# Constructing a Cryptosystem Algorithm by Applying the Rosenberg-Strong Pairing Function on the Elliptic Curve Cryptography

Sarah.H. Namous[1*], Hamza B.Habib[2] ,and Rifaat Z.Khalaf [3]
[1,2,3] College of Science ,University of Diyala , Diyal, Iraq

**Abstract –**In this paper, we propose a new algorithm to improve the security of the Elliptic Curve Cryptography algorithm, ECC, using the Rosenberg-Strong pairing function. In the proposed algorithm, the Rosenberg-Strong pairing function is applied to the EEC to convert the two ciphertexts (two points on the elliptic curve) to only one ciphertext (one point on the elliptic curve). That is, only one ciphertext is transmitted in the proposed algorithm leading to less transmitting time compared with the classical EEC. The analysis of security illustrates that the proposed algorithm is secure against the attacks of the most common attacking algorithms. Therefore, using the proposed algorithm reduces the transmitting time and provides a high-security level. Thus, the proposed algorithm is secure and efficient, and it provides less transmitting time compared with the classical ECC.

## 1. Introduction

Cryptography, which is an application of Number Theory, is a structure or scheme that converts the plaintext to a ciphertext and transmits it securely via a technological channel [1]. The Rosenberg-Strong Pairing Function in Mathematics encodes the non-negative integer numbers to only one non-negative integer. That means any pair of integers can be represented by a single number. This number can be decoded back and determined which pair it represents by using the Rosenberg-Strong Pairing Function inverse [2]. Elliptic Curve Cryptography, ECC, was proposed independently by Victor Miller and Neal Koblitz in the mid-1980s [3] and [4]. The ECC, which is a public-key encryption algorithm, has advantages over other public-key encryption schemes in terms of security, cost and efficiency of implementation because of the short key length it uses. The public key in the EEC is a random constant (private key) times a point on an elliptic curve, EC [5] and [6]. In this paper, we propose a new cryptosystem algorithm based on applying the Rosenberg-Strong Pairing Function on the ECC. In the proposed algorithm, the sender encrypts the plaintext by the ECC and then uses the Rosenberg-Strong Pairing Function to convert each ciphertext (each point) to a single number. That is, the two ciphertexts (the two points) are converted to a single ciphertext (one point), and then it is sent to the recipient.

* Corresponding author: mathmathsara99@gmail.com

To get the plaintext back, the recipient first uses the inverse of the Rosenberg-Strong Pairing Function and then decrypts it using the ECC decryption process. Therefore, the application of the Rosenberg-Strong Pairing Function on the EEC reduces the data transmitting time. Moreover, the security analysis shows that the proposed algorithm is secure against the most attacking algorithms compared to the classical EEC algorithm. Therefore, the proposed algorithm is secure and fast in transmitting data via digital communication channels.

The rest of this paper is organized as follows. In Section 2, the Rosenberg-Strong Pairing Function is described. In Section 3, the ECC is discussed. In Section 4, the proposed algorithm is presented. In Section 5, the Security analysis of the proposed algorithm is discussed. Finally, the conclusion is provided in Section 6.

## 2. Rosenberg-Strong Pairing Function

The Rosenberg-Strong pairing function of the nonnegative integers, see [2] and [7-11], is given by

$$\delta = (max(x,y))^2 + max(x,y) + (x - y). \quad (1)$$

where $\max(x,y)$ refers to the point's shell number. The Rosenberg-Strong pairing function is represented in Figure 1, such that, the appearance of the points in a consecutive enumeration $\delta^{-1}: \mathbb{N} \to \mathbb{N}^2$ are connected by arrows if and only if they have the same quantity $\max(x,y)$ in common, see [7] and [11].

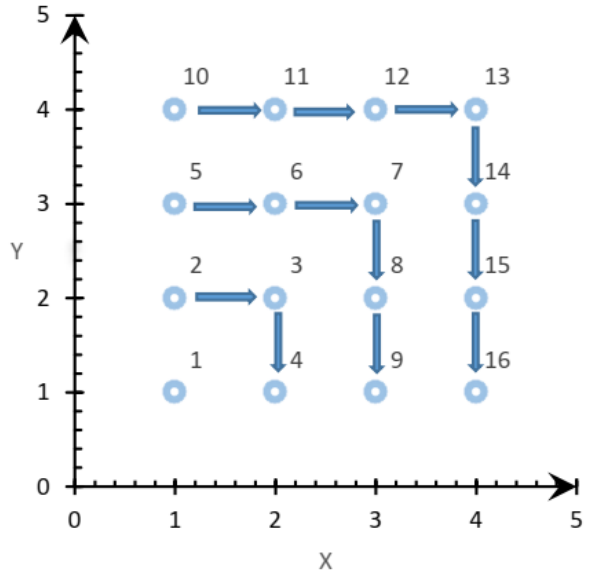


**Fig1**: The Rosenberg-Strong pairing function

.

