Elliptic Curve Cryptography for Image Encryption

Hokki Suwanda 13519143[1]
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
[1]13519143@std.stei.itb.ac.id

***Abstract*—Images being sent through the network needs to be secured. The security is currently a very big issue. Many methods and means have been defined to secure images sent through the network, whether by private means or by using public. One major method is elliptic curve image cryptography system. The method involves encrypting and decrypting image using the principle of elliptic curve arithmatics.**

***Keywords*—decrypting, elliptic curve, encrypting, security.**

I. INTRODUCTION

Privacy is a thing needed by everyone. Even more, all companies also need privacy. Generally, every party needs privacy. Until now, there have been many ways to keep the privacy of a party because there are secrets that cannot be told to other party. Privacy is not only about secret, but also ways of communication. Many private ways of communication exists. One of the private ways is cryptography. Ironically though, some cryptography also require public involvement to be successful.

Prime numbers is one of many big elements in mathematics, especially discrete mathematics. There have been so many scientists investigating and researching about prime numbers. Not only investigating and researching prime numbers, scientists also spent so much time on identifying prime numbers by using methods of primality testing. Primality testing can also be tricky, as not all the result is correct. Some primality testing algorithm existed right now can still be wrong in proving a prime, especially a very big prime number

Prime number is a very important fundament in number theory. Various algorithms regarding prime numbers have been publicly announced. The first algorithm to be presented is the Sieve of Eratosthenes. Long after, many more scientists presented algorithms about prime numbers. Some renowned scientists are Fermat, Euler, Gauss, and Legendre. These inventions starts some approaches to primality testing in the 1970's.

On cryptography, prime numbers are used so casually. Its use in cryptography is so natural as if cryptography inhales and exhales prime numbers. Generating prime numbers are very important on cryptography. Not only that, there are also many intersections between generating prime numbers and primality test, also generating prime numbers and cryptography. Innovations of prime number changes the world of mathematics and number theory, especially cryptograpy.

Cryptography are used and being strengthened using prime numbers to increase its security. Due to the wide usage of images in the internet. The security and privacy of images being sent through networks is an urgent issue. Recently, many researchers have introduced many ways in image cryptography. The ways, of course, are classified into symmetric key and public key. Symmetric key cryptography are easy to implement and require less time than public-key cryptography. However, the key management has become a very big problem nowadays, especially with advanced technologies, because the key needs to be distributed in the network. Public-key cryptography systems can solve the problem of key management because it uses two types of keys, private key and public key. Public key, as it sounds, is known to the public. Private key, however, is only known to an individual and does not need to be distributed. Identifying the private key from public key is possible, but not easy.

The hardest mathematical problems in world of cryptography are discrete logarithm and integer factorization. These two problems are used in many public-key cryptography and digital signature algorithms. In 1985, Miller and Koblitz introduced elliptic curve public-key cryptography scheme. In fact, cryptographers state that elliptic curve cryptography scheme has higher security and performance efficiency compared to other public-key cryptography algorithm. The parameters used on elliptic curve cryptography are smaller than other algorithms, but the level of security is equivalent. Elliptic curve Diffie-Hellman key exchange system is used widely. Many schemes for image encryption have been proposed.

II. PRIME NUMBER

A. Prime Number

Prime number is an integer bigger than 1 that has no factors except 1 and itself. The simple statement denotes the definition of prime number. Even so, identifying the elements of set of prime numbers are not a trivial matter. Not only that, determining whether a number

B. Probabilistic Primality Test

Fermat's theorem asserts that if n is a prime and a is any integer where

$$1 \leq a \leq n-1 \quad (1)$$

Then

$$a^{n-1} \equiv 1 \ (mod\ n) \quad (2)$$

Given a positive integer n to be identified, finding any integer a

in the given interval that does not satisfy the equivalence above is enough to prove that n is not a prime. For odd composite integer n, An integer a such that the equivalence is not satisfied is called a *Fermat witness* for n. With the same n and a, n is a *pseudoprime to the base a* if it satisfies Fermat's theorem. An example of pseudoprime is odd composite integer $n = 341$, because it satiesfies the theorem for $a = 2$.

The algorithm of Fermat primality test is as follows.

FERMAT(n)
INPUT : an odd integer $n \geq 3$ and security parameter $t \geq 1$.
OUTPUT : an answer "prime" or "composite" to the question : "Is n prime?"

