise which in eavesdropper's point of view is just a regular bitstream of data.

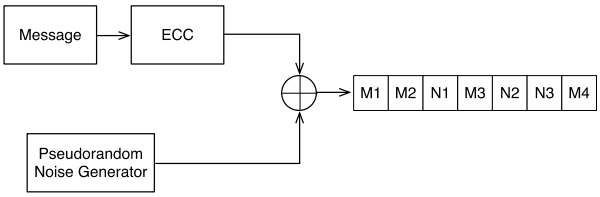

Fig. 1. Crypto + Stego

Figure 2 shows the modified El-Gamal Encryption to enable the process of encryption by embedding the noise to the message. Each letter will be represented as a point in a curve to be multiplied by corresponding value of *k* to be transformed to corresponding value A.

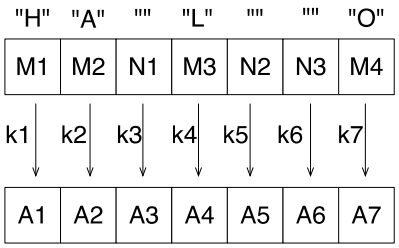

Fig. 2. Modified El-Gamal Encryption

The process of decryption is shown in Figure 3, where the use of El-Gamal process, the removal of the noise is done by identifying the message that contains zero, which is the point that is not on curve of the defined curve equation chosen for the encryption - decryption process.

Figure 3 shows the process of mixing the message and noise that later will be decryption like shown in Figure 4.

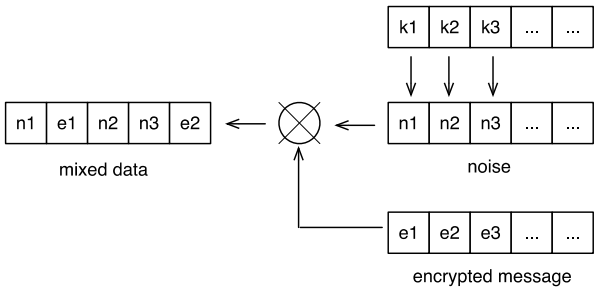

Fig. 3. Mixing message and noise

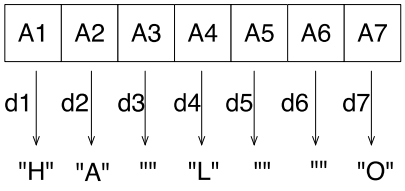

Fig. 4. Modified El-Gamal Decryption

2.2. Randomness and Noise

The basic of our proposed idea is as follows. (See Figure 4.) We have cover message as a stream of noise, n_1, n_2, n_3, n_4, n_5, ... This is going to be mixed with a stream of encrypted message, e_1, e_2, e_3, e_4, e_5, ... Thus, we may have a data stream like $n_1,e_1,n_2,n_3,e_2,e_3,n_4,e_4,...$ To make it difficult for an attacker to realize that there is a message in the data stream, the format of encrypted message and noise should be similar. If the length of encrypted message is 299-bit, then the length of noise is also 299-bit.

In our implementation, the noise is actually an encrypted message generated from null message. The encryption algorithm used is a modified ElGamal encryption scheme, which is going to be elaborated in the next section. The noise is generated from a sequence of random numbers, k_1, k_2, k_3, k_4, k_5, ... That is, $n_i = f(k_i)$, where f is a modified ElGamal function.

A good random number generator algorithm exhibits a stream of numbers that look like noise. We measure that quality of the noise by measuring the randomness of stream of k. An example of randomness testing is available in [4]. The resulting stream of noise, n_i, should have the same property as the random number generator. In fact, since the actual noise is generated from the modified ElGamal function, there is an additional "randomness" produced due to the behavior of encryption algorithm. Thus, the stream of noise should be good.

Our main task is to show that the stream of mixed data, such as $n_1,e_1,n_2,n_3,e_2,e_3,n_4,e_4,....$, still has a noise-like behavior or random. There is a further complication since our stream of data are not 0 and 1 only, but points on a curve. Thus, e_1 is a point with coordinate (x_1,y_1), where x_1 and y_1 are m-bit numbers. Similarly, n_1 is also a point. We have to find a random testing algorithm suitable for this.

