BAREKENG: Journal of Mathematics and Its Application
December 2022 Volume 16 Issue 4 Page 1159–1166
P-ISSN: 1978-7227 E-ISSN: 2615-3017
https://doi.org/10.30598/barekengvol16iss4pp1159-1166

MODIFICATION OF POLLARD RHO ALGORITHM USING NEGATION MAPPING

Sa'aadah Sajjana Carita [1*], **Herman Kabetta**[2]

[1,2] *Cryptographic Engineering Study Program, Cryptography Department, National Cyber and Crypto Polytechnic Jl. Raya H. Usa, Kab. Bogor, 16120, Indonesia*

Corresponding author's e-mail: [1*] *sajjana.carita@bssn.go.id*

Abstract. *El Gamal encryption was introduced in 1985 and is still commonly used today. Its hardness is based on a discrete logarithm problem defined over the finite abelian cyclic group. The group chosen in the original paper was $\mathbb{Z}_p$, but later it was proven that using the group of Elliptic Curve points could significantly reduce the key size required. The modified El Gamal encryption is dubbed its analog version. This analog encryption bases its hardness on Elliptic Curve Discrete Logarithm Problem (ECDLP). One of the fastest attacks in cracking ECDLP is the Pollard Rho algorithm, with the expected number of iterations $\sqrt{\frac{\pi n}{2}}$, where n is the number of points in the curve. This paper proposes a modification of the Pollard Rho algorithm using a negation map. The experiment was done in El Gamal analog encryption of elliptic curve defined over the field $\mathbb{Z}_p$, with different values of small digit p. The modification was expected to speed up the algorithm by $\sqrt{2} \approx 1.4$ times. The average of speed up in the experiment was 1.9 times.*

Keywords: *El gamal encryption, elliptic curve cryptography, negation map, pollard rho algorithm.*

Article info:

Submitted: 28th April 2022 *Accepted: 9th October 2022*

How to cite this article:
S. S. Carita and H. Kabetta, "MODIFICATION OF POLLARD RHO ALGORITHM USING NEGATION MAPPING", *BAREKENG: J. Math. & App.*, vol. 16, iss. 4, pp. 1159-1166, Dec., 2022

1. INTRODUCTION

Cryptography studies mathematical techniques for transmitting secret messages through insecure channels [1]. In cryptography, the message to be sent is called plaintext. The encryption process is carried out with a key to produce ciphertext, which can only be returned to the original message through decryption. An encryption and decryption algorithm, along with all possible plaintext, ciphertext, and keys, is called a cryptosystem [2].

Cryptography can be categorized into two, namely symmetric and asymmetric cryptography. Symmetric cryptography uses only one secret key for decryption and encryption. In contrast, asymmetric cryptography uses a public key for encryption and a secret key for decryption. In practice, asymmetric cryptography, which is generally slower than symmetric cryptography, is used to distribute symmetric cryptographic keys. [3].

The asymmetric cryptographic scheme requirement is computationally very difficult to decrypt without knowing the secret key. The difficulty of this process can be guaranteed by selecting mathematical problems that have proven difficult to solve [4]. One of the commonly used asymmetric encryption algorithms is El Gamal. Taher El Gamal introduced this algorithm around 1985 [5]. The frequently used El Gamal cryptosystem is defined on the $\mathbb{Z}_p$ field, where p is prime and is based on the Diffie Hellman key exchange algorithm. [6].

The El Gamal cryptosystem can not only be applied to the $\mathbb{Z}_p$ field, but also to any cyclic commutative group [7]. We will explain later about Elliptical Curves and that the points on these curves form cyclic groups. The El Gamal algorithm can also be applied to elliptic curves, called an El Gamal encryption analog [8]. This analog algorithm has the advantage that the required key size is smaller than the initial algorithm with the same level of security. For example, the El Gamal cryptosystem with public and secret key sizes of 3072-bit and 256-bit, respectively, is equivalent to El Gamal's analog algorithm with a key size of 163-bits [9].