1. For i from 1 to t :
   1.1. Choose an integer a, where $2 \leq a \leq n-2$
   1.2. Compute $a^{n-1} \bmod n$ and store it in r
   1.3. If $r \neq 1$, n is composite
2. n is prime

If the algorithm determine n as composite, then n is surely composite. However, if the algorithm determine n as prime, there are chances that n is not a prime number. Other than Fermat's algorithm, there exists many other algorithms such as Miller-Rabin test, and Solovay-Strassen test.

Because of the probabilistic primality, mathematicians looked for and analyzed ways to prove primality of an integer n. Such ways are called true primality test.

C. True Primality Test

As its name suggests, true primality test is a test to prove that a number is a prime number accurately. However, before applying true primality test to an integer n, the integer to be proven should be tested by probabilistic primality test. True primality test algorithms mostly have high time complexity. There are many true primality test algorithms, some of them are Lucas-Lehmer true primality test, factorization test, Pocklington's theorem, Jacobi sum test, and elliptic curve true primality test.

Usually, for asymmetric cryptography, a very high prime number is needed so that the security of the code is more guaranteed. It is hard to know whether a very big number is a prime number. Lucas-Lehmer primality test algorithm is as follows.

INPUT : a number $n = 2^s - 1$ with $s \geq 3$.
OUTPUT : an answer "prime" or "composite" to the question : "Is n prime?"

1. Check whether s has factors between 2 and $\lfloor \sqrt{s} \rfloor$, if it does, n is composite
2. Let $u \leftarrow 4$
3. For i from 1 to $s-2$, compute $(u^2 - 2) \bmod n$ and store the result in u.
4. If $u = 0$, n is prime. Else n is composite

Lucas-Lehmer uses numbers called as Mersenne number, which is $n = 2^S - 1$ for $S \geq 3$.

Another way to prove primality is by using Pocklington's theorem for factorization. Pocklington's theorem is as follows. For an integer $n \geq 3$, let n = RF + 1 where the prime factorization of F is $F = \prod_{j=1}^{t} q_j^{e_j}$. If there is an integer a satisfying Fermat's theorem and $gcd(a^{\frac{n-1}{q_j}} - 1, n) = 1$ for each j, $1 \leq j \leq t$, Then every prime divisor of n is congruent to 1 modulo F. Furthermore, if $F > \sqrt{n} - 1$, then n is a prime number.

Another method for true primality test is using elliptic curves, the same method used for DSA in cryptography. Pocklington's theorem is analogue to an elliptic curve. With that fact, this algorithm was made. This method are also called as Atkin's test. This algorithm run at a very little time. This algorithm can prove the primality of numbers more than 1000 digits long (in decimal). Even though this algorithm run at a very little time, the details of the algorithm are very complicated and hard to understand.

In cryptography, DSA (Digital Signature Algorithm) also require two prime numbers, just like usual .

D. Prime Number Generation

In cryptography, DSA (Digital Signature Algorithm) also require two prime numbers, similar to public-key cryptography (or asymmetric cryptography). Thus, generating prime numbers, especially big prime numbers, are very important in a matter of public-key cryptography. One example is NIST DSA which requires two primes p and q where q is a 160-bit prime (49 digits in decimal), p is a L-bit prime, $L = 512 + 64l$ for some $0 \leq l \leq 8$, and q divides $p - 1$. The prime number generated by NIST DSA makes use of probabilistic Miller-Rabin primality test.

INPUT : integer l satisfying the paragraph above.
OUTPUT : integers q and p, where $p \equiv 1 \pmod q$, satisfying requirements in the paragraph above

1. Compute L and find n and b such that $L - 1 = 160n + b$
2. Repeat :
   2.1. Choose a random 160-bit-or-less integer s.
   2.2. Form q from s by setting the most significant and least significant bits to 1.
   2.3. Implement Miller-Rabin test on q
   Until q is determined as prime.
3. Let $i \leftarrow 0$, $j \leftarrow 2$
4. While $i < 4096$ do :
   4.1. For k from 0 to n do : let $V_k \leftarrow (s + j + k) \bmod 2^g$
   4.2. Let $W = V_0 + V_1 2^{160} + V_2 2^{320} + \ldots + Vn 2^{160n}$ and $X = W + 2^{L-1}$.
   4.3. Compute $c = X \bmod 2q$ and set $p = X - (c - 1)$.
   4.4. Implement Miller-Rabin test on p. If p is a probable prime, stop..
   4.5. Increase i by 1 and j by $n + 1$.
5. Return to step 2

The algorithm of NIST DSA can be used to generate a very big prime by enhancing the algorithm with elliptic curve algorithm to generate even bigger prime number. However, the algorithm is very complex. Other than NIST method for generating DSA primes, Maurer's recursive algorithm can be used to generate primes. However, the weakness of this algorithm is that it is running heavily, because of its recursive nature, and that it takes too much time.