2.3 ElGamal

The proposed idea can be implemented as modified El-Gamal encryption scheme. The basic ElGamal encryption scheme [2], which is commonly used in Elliptic Curve Cryptography (ECC), is illustrated below. A plaintext m is represented as a point in the curve M, and then encrypted by adding it to kQ where k is a randomly selected integer and Q is the intended recipient's public key. Both parties agree on F (the finite field), E (equation of elliptic curve), P (base point on E), and n (the order of P).

Basic ElGamal Elliptic Curve Encryption.:
INPUT: domain parameters (F, E, P, n), public key Q, plaintext m.
OUTPUT: Ciphertext (C_1, C_2)
1) Represent the message m as a point of M in E
2) Select random $k \in [1, n-1]$
3) Compute $C_1 = kP$
4) Compute $C_2 = M+kQ$
5) Return (C_1, C_2)

Basic ElGamal Elliptic Curve Decryption.:
INPUT: Domain parameters (F, E, P, n), private key d, ciphertext (C_1, C_2).
OUTPUT: Plaintext m
1) Compute $M = C_2 - dC_1$, and extract m from M
2) Return (m)
Note that $Q = d.P$.

3 Implementation

As noted earlier, issues that we have to tackle are how to differentiate data and noise in the receiving end and can this process be automated. One of the ideas that we come up is to create a different set of data and noise. One possible way to do this is to assign noise as zero message. The rationale is simple. If there is no message, then the stream of data should exhibit noise. Thus, no message means noise. Fortunately, this can be implemented by modifying ElGamal. This way, the process of encryption and decryption will be easy for the authorized parties while it will add high complexity for the unauthorized party, due to the inserted noise.

3.1. Noise Auto-detection

We modify ElGamal encryption and decryption process to make the process of decryption easier on the receiving end. Message is generated by the original ElGamal algorithm, as shown previously. Noise is generated by using $M = 0$, as follow:

$n_1 = (C_1, C_2)$
$C_1 = k_1.P$
$C_2 = k_1.Q$

On the receiving end, the receiver tries to recover the message based on C_1, C_2. If the resulting message M is not zero, it is the actual data. If the resulting message $M = 0$, then it is noise. The recovering process is secure since only the correct receiver has the private key d.
Please note that "0" message is not 0.

3.2 Modified ElGamal

In the implementation, we have to substract C_2 with dC_1 to produce M before making decision whether it is data or noise. Substraction or addition in curve is considered an expensive operation. For noise, C_2 and dC_1, are the same. Checking that they are the same can be done by XOR-ing them, reducing the computation. Thus, noise detection can be further improved by XOR operation only.

Modified ElGamal Elliptic Curve Encryption.:
INPUT: Elliptic curve domain parameters (F, E, P, n), public key Q, plaintext m.
OUTPUT: Ciphertext (C_1, C_2)
1) Represent the message m as a point of M_1 in E
2) Select random $k \in [1, n-1]$
3) Compute $C_1 = kP$
4) Compute $C_2 = M_1 + kQ$
5) Return *for plaintext, $M_1 \neq 0$

Basic ElGamal Elliptic Curve Decryption.:
INPUT: Domain parameters (F, E, P, n), private key d, ciphertext (C_1, C_2).
OUTPUT: Plaintext m
1) Compute $M = C_2 - dC_1$, and extract m from M
2) Return (m)

4 Conclusion

In this paper we proposed the used of steganography to increase the security of encrypted message by mixing message with noise. An attacker would see the sequence of mixed data as noise, unsuspecting that there is encrypted message in it. Trying to brute-force attack the sequence of mixed data will lead to decrypting noise.

To automate the process, noise is produced by using modified ElGamal algorithm. We have shown in Figure 5 that the process of detection can be done easily and automatically on the receiving end.

Our work has a totally different approach compared to existing methods. We are utilizing the mathematical properties of elliptic curve for embedding the noise to the message in a way that it can be recovered easily by the authorized receiver. Taking this advantage we get an efficient algorithm for encryption and decryption, with higher level of security given by the inserted noise.

Acknowledgement

This research is supported by Riset and Inovasi KK ITB.