The difficulty of attacking the El Gamal encryption analog lies in the Elliptic Curve Discrete Logarithm Problem (ECDLP), namely the difficulty of getting the secret key value $k \in \mathbb{Z}^+$ if the public key is known, at points P and $Q = kP$ on the curve [10]. However, to prove the security of our cryptosystem, we must try to crack the system. One of the fastest algorithms to attack ECDLP is the Pollard Rho algorithm [11], which will be explained in Chapter 2. This research modified the Pollard Rho algorithm by applying negation mapping to reduce iteration steps. It has the benefit of potentially reducing computational resources to carry out attacks against analogs of the El Gamal algorithm.

2. RESEARCH METHOD

This chapter discusses elliptic curve cryptography, analogs of the El Gamal algorithm, Pollard Rho algorithm, negation mapping, and modifications to the Pollard Rho algorithm using negation mapping.

2.1 Elliptical Curve Cryptography

The explanation of this elliptic curve is processed from [12] unless otherwise stated.

Definition 1

For prime numbers $p \neq 2,3$, elliptic curve $E(\mathbb{Z}_p)$ is the set of points $(x, y) \in \mathbb{Z}_p \times \mathbb{Z}_p$ that fulfills

$$y^2 = x^3 + Ax + B, \quad (1)$$

with $A, B \in \mathbb{Z}_p$ and $4A^3 + 27B^2 \neq 0$, that is added "dot at infinity" $\mathcal{O}$ an identity to the sum of points.

The following explains the addition of points on an elliptic curve. The image in this section is obtained from [13].

In Equation (1), there are 1 or 3 possible real values for x, so there are two possible sketches of the elliptic curve graph, which is given in Figure 1.

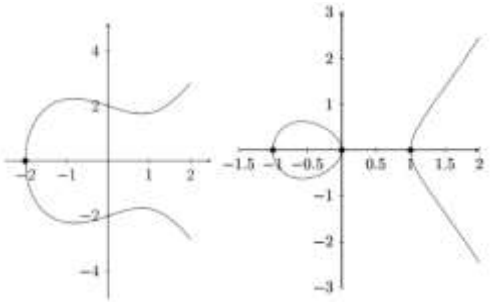

Figure 1. Elliptical curve graph sketch

Addition of two points $P = (x_1, y_1)$ and $Q = (x_2, y_2)$ generate point $P + Q = (x_3, y_3)$ with a calculation that is divisible by 4 cases.

Case 1. $x_1 \neq x_2$ (illustrated in Figure 2)

Draw a line through P and Q and intersect the curve E back at point R. Point $P + Q$ is a reflection of R to the x-axis. The explicit formula is given in Equation (2).

$$x_3 = m^2 - (x_1 + x_2),\ y_3 = m(x_3 - x_1) + y_1, \text{ with } m = \frac{y_2 - y_1}{x_2 - x_1} \tag{2}$$

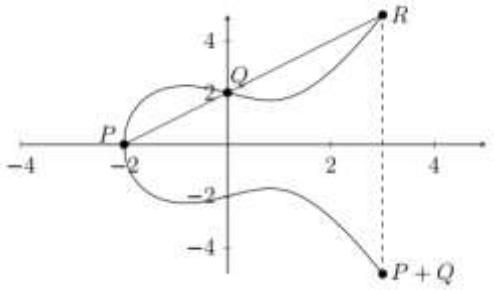

Figure 2. Case 1 point sum

Case 2. $x_1 = x_2, y_1 = -y_2$ (illustrated in Figure 3)

In the case of P_1 and P_2, they are negated by each other. The straight line joining the two points is vertical, so

$$P_1 + P_2 = \mathcal{O} \tag{3}$$

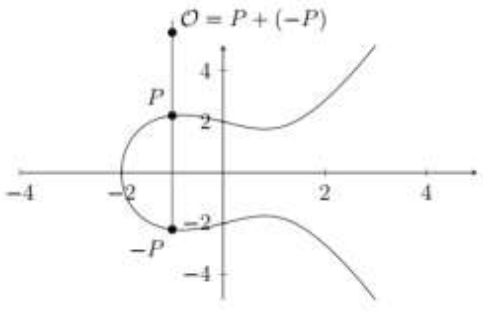

Figure 3. Case 2 point sum

Case 3. $x_1 = x_2, y_1 = y_2$ with $y_1 \neq 0$ (illustrated in Figure 4)

In this case, the point $P_1 = P_2$. The line drawn is a tangent at the point P_1 and $P_1 + P_2 = 2P_1 = (x_3, y_3)$ can be obtained by Equation (4).