III. CRYPTOGRAPHY

A. Cryptography

Cryptography is an application of number theory, especially modular arithmatics and prime number. According to KBBI, cryptography has two meanings: investigation of secret codes, and a technique using mathematic algorithm to change data to hide its meaning so that people without the key cannot undo the changes. These can also mean that cryptography can be used as a mean of private communication between two or more people (or sides) affiliated with each other to prevent other people from knowing their secrets.

B. Public-Key Cryptography

Symmetric cryptography is a system of cryptography where the encryption key is the same as the decryption key. Symmetric cryptography is very versatile to brute-force attack, as it does not require much time to be solved. On the other hand, symmetric cryptography is very limited. Right now, on 64-bit operating system, there exists only 256 characters that can be encoded to ASCII Codes. Because of its versatility, ways other than symmetric cryptography were researched. The solution was public-key cryptography.

Public-key cryptography is also called as asymmetric cryptography. Unlike symmetric-key cryptography, the encryption key and the decryption key is different. Just like the name, public-key cryptograpy involves on informing keys publicly. Only the accomplice can know the information behind the public key and decrypt the message. Public-key cryptography schemes are often slower than symmetric cryptography because the problem is more complex.

There are many public-key cryptography algorithm. Some of them are RSA, Blum-Goldwasser, Pohlig-Hellman, El-Gamal algorithm, and elliptic curve algorithm. The El-Gamal algorithm makes use of the prime generator on generating a very big prime p. The security of this algorithm is based on the discrete logarithm problem and the implementation of Diffie-Hellman key exchange agreement. Basic El-Gamal cryptography algorithm is as follows:

1. Generate a prime p and a generator α. The prime and the generator are used for all sides.
2. The receiver generates a random integer A as his/her private key and a corresponding public key (p, α, α^A)
3. On encrypting the message, the sender should have received the public key. Represent the message as an integer m in interval $[0, p-1]$.
4. Generate a random integer B, the sender's private key, and send the encrypted message as (α^B, $m.(\alpha^A)^B \bmod p$) to the receiver. Let it be at the form of (γ, δ)
5. The receiver uses his/her private key A and receiver the message (as an integer m) by computing (γ^{-A}). $\delta \bmod p$.

The bigger the prime p and the generator α being used, means that the chances of it being unsecure is small. The reason is that it is hard to identify the big prime number being used. Thus, needing very much time to identify the prime p.

C. Digital Signature

One application of public-key cryptography for authentication service is digital signature. In signing and encrypting a message, the sender uses the private key (which is only known by the sender). The signature is then sent to the receiver together with the true message. Usually, to prevent the similarity of digital signature and the true message, hash functions are used before signing. An example of it in daily life is our signature in identification card. The private key is the signature. A *digital signature* is only known to its signer and contains some secret. There are many applications of digital signature, two of them are authentication and data integrity.

The first found method for digital signature was RSA signature scheme, which is still a recurring techniques even now. Of course, there are other digital signature techniques, with the different being in functionality, implementation, and time. There are some public-key cryptography used for signing, like El-Gamal cryptography system and RSA. On the other hand, there are also algorithms built completely only for signing like DSA and DSS.

According to Schneier, digital signature scheme must have some characteristics as follows:

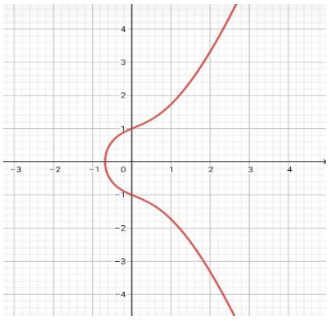
1. Signature is authentic.
2. Signature is a prove that the person giving the signature is affiliated.
3. Signature is a part of document, cannot be moved nor used more than once.
4. Signature is a prove that the person giving the signature has signed a document.
5. Signed documents are unmodifiable.