References

[1] Khalil Challita and Hikmat Farhat. Combining steganography and cryptography: New directions. International Journal on New Computer Architectures and Their Applications (IJNCAA), 1(1):199–208, 2011.
[2] Darrel Hankerson, Alfred Menezes, and Scott Vanstone. Guide to Elliptic Curve Cryptography. Springer-Verlag New York, Inc., 2004.
[3] Nicholas Hopper, Luis von Ahn, and John Langford. Provably secure steganography. IEEE Transactions on Computers, 58(X):1–15, 2009.
[4] George Marsaglia and Wai Wan Tsang. Some difficult-to-pass tests of randomness. Technical report.
[5] Michael Rosing. Implementing Elliptic Curve Cryptography. Manning Publications Co., 1999.
[6] Dipti Kapoor Sarmah and Neha Bajpai. Proposed system for data hiding using cryptography and steganography. Technical report, Departement of Computer Engineering, Maharashtra Academy of Engineering.
[7] A Tomita and O Hirota. Security of classical noise-based cryptography. J. Opt. B: Quantum Semiclass, pages 705–710, 2000.

[8] Dominic Welsh. Codes and Cryptography. Oxford University Press, New York, 1988.

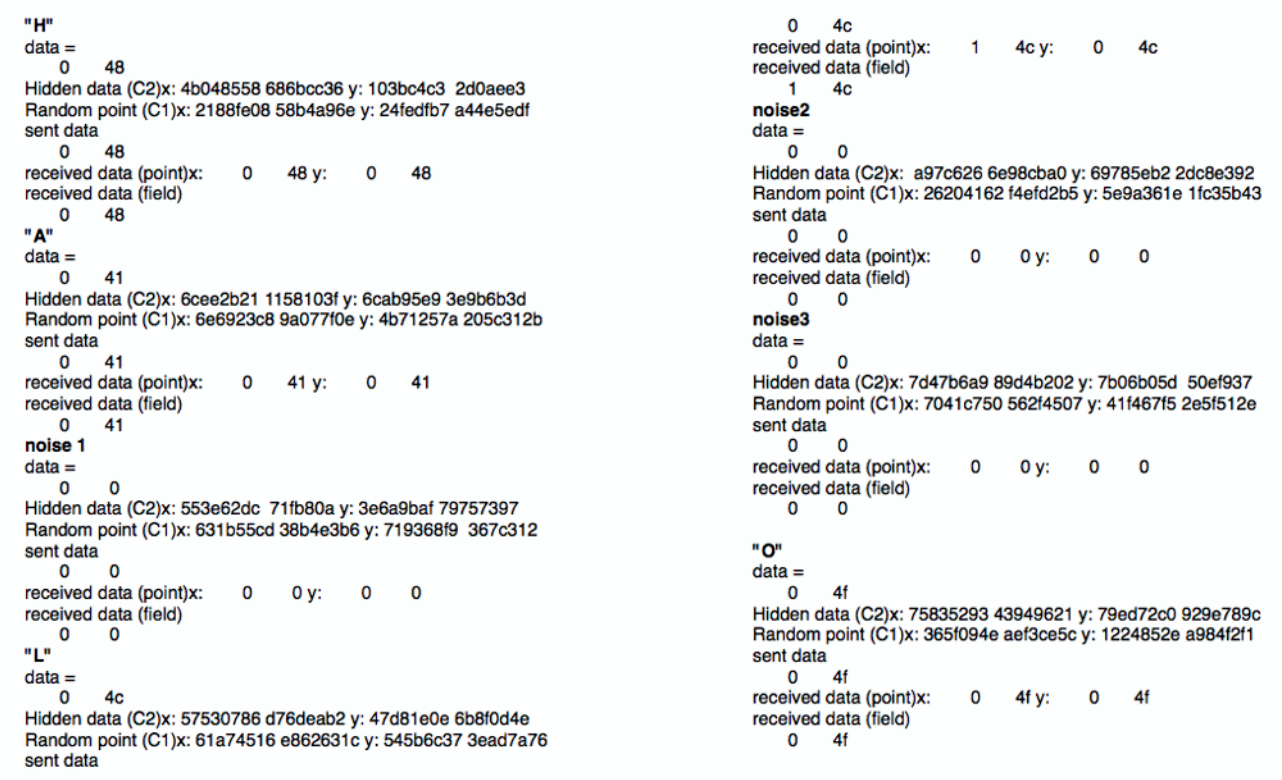

Fig. 5. Stego-ECC Encrypt Decrypt Process