$$x_3 = m^2 - 2x_1, y_3 = -m(x_3 - x_1) - y_1, \text{ with } m = \frac{3x_1^2 + A}{2y_1} \quad (4)$$

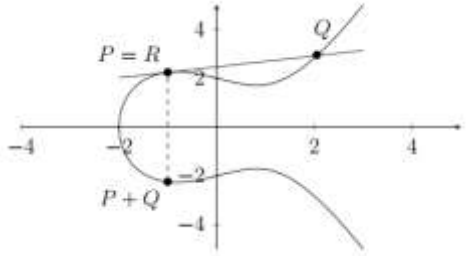

Figure 4. Case 3 point addition

Case 4. $x_1 = x_2, y_1 = y_2 = 0$ (illustrated in Figure 5)

In this case, the line drawn is a tangent at point P_1 and is a vertical line, so as in Case 2,

$$P_1 + P_2 = 2P_1 = \mathcal{O} \quad (5)$$

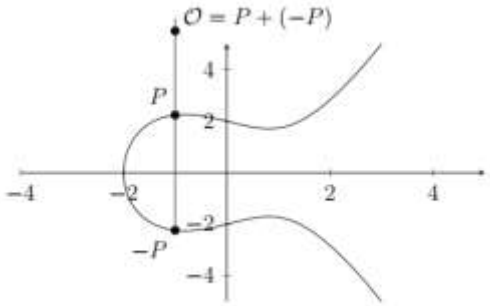

Figure 5. Case 4 point sum

From Case 1 to 4, it can be shown that the points on the elliptic curve form a commutative group for summation. Evidence of this statement is found in [12]. Here is defined the product of a non-negative integer by a point on the elliptic curve.

Definition 2

Let P be a point on the elliptic curve E, then for every integer k, the point kP is defined in Equation (6).

$$\underbrace{P + P + \cdots P}_{k} \quad (6)$$

Given any point P and point $Q = kP$, it is very difficult to find the value of k. This mathematical problem is called the Elliptic Curve Discrete Logarithm Problem (ECDLP). It guarantees the difficulty of solving the El Gamal encryption analog. The following describes the analog algorithm:

2.1 El Gamal Analog Encryption Algorithm

This algorithm is reprocessed from [14]. Suppose Alice is to send a message to Bob.

1. Alice and Bob agree on an elliptic curve $\boldsymbol{E}: \boldsymbol{y^2 = x^3 + Ax + B}$ and a large prime number to form the field $\mathbb{Z}_{\boldsymbol{p}}$. They also have one point $\boldsymbol{P} \in \boldsymbol{E}$.
2. Bob and Alice choose arbitrary integers $\boldsymbol{k_1}$ and $\boldsymbol{k_2}$, respectively. These two numbers are the secret key.
3. Bob and Alice each count $\boldsymbol{Q = k_1 P}$ and $\boldsymbol{R = k_2 P}$, which is then used as the public key.
4. Alice turns her message into points on the curve $\boldsymbol{E}$ (in a way that is not discussed in this paper) then encrypt each point, for example, $\boldsymbol{M} \in \boldsymbol{E}$, become $\boldsymbol{S = M + k_2 Q}$ (using Bob's public key Q) then sends it.

5. Bob decrypts $\boldsymbol{S}$ to $\boldsymbol{S - k_1 R = M + k_2 Q - k_1 R = M + k_1 k_2 P - k_1 k_2 P = M}$. This process requires Alice's public key $\boldsymbol{R}$ and Bob's secret key $\boldsymbol{k_1}$.

As mentioned earlier, the difficulty of breaking the analog of El Gamal encryption lies in ECDLP is obtaining the secret key value k_1 if the public keys P and Q are known. However, to prove the security of our cryptosystem, we must try to crack the system.

One commonly used algorithm to attack ECDLP is the Pollard Rho algorithm, which will be explained in Chapter 2. This study will modify the Pollard Rho algorithm by applying a negation mapping to reduce the iteration step.

2.2 Pollard Rho's Algorithm

For example, $E(\mathbb{Z}_p)$ elliptic curve over the field $\mathbb{Z}_p$ so that $|E| = n$, and P and Q point on E so that $Q = kP$. The purpose of the Pollard Rho algorithm is to get k [14].