One example of digital signature algorithm on signing a message x is El-Gamal algorithm which is similar to El-Gamal public-key cryptography system is as follow:

1. Given p, α, α^A with A the receiver's private key
2. Generate a random integer B, the sender's private key, and send the sign as (γ, δ) where they are ($\alpha^B \bmod p$, $(x - A\gamma)k^{-1} \bmod (p-1)$) to the receiver.
3. On verifying the signature, check whether $(\alpha^A)^\gamma \gamma^\delta \equiv \alpha^x \pmod p$

D. Elliptic Curve Cryptography

An elliptic curve is an equation that consists of two variables along with their coefficients. Mathematically, the variables and coefficients are of an infinite field. However, in cryptography, the variables and coefficients are limited to a finite field. The equation of an elliptic curve is $y^2 = x^3 + ax + b$. An example of elliptic curve is $y^2 = x^3 + x + 1$. As seen in the plot below, the curve is horizontally symmetric.



Picture 1. The plot of $y^2 = x^3 + x + 1$ in *GeoGebra*

As there are many different elliptic curves, an elliptic curve is chosen as a rule. Other elliptic curves must have similar plot to the said curve. The condition is that $4a^3 + 27b^2 \neq 0$ must be fulfilled.

Let $p > 3$ be a prime. An elliptic curve $y^2 = x^3 + ax + b$ over Z_p is a set E of all (x, y) that satisfies $y^2 \equiv x^3 + ax + b \pmod p$, with a and b are constants that satisfy $4a^3 + 27b^2 \neq 0 \pmod p$. And then, a particular point O, which is point at infinity, is aslo added to set E as an element.

Elliptic curve includes three operations:

1. Addition.
   For given P and Q where $P(x_1, y_1) + Q(x_2, y_2) = R(x_3, y_3)$, R is calculated by
   $$x_3 = m^2 - x_1 - x_2 \quad (3)$$
   $$y_3 = m(x_1 - x_3) - y_1 \quad (4)$$
   $$m = (y_2 - y_1)(x_2 - x_1)^{-1} mod\ p, P \neq Q \quad (5)$$
   $$m = (3x_1^2 + a)(2y_1)^{-1}\ mod\ p, P = Q \quad (6)$$
2. Subtraction
   $$P(x_1, y_1) - Q(x_2, y_2) = P(x_1, y_1) + Q(x_2, y_2) \quad (7)$$
3. Multiplication
   Multiplication is repeated addition. Multiplication for $k = 2$ is often called doubles, as the result is $P + P$ which is the double of the base point P.
   $$kP = P + P + \cdots + P\ for\ k\ times \quad (8)$$

Let $Q = kP$ where Q, P is in $E_p(a, b)$ and $k < p$. Calculating Q given k and P is very easy compared to calculating k given P and Q. The difficulty of calculating k given Q and P is called discrete logarithm problem for elliptic curves. To prevent the receiver to be confused from the discrete logarithm problem, Diffie and Hellman proposed a key exchange protocol. The protocol is called Diffie-Hellman key exchange protocol. The protocol, for elliptic curve cryptography, is as follows.

1. The sender A picks a random private integer $k_A < p - 1$ and computes the public key $K_A = k_A.P$ and sends it to the receiver, B.
2. The receiver B picks a random private integer $k_B < p - 1$, and computes the public key $K_B = k_B.P$ and sends it to the sender, A.
3. Both the sender and the receiver computes their shared secret key $K_{AB} = k_B.K_A = k_A.K_B$.

Diffie-Hellman key exchange protocol is popularly used in elliptic curve cryptography.

One scheme of elliptic curve cryptography is El-Gamal elliptic curve cryptography system. The parameters needed for El-Gamal elliptic curve cryptography system is the prime p, elliptic curve E_p, a generator point P. Before encrypting a message, assume both the sender and the receiver has traded their public key to each other. The scheme is as follows:

1. Generate a random integer $k_B < p - 1$ which is the private key of the receiver.
2. The receiver computes $K_B = k_B P \pmod p$
3. The receiver then sends the public key in form of thee tuple (p, P, K_B) to the sender

Any sender affiliated with the receiver can then encrypt a message by using the public key sent by the receiver as follows:

4. Let the message be represented as a point x on the elliptic curve.
5. Generate a random integer $k_A < p - 1$ which is the private key of the sender
6. Compute $K_A = k_A P$.
7. Encrypt the message and form a pair of points $(K_A, x + k_A K_B)$ where $k_A K_B$ is their shared secret key K_{AB}.