The Rosenberg-Strong pairing inverse function is given by

$$\delta^{-1} = \begin{cases} (\delta - m^2, m) & \text{if } \delta - m^2 < m, \\ (m, m^2 + 2m - \delta) & \text{otherwise.} \end{cases} \quad (2)$$

where $m = \lfloor \sqrt{\delta} \rfloor$.

**Example 2.1:** Again consider the point $(x, y) = (3,8)$, by Equation (1) we have

$$\begin{aligned} \delta &= (\max(x,y))^2 + \max(x,y) + (x - y) \\ &= (\max(3,8))^2 + \max(3,8) + (3 - 8) \\ \delta &= 67 \end{aligned}$$

To get back the initial numbers, we have,

$$m = \lfloor \sqrt{\delta} \rfloor$$

$$m = \lfloor \sqrt{67} \rfloor$$

$$m = 8$$

Since $\delta - m^2 < m$, then by Eq. (2)

$$\delta^{-1} = (67 - 8^2, 8) = (3, 8).$$

Then, $(x, y) = (3,8)$.

## 3. The Elliptic Curve Cryptography

**Definition 3.1:** [12-13] An elliptic curve, EC, over a finite field $F_p$ is given by

$$E_p(a, b): y^2 \equiv (x^3 + ax + b) \pmod{p},$$

where a , b are constant and $4a^3 + 27b^2 \not\equiv 0 \pmod{p}$.

**Theorem 3.1:** See [1] Let $R = (x_1, y_1)$ and $W = (x_2, y_2)$ be two points on the $E_p(a, b)$. Then, the addition and doubling of R and W are given by,

$$R + W = \begin{cases} E_\infty & \text{if } x_1 = x_2 \text{ , } y_1 = -y_2, \\ (x_3, y_3) & \text{otherwise.} \end{cases}$$

Where $E_\infty$ is the point at infinity,

$$x_3 \equiv (\lambda^2 - x_1 - x_2) \pmod{p},$$

$$y_3 \equiv (\lambda(x_1 - x_3) - y_1) \pmod{p}.$$

and

$$\lambda \equiv \begin{cases} \left(\frac{3x_1^2 + a}{2y_1}\right) (\mathrm{mod}\, p) & \text{if } R = W; \\ \left(\frac{y_2 - y_1}{x_2 - x_1}\right) (\mathrm{mod}\, p) & \text{if } R \neq W. \end{cases}$$

**Theorem 3.2:** [1] Let $E_p(a, b): y^2 \equiv (x^3 + ax + b) \pmod p$, $\left(\frac{x^3+ax+b}{p}\right)$ is the Legendre Symbol, be an elliptic curve over $F_p$. The number of points on $E_p(a, b)$ is given by

$$\left|E_p(a, b)\right| = 1 + p \sum_{x \in F_p} \left(\frac{x^3 + ax + b}{p}\right) = 1 + p + \epsilon$$

**Theorem 3.3: (Hasse Theorem)** [1], $|\epsilon| \leq 2\sqrt{p}$.

Thus, we can have

$$1 + p - 2\sqrt{p} \leq \left|E_p(a, b)\right| \leq 1 + p + 2\sqrt{p}.$$

The EEC algorithm, see [12-14], is as follows: Alice the sender, and Bob the receiver agree on an elliptic curve $E_p(a, b): y^2 \equiv (x^3 + ax + b) \pmod p$.

**The Key Generation process**

Alice and Bob decide to employ an elliptic curve $E_p(a, b)$ to communicate the message, where p is prime and choose the point G on $E_p(a, b)$ .

Bob selects a random positive integer α which is the private key and calculates the public key $A = \alpha G$.

**The Encryption process**

If Alice wants to send the message M to Bob, Alice uses a random number γ. She calculates

$$E_1 = \gamma G \tag{3}$$

$$E_2 = M + \gamma B. \tag{4}$$

Alice send $E_1$ and $E_2$ to Bob.

**The Decryption process**

Bob receives $E_1$ and $E_2$, then he decrypts the message as follows,

$$M = E_2 - \beta E_1. \tag{5}$$

## 4. The Proposed Algorithm

In this algorithm, two interesting topics in Mathematics, which are the Rosenberg-Strong Pairing Function and the EEC, are combined to construct a cryptosystem algorithm. The Rosenberg-Strong Pairing Function is used to encode

a pair of non-negative integers to one non-negative integer, and the EEC is used to encode the plaintext to a ciphertext based on mathematical steps. The proposed algorithm is presented below.