1. Use a hash function to partition E into three sets S_1, S_2, S_3 with the same relative size, and $O \notin S_2$.
2. Define *random walk:*
$$R_{i+1} = f(R_i) = \begin{cases} Q + R_i, & R_i \in S_1 \\ 2R_i, & R_i \in S_2 \\ P + R_i, & R_i \in S_3 \end{cases} \quad (7)$$
3. For example, $R_i = a_i P + b_i Q$, then from the equation (7),
$$a_{i+1} = \begin{cases} a_i, & R_i \in S_1 \\ 2a_i, & R_i \in S_2, \\ a_i + 1, & R_i \in S_3 \end{cases}$$
$$b_{i+1} = \begin{cases} b_i + 1, & R_i \in S_1 \\ 2b_i, & R_i \in S_2. \\ b_i, & R_i \in S_3 \end{cases} \quad (8)$$
Start with $R_0 = P, a_0 = 1, b_0 = 0$ and pair figure (R_i, R_{2i}) using equation (8) to find similarities $R_m = R_{2m}$ for a m.
After finding the similarities, then we get $R_m = a_m P + b_m Q$ and $R_{2m} = a_{2m} P + b_{2m} Q$, so that
$$k = \frac{a_{2m} - a_m}{b_m - b_{2m}} \ (mod\ n) \quad (9)$$

Based on [15], the estimated number of iterations until a similarity is found is $\sqrt{\frac{\pi n}{2}}$. This study modified the Pollard Rho algorithm to reduce the number of iterations by applying a negation mapping.

2.3 Pollard Rho Algorithm Mapping and Modification

Definition 3

For example, $P = (x, y) \in E$. A negation mapping is a mapping $\psi(P) = -P = (x, -y)$.

When viewed as a subgroup automorphism constructed by P, i.e. P〉 , this mapping has an order of 2, because $\psi^2(P) = \psi(\psi(P)) = \psi(-P) = -(-P) = P$, back to the starting point.

To speed up the algorithm, random walks are defined in subgroups $\langle P \rangle$ instead of at point P. If the order of the automorphism is u, then the expected number of iterations is $\sqrt{\frac{\pi m}{2u}}$, so the expectation for the negation mapping is $\frac{\sqrt{\pi m}}{2}$, or around $\sqrt{2}$ faster than the initial algorithm [16]. The following is the proposed modification algorithm.

1. Choose any prime number p.
2. Choose a value $A, B \in \mathbb{Z}_p$ with $4A^3 + 27B^2 \neq 0$
3. Find all points on the elliptic curve $E(\mathbb{Z}_p)$, namely, all $(x, y) \in \mathbb{Z}_p \times \mathbb{Z}_p$ which fulfill Equation (1).

4. Calculate $n = |E|$, and check if n is prime. If not, repeat step 2. This prime value ensures that inverse multiplication mod n is always possible. In addition, the prime value of n avoids vulnerability to Pohlig-Hellman and MOV attacks [17]–[19].
5. Choose a point P at random from the points found in step 3. Choose any natural number k and calculate $Q = kP$.
6. Use the following hash function for Step 1 algorithm 3.2:
   a. Calculate value ϕ namely approximation $\frac{\sqrt{5}-1}{2}$ up to 50 digits after comma [20].
   b. Define function $v^*: E \to [0,1)$ where
   $$v^*\big(P=(x,y)\big) = \begin{cases} \phi y - \lfloor \phi y \rfloor, & P \neq \mathcal{O} \\ 0, & P = \mathcal{O} \end{cases} \quad (10)$$
   c. Define a hash function $v: E \to \{1, \cdots, r\}$ from the equation (10):
   $$v(P) = \lfloor r \cdot v^*(P) \rfloor + 1 \quad (11)$$
   d. Partition S_1, S_2, S_3 from the equation (11):
   $$S_1\{P \in E | v(P) = 1\}$$
   $$S_2\{P \in E | v(P) = 2\} \quad (12)$$
   $$S_3\{P \in E | v(P) = 3\}$$
7. Suppose $R_i = a_i P + b_i Q$, then from (7),
   $$a_{i+1} = \begin{cases} a_i, & R_i \in S_1 \\ 2a_i, & R_i \in S_2 \\ a_i + 1, & R_i \in S_3 \end{cases}$$
   and (13)
   $$b_{i+1} = \begin{cases} b_i + 1, & R_i \in S_1 \\ 2b_i, & R_i \in S_2 \\ b_i, & R_i \in S_3 \end{cases}$$
8. Start with $R_0 = P, a_0 = 1, b_0 = 0$ and pair figure (R_i, R_{2i}) until the similarities are found $R_m = R_{2m}$ or $R_m = -R_{2m}$ for a m.
9. Suppose
   $$R_m = a_m P + b_m Q$$
   $$R_{2m} = a_{2m} P + b_{2m} Q \quad (14)$$
   If obtained $R_m = R_{2m}$, so
   $$x = \frac{a_{2m} - a_m}{b_m - b_{2m}} \ (mod\ n) \quad (15)$$
   If obtained $R_m = -R_{2m}$, so
   $$x = -\frac{a_{2m} + a_m}{b_m + b_{2m}} \ (mod\ n) \quad (16)$$