The receiver then decipher the cipher message by using K_A sent by the sender as follows:

8. Let the receiver received the cipher message in a form of tuple (y_1, y_2)
9. The receiver computes their shared secret keys K_{AB} by $k_B.y_1$
10. Subtract K_{AB} from y_2. The result is the true message from the sender to the sender.

By further improving the algorithm, it can now encrypt and decrypt an image, whether it is grayscale or colored. The method used is basically the same, but more tricky. What makes it tricky is that it can fail. If there is a parameter being used incorrectly, then the encryption can fail. The same thing goes to decrypting the image. If a parameter is used incorrectly, the decryption can fail and the true image can not be retrieved.

E. Image Encryption

Encrypting an image is a continuation of encrypting a message. In encrypting a message, we encrypt every two consecutive characters as a point on the curve, resulting in another point which is then converted into two characters again. Because image is made of pixels, images can be encrypted by manipulating the coordinates of pixels or the color composition. Encrypting the coordinates of pixels will make the encryption easy to decrypt. Yet encrypting the color compositions is tricky to do.

If the sender wants to send an image m, the image of course needs to be encrypted. Images are made up of pixels. If a cryptography algorithm is implemented on every pixel, the time taken will be very long. However, if the images are grouped or divided into blocks of pixels, the number of operations will be decreased drastically. Thus, taking much less time and effort. The number of pixels that can be grouped depends on the prime p chosen (p as in Z_p and E_p). As the parameter p of the elliptic curve increases, more pixels can be grouped together. For example, for p a 128 bit prime, 16 pixels can be grouped together, for p a 512 bit prime, 64 pixels can be grouped together. The key sharing method is by using Diffie-Hellman Key Exchange method. To encrypt the image, the sender (or the medium) will do the following

1. Divides the image m into blocks of pixels. Convert every block of pixels into a very big integer that is smaller than p.
2. Pair up the results obtained from step 1 and store them in P_m as an input point for ECC system
3. Select a random integer k as the private key and compute kP and kK_B
4. Compute the addition of kK_B and each value of P_m, let the result be P_R. Accumulate values of P_R as a list of cipher text
5. Convert the cipher text to value in the interval [0, 255]
6. Convert the list into a cipher image

The encryption method can be further enhanced by

implementing the encryption on every pixel of the image, multiple hash functions, chaotic system, or digital signatures. More ways can also be used. For example, using bigger number as the parameters, or using a prime generator to generate the parameters. The latter method is not recommended as it is difficult to record the parameters used. One example of optimized method is down below.

1. Choose an elliptic curve over Z_p, any integer a, b, and any point P at the elliptic curve
2. For every pixel. Decompose the pixel into three channels : red, green, and blue. For each channel, convert the values into a very big integer that is smaller than p
3. Pair up every two results from step 3 and store it as a point Q.
4. For every Q, compute the encrypted pixel R by adding Q to the shared secret key K_{AB}.
5. The value of each R should be very big. Convert the result to be in the interval [0, 255].
6. Merge the result R of the same block for three different channels into a tuple C of three elements, which shows the color composition of the said pixel.
7. The sender can further add more security by providing a digital signature using hash functions

The encrypted image is then sent to the receiver to be decrypted.

Picture 2. True image

Picture 3. Encrypted image

Picture 4. Decrypted image

F. Image Decryption

When the receiver received an encrypted image, the receiver needs to decrypt the cipher image. If the sender provided a digital signature for the encrypted image, the receiver has to verify the signature to identify whether the image is from the sender. By assuming that the receiver received the image encrypted with the optimized method above, the receiver have to decrypt the image by following the steps below.

1. Verify the signature of the image. If the signature is verified correctly, continue to step 2.
2. Each pixel should have the RGB values which is in the interval of [0, 255].
3. Decompose each pixel into three channels, each representing red, green, and blue
4. For every channel, convert the value in interval [0, 255] to a very big integer smaller than p
5. Form a pair for every two successive big integers, and store it as R.
6. Let the public key K_A is sent by the sender. The receiver computes the shared secret key K_{AB} by multiplying K_A with the receiver's private key k_B.
7. Subtract the shared secret key from R, let the result be Q. Q is a point on the elliptic curve
8. For every Q, split the absis and the ordinate as two consecutive values. Each value is a very big integer smaller than p
9. Convert each big integer into a value in the interval [0, 255]

IV. SIMULATION RESULT

The specifications of laptop being used in the simulation is Intek(R) Core(TM) i3-2370M CPU @ 2.40GHz, 16GB RAM, Python 3.8 on Windows 7 (64-bit). The elliptic curve used is 256 bit standard elliptic curve from brainpool. The parameters of the elliptic curve are as follows:

p = 76884956397045344220809746629001649093037950200943055203735601445031516197751;

a = 56698187605326110043627228396178346077120614539475214109386828188763884139993;

b = 17577232497321838841075697789794520262950426058923084567046852300633325438902;

P = (63243729749562333355292243550312970334778175571054726587095381623627144114786, 38218615093753523893122277964030810387585405539772602581557831887485717997975.

The keys used are as follows:

K_B = (33874030876639903844966121129493451012840373632955525894922405493918875061651, 29818450905109508559617010717888628375364158100660378833726564116780315410377);

k_B = 76159756724549015673500497763136124364084154405538822063208617086554866849 74;

K_A = (40813494392004829032540258834179916309364136341594960116449051598828856101709, 58626250424632274958012609589348403196480545956260645973454106396601636021250);

k_A = 53010406328880877434018021562413287898703947701252470024537641509420262043417;

K_{AB} = (448634522843820414701023913935323862263112458136872709628301070882361184636843263514115296925381492925371186067369960442870022800990397785476377316670596).

Where K_B is the public key sent by the receiver, k_B the receiver's private key, K_A is the public key sent from the sender, k_A is the sender's private key, and K_{AB} their shared secret key. The sender sends the encrypted image as shown in picture 3. The public key of the receiver was used to encrypt the image. When the receiver receives the cipher image, the private key sent by the sender and the receiver's private key is used to decipher the image into what is shown in picture 4.

When tested manually using python, the encryption is successful. However, for unknown reasons, the decryption is not perfect. The decryption produces an image similar to the true image, and not the image before it was encrypted. The comparison of the true image, encrypted image, and decrypted image is shown in figure 2, 3, and 4 respectively.

V. Conclusion

Cryptography has many uses, one of them is to secure image sent in the internet. There are many algorithms of cryptography available. One of them is Elliptic Cuve algorithm. Elliptic Curve algorithm makes use of Diffie-Hellman Key Exchange Algorithm. Elliptic Curve is proven to be more secure than other cryptography algorithm with the same size of parameters.

VI. Acknowledgment

The writer is grateful to lecturers of IF2120 Discrete Mathematics who taught, helped, and guided the writer on weekly lectures. None of this would be possible without the lecturers giving knowledge. The writer is also grateful to friends that helped the writer when the writer was very confused on what to write and on answering some hard questions. The writer is very grateful to close friends who motivated and encouraged the writer when writing this paper.

References

[1] Sumarkidjo, dkk. (2007). *Jelajah Kriptologi*. Jakarta : Lembaga Sandi Negara.

[2] Dwiyanti, Latifa. (2020). *Database Security*. https://www.youtube.com/playlist?list=PLoJAIIF7j9wcI2Le3JemGxoITo VhWXWzW. Diakses pada 26 November, 2020.

[3] Dwiyanti, Latifa. (2020). *Database Security #2*. https://www.youtube.com/playlist?list=PLoJAIIF7j9wcay_l8YNTOXZbe ZSe2QQhk. Diakses pada 26 November 2020.

[4] Uzunkol, Osmanbey. (2004). *Atkin's ECPP (Elliptic Curve Primality Proving) Algorithm*. Technical University of Kaiserslauten.

[5] A.O.L Atkin, F. Morain. (1993_. *Elliptic Curves and Primality Proving*. American Mathematical Society.

[6] Everest, Graham. (2003). *Primes Generated by Elliptic Curves*. American Mathematical Society.

[7] Gallier, Jean and Jocelyn Quaintance. (2019). *Notes on Primality Testing and