**The Agreed Algorithm's Construction**

Let the initial point on the elliptic curve $E_p(a, b)$ is $R = (x_1, y_1)$, then based on Theorem 3.1 a positive integer k times R means R is added k times. That is,

$$kR = \overbrace{R + R + \cdots + R}^{k}$$

Now, the first letter "A" = R, the second letter "B" = 2R, and so on the rest of the alphabet and then numbers. The agreed algorithm consists of aa of the points on the $E_p(a, b)$ and $E_\infty$.

**Encryption Process**

Alice transforms the plaintext into corresponding points on $E_p(a, b)$ based on the agreed algorithm. After selecting a random positive integer γ, then Alice does the calculation $E_1 = (x_1, x_2)$ and $E_2 = (x_1, x_2)$ by Eq. (3) and Eq. (4) respectively.

Alice applies the Rosenberg-Strong Pairing Function for the points $E_1$ and $E_2$ as shown in Eq. (6) and Eq. (7) respectively.

$$\delta_{E_1} = (\max(x_1, y_1))^2 + \max(x_1, y_1) + (x_1 - y_1), \tag{6}$$

and

$$\delta_{E_2} = (\max(x_2, y_2))^2 + \max(x_2, y_2) + (x_2 - y_2). \tag{7}$$

Then, $(\delta_{E_1}, \delta_{E_2})$ is the ciphertext, and it is sent to Bob.

**The Decryption Process**

After receiving $(\delta_{E_1}, \delta_{E_2})$, Bob applies the Rosenberg-Strong pairing inverse function, as shown in Eq. (8) and Eq. (9).

$$\delta_{E_1}^{-1} = \begin{cases} (\delta_{E_1} - m_1^2,\ m_1) & \text{if } \delta_{E_1} - m_1^2 < m_1, \\ (m_1, m_1^2 + 2m_1 - \delta_{E_1}) & \text{otherwise.} \end{cases} \tag{8}$$

and

$$\delta_{E_2}^{-1} = \begin{cases} (\delta_{E_2} - m_2^2,\ m_2) & \text{if } \delta_{E_2} - m_2^2 < m_2, \\ (m_2, m_2^2 + 2m_2 - \delta_{E_2}) & \text{otherwise.} \end{cases} \tag{9}$$

where $m_1 = \lfloor \sqrt{\delta_{E_1}} \rfloor$ and $m_2 = \lfloor \sqrt{\delta_{E_2}} \rfloor$.

Now, Bob performs the decryption process as follows:

$$M = \delta_{E_2}^{-1} - \beta\delta_{E_1}^{-1} \tag{10}$$

**Example 4.1.** For simplicity consider the elliptic curve $E_{29}(8,6)$: $y^2 \equiv x^3 + 8x + 6 \pmod{29}$, and the initial point on the $E_{29}(8,6)$ is $(0,8)$. The agreed algorithm of $E_{29}(8,6)$ is given the Table 1. In this example, the Maple programming language is employed for the calculations.

**Table 1**: The agreed algorithm of $E_{29}(8,6)$.

| Letters | Point | Letters | Point | Number | Point |
|---|---|---|---|---|---|
| A | (0,8) | N | (10,10) | 1 | (7,17) |
| B | (22,10) | O | (26,10) | 2 | (13,25) |
| C | (13,4) | P | (26,19) | 3 | (22,19) |
| D | (7,12) | Q | (10,19) | 4 | (0,21) |
| E | (17,3) | R | (6,26) | | $E_\infty$ |
| F | (16,24) | S | (3,12) | | |
| G | (14,7) | T | (2,28) | | |
| H | (19,12) | U | (11,27) | | |
| I | (9,13) | V | (9,16) | | |
| J | (11,2) | W | (19,17) | | |
| K | (2,1) | X | (14,22) | | |
| L | (3,17) | Y | (16,5) | | |
| M | (6,3) | Z | (17,26) | | |

Also, suppose Bob selects the point $G = (11,2)$ and selects the private key $\beta = 8$. Then, Bob calculates the public key as

$$B = \beta G = 8(11,2) = (6,26).$$

Assume Alice wishes to send Bob the message M = "CIPHER", then Alice converts M to suitable points based on Table. That is, "C" = (13, 4), and Alice chooses $\gamma = 3$. Then, by Eq. (3) and Eq. (4)

$$E_1 = \gamma G = 3(11,2) = (0,21),$$

$$E_2 = C + \gamma B$$

$$= (13,4) + 3(6,26)$$

$$E_2 = (13,4) + (19,17) = (17,26).$$

By Eq. (6) and Eq. (7) respectively, we have

$$\delta_{E_1} = (\max(0,21))^2 + \max(0,21) + 21 = 441.$$

and

$$\delta_{E_2} = (\max(17,26))^2 + \max(17,26) + 17 - 26 = 693.$$

Then, Alice sends $(441, 693)$ to Bob.