3. RESULT AND DISCUSSION

In this study, 10 prime p numbers were randomly selected, with 5 numbers with 2 digits and the remaining 5 with 3 digits. The selection of the p-value with small digits was made due to the limitations of the computational device. However, this experiment can provide a comparison between the many iterations of the initial algorithm and its modifications, i.e., in the worst case, the number of iterations of the modified Pollard Rho algorithm is equal to the number of iterations of the initial algorithm. In the best case, the modified algorithm has 6 iterations, while the initial algorithm has 26 iterations, meaning that it has succeeded in speeding up the process to 6.5 times. The experimental results are presented in Figure 5.

p	A,B	Titik P(x,y)	k(Q=kp)	Titik Q(x,y)	2i (Ri=R2i)	2i (Ri=-R2i
41	15 , 7	[20, 5]	5	[41, 41]	R 4	R 2
23	22 , 21	[5, 16]	5	[17, 15]	R 4	R 4
19	14 , 4	[5, 16]	6	[9, 7]	R 36	R 36
67	6 , 64	[31, 62]	1	[7, 39]	R 20	R 20
37	19 , 24	[23, 23]	2	[3, 16]	R 6	R 6
719	207 , 2	[674, 366]	4	[548, 350]	R 42	R 28
541	97 , 82	[144, 308]	1	[148, 364]	R 28	R 28
997	62 , 960	[888, 57]	2	[92, 753]	R 12	R 4
293	50 , 53	[163, 56]	5	[45, 227]	R 26	R 6
191	22 , 18	[135, 6]	7	[73, 186]	R 8	R 8

Figure 6. Experiment Results

In the experiment, the values A and B for the elliptic curve Equation (1) are chosen such that the number of points on the elliptic curve $E(\mathbb{Z}_p)$ is prime. Next, a random point is chosen $P(x, y) \in E$ and any natural number k for further calculation $Q = kP$ according to (2)-(6). According to (9), (15), and (16), it can be concluded that the value of k can be obtained after finding the value of i in iteration so that a "collision" occurs.

$$R_i = R_{2i} \quad (17)$$

Therefore, the two rightmost columns are assigned the value of $2i$ when the collision occurs. It means that both algorithms execute many iterations. Therefore, the rightmost column is the number of iterations of the modification algorithm.

The worst case, where the number of iterations of the modified algorithm is the same as the initial algorithm, occurs several times, for example, in the case of $p = 23$ and $p = 541$. The best case occurs at $p = 293$, where the modified algorithm runs 6.5 times faster than the original algorithm. It is much better than the modification algorithm's expected acceleration, i.e.$\sqrt{2} \approx 1.4$ times.

This study is a research development by [14] and [21]. Research [14] used the Pollard Rho algorithm without negation mapping, while research [21] used negation and Frobenius mapping but applied it to different fields, namely $GF(2^n)$. This study does not write down the results of the application of the Frobenius mapping because it does not change the number of iterations.

4. CONCLUSION

The application of negation mapping to the Pollard Rho algorithm can reduce the number of iterations that must be run until the ECDLP is solved. In the worst case, the modified algorithm is as fast as the original algorithm, while it runs 6.5 times faster in the best case. The average acceleration of the modified algorithm is 1.9 times, greater than the theoretical expectation i.e., $\sqrt{2} \approx 1.4$ times. It can provide resource savings when carrying out attacks with the Pollard Rho algorithm, especially when the field selected for the Elliptical Curve uses large prime numbers.