Since $m_1 = \lfloor\sqrt{441}\rfloor = 21$, and $\delta_{E_1} - m_1^2 < m_1$, then by Eq. (8) we have

$$\delta_{E_1}^{-1} = (441 - 21^2, 21) = (0, 21).$$

Also, since $m_2 = \lfloor\sqrt{693}\rfloor = 26$ and $\delta_{E_2} - m_2^2 < m_2$, then by Eq. (9) we have

$$\delta_{E_2}^{-1} = (693 - 26^2, 26) = (17,26).$$

Now, by Eq. (10)

$$M = (17,26) - 8(0,21)$$

$$= (17,26) - (19,17)$$

$$M = (17,26) + (19,12) = (13,4)$$

In the same way, the rest of the message $M$ is encrypted and decrypted, see Table 2.

**Table 2:** The message $M$ is encrypted and decrypted by the proposed algorithm.

| **M** | The Points | Encryption by Alice | | | Rosebery Pairing Function | Decryption by Bob | | | | | **M** |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | $\gamma$ | $E_1$ | $E_2$ | $(\delta_{E_1}, \delta_{E_2})$ | $m_1$ | $\delta_{E_1}^{-1}$ | $m_2$ | $\delta_{E_2}^{-1}$ | M | |
| C | (13,4) | 3 | (0,21) | (17,26) | (441,693) | 21 | (0,21) | 26 | (17,26) | (13,4) | C |
| I | (9,13) | 7 | (19,12) | (2,1) | (387,7) | 19 | (19,12) | 2 | (2,1) | (9,13) | I |
| P | (26,19) | 10 | (14,7) | (11,2) | (217,141) | 14 | (14,7) | 11 | (11,2) | (26,19) | P |
| H | (19,12) | 11 | (10,19) | (2,28) | (371,786) | 19 | (10,19) | 28 | (2,28) | (19,12) | H |
| E | (17,3) | 9 | (13,25) | (3,17) | (638,292) | 25 | (13,25) | 17 | (3,17) | (17,3) | E |
| R | (6,26) | 5 | (3,12) | (26,10) | (147,718) | 12 | (3,12) | 26 | (26,10) | (6,26) | R |

For the classical ECC algorithm, the message M is encrypted and decrypted as shown in Table 3

**Table 3:** The message $M$ is encrypted and decrypted by the classical ECC algorithm.

| **M** | The Point | Encryption by Alice | | | Decryption by Bob |
|---|---|---|---|---|---|
| | | $\gamma$ | $E_1$ | $E_2$ | M |
| C | (13,4) | 3 | (0,21) | (17,26) | (13,4) |
| I | (9,13) | 7 | (19,12) | (2,1) | (9,13) |
| P | (26,19) | 10 | (14,7) | (11,2) | (26,19) |
| H | (19,12) | 11 | (10,19) | (2,28) | (19,12) |
| E | (17,3) | 9 | (13,25) | (3,17) | (17,3) |
| R | (6,26) | 5 | (3,12) | (26,10) | (6,26) |

## 5. The Security Analysis

### i) Attacking the Private Key

From Table 2 and Table 3 in Section 4, we can notice the simplicity of the steps that are performed in the classical algorithm compared to the proposed algorithm. These steps make the classical algorithm available for the attacking algorithms. The most common attacking methods of the Discrete Logarithm Problem (DLP), are Baby-Step Giant-Step, Pollard's Lambda, Pollard's Roh, and Index Calculus.

In this section, we will discuss Pollard's Lambda and Pollard's Roh methods in detail.

The strength of the ECC depends on the DLP. The Pollard's Roh method would find the private key in time $\sqrt{N}$, where N is the cyclic order of the EC with the generator point G. Pollard's Lambda method on the other hand is used to find the equivalent and the private key in the time $\sqrt{N}$. If both methods are implemented at the same time, then the time of finding the private key would be reduced. Also, both methods depend on the probability; however, they do not guarantee completion in a fixed time. For example, if

$N = 5432176432987656321485732941236201130215376529038562374201$, then, the number of the steps is $\sqrt{N} = 7.92282 \times 10^{28}$.

Suppose that each step needs a time of 0.0000001 part of the second, then

$$\left(\frac{\text{The number of steps}}{\frac{\text{The time required for each step}}{3600}}\right) \times 86400 = \left(\frac{7.92282 \times 10^{28}}{\frac{0.0000001}{3600}}\right) \times 86400$$

$$= 22007.8 \times 10^{28} \times 86400 = 190147 \times 10^{28} = 1.90147 \times 10^{23}.$$

That is, it needs $1.90147 \times 10^{23}$ days to find the private key. Therefore, it means that there is no benefit to using these methods because they need hundreds of years to get the private key, and at that time the secret data would be useless.

#### ii) The Ciphertext-Only Attack

From the transmitted ciphertext, $E_1$ and $E_2$, the attacker can guess the value of the used p in the equation of $E_p(a, b)$, from the results of $(x, y)$. In the proposed algorithm, the points are converted to new points by using the pairing function, such that, the range of the points became $(\max(x, y))^2 + \max(x, y) + (x - y)$. Therefore, it is difficult for the attacker to guess the value of the used p in the equation of $E_p(a, b)$ that makes such an attack a useless attack.

#### iii) The Known-Plaintext Attack

Suppose that the attacker could get the plaintexts $(x_1, y_1), (x_2, y_2)$, and $(x_3, y_3)$ and the corresponding ciphertexts, then he still cannot get the plaintext of the whole message because of the randomness of the used key. For example, if the plaintext is "aaa", then the corresponding ciphertext is "xrp". That is, the known- plaintext attack is useless.

#### iv) The Brute Force Attack

Any cryptosystem algorithm that is looking for security would use a large private key. If the used private key is large, then this type of attack becomes useless.

## 6. Conclusion

We propose a new cryptosystem algorithm based on the use of ECC and the Rosenberg-strong pairing function. In the classic EEC, two ciphertexts are sent, $E_1$ and $E_2$, which are two points on the EC. In the proposed algorithm, these two points are paired to only one point by using the Rosenberg-strong pairing function. That means, less transmitting time in the proposed algorithm compared to the classical EEC. Moreover, the security analysis shows that the proposed algorithm provides more security against the most attacking algorithms compared to the classical EEC. Therefore, the proposed algorithm is fast and secure for transmitting data.

## References

1 Yan, S.Y., 2013. Computational number theory and modern cryptography. John Wiley & Sons.
2 Szudzik, M., 2006. An elegant pairing function. In Wolfram Research (ed.) Special NKS 2006 Wolfram Science Conference (pp. 1-12).
3 Miller, V.S., 1985, August. Use of elliptic curves in cryptography. In Conference on the theory and application of cryptographic techniques (pp. 417-426). Berlin, Heidelberg: Springer Berlin Heidelberg.
4 Koblitz, N., 1987. Elliptic curve cryptosystems. Mathematics of computation, 48(177), pp.203-209.
5 Zhang, P., Li, Y. and Chi, H., 2022. An elliptic curve signcryption scheme and its application. Wireless Communications and Mobile Computing, 2022.

6 Bao, J., 2022, April. Research on the security of elliptic curve cryptography. In 2022 7th International Conference on Social Sciences and Economic Development (ICSSED 2022) (pp. 984-988). Atlantis Press.

7 Szudzik, M.P., 2017. The rosenberg-strong pairing function. arXiv preprint arXiv:1706.04129.

8 Rosenberg, A.L. and Strong, H.R., 1972. Addressing arrays by shells. IBM Technical Disclosure Bulletin, 14(10), pp.3026-3028.

9 Rosenberg, A.L., 1974. Allocating storage for extendible arrays. Journal of the ACM (JACM), 21(4), pp.652-670.

10 Goubin, L., Guilley, S., Fournier, J., Jauvart, D., Moreau, M., Rauzy, P. and Rondepierre, F., 2017. Nadia El Mrabet. Guide to Pairing-Based Cryptography.

11 Szudzik, M.P., 2018. Binary Proportional Pairing Functions. arXiv preprint arXiv:1809.06876.

12 Hankerson, D. and Menezes, A., 2021. Elliptic curve cryptography. In Encyclopedia of Cryptography, Security and Privacy (pp. 1-2). Berlin, Heidelberg: Springer Berlin Heidelberg.

13 Koblitz, N., Menezes, A. and Vanstone, S., 2000. The state of elliptic curve cryptography. Designs, codes and cryptography, 19, pp.173-193.

14 Natanael, D. and Suryani, D., 2018. Text encryption in android chat applications using elliptical curve cryptography (ECC). Procedia Computer Science, 135, pp.283-291.