REFERENCES

[1] D. R. Kohel, “Cryptography,” 2015. http://iml.univ-mrs.fr/~kohel/tch/M2-CryptoSymetrique/crypto.pdf (accessed May 23, 2022).

[2] B. Schneier, *Applied cryptography: Protocols, algorithm, and source code in C*. The University of Michigan: Wiley, 1996.

[3] A. J. Menezes, P. C. Van Oorschot, and S. A. Vanstone, *Applied Cryptography*. Boca Raton: Taylor & Francis, 1997.

[4] ECRYPT (European Network of Excellence in Cryptology), “Hardness of the Main Computational Problems Used in Cryptography,” Leuven, 2007.

[5] T. Elgamal, “A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms,” *IEEE Trans. Inf. Theory*, vol. 31, no. 4, pp. 469–472, 1985.

[6] W. Diffie and M. E. Hellman, “New Directions in Cryptography,” *IEEE Trans. Inf. Theory*, vol. 22, no. 6, pp. 644–654, 1976.

[7] H. R. Hashim, “The Discrete Logarithm Problem in the ElGamal Cryptosystem over the Abelian Group U(n) Where n= p^m ,or 2p^m,” *Int. J. Math. Trends Technol.*, vol. 7, no. 3, pp. 184–189, 2014.

[8] N. Koblitz, “Elliptic curve cryptosystems,” *Math. Comput.*, vol. 48, no. 177, pp. 203–209, 1987.

[9] K. Lauter, “The advantages of elliptic curve cryptography for wireless security,” *IEEE Wirel. Commun.*, vol. 11, no. 1, pp. 62–67, 2004.
[10] A. Blumenfeld, “Discrete Logarithms on Elliptic Curves,” *J. Rose-Hulman Undergrad. Math. J.*, vol. 12, no. 1, pp. 30–57, 2011.
[11] A. A. Neamah, “New Collisions to Improve Pollard’s Rho Method of Solving the Discrete Logarithm Problem on Elliptic Curves,” *J. Comput. Sci.*, vol. 11, no. 9, pp. 971–975, 2015.
[12] J. H. Silverman, *The Arithmetic of Elliptic Curves*. New York: Springer, 2009.
[13] ISARA, “Isogeny-Based Cryptography Tutorial,” 2019. [Online]. Available: https://www.isara.com/resource-center/isogeny-based-cryptography-tutorial.html.
[14] M. Z. Seet, J. Franklin, and P. Brown, “Elliptic Curve Cryptography Improving the Pollard-Rho Algorithm,” University of New South Wales, Australia, 2007.
[15] J. M. Pollard, “Monte Carlo Methods for Index Computation (mod p),” vol. 32, no. 143, pp. 918–924, 1978.
[16] I. Duursma, P. Gaudry, and F. Morain, “Speeding up the Discrete Log Computation on Curves with Automorphisms,” in *Advances in Cryptology - ASIACRYPT’99*, 1999, pp. 103–121.
[17] A. Menezes, E. Teske, and A. Weng, “Weak Fields for ECC,” in *Topics in Cryptology – Cryptographers’ Track at the RSA Conference*, 2004, pp. 366–386.
[18] D. Hankerson, A. Menezes, and S. Vanstone, *Guide to Elliptic Curve Cryptography*. New York: Springer, 2004.
[19] A. J. Menezes, T. Okamoto, and S. A. Vanstone, “Reducing Elliptic Curve Logarithms to Logarithms in a Finite Field,” *IEEE Trans. Inf. Theory*, vol. 39, no. 5, pp. 1639–1646, 2003.
[20] E. Teske, “Speeding Up Pollard’s Rho Method for Computing Discrete Logarithms,” in *International Algorithmic Number Theory Symposium*, 1998, pp. 541–554.
[21] I. Muchtadi-Alamsyah, T. Ardiansyah, and S. S. Carita, “Pollard Rho Algorithm for Elliptic Curves over GF(2^n) with Negation Map, Frobenius Map, and Normal Basis,” *Far East J. Math. Sci.*, no. IV, pp. 385–402, 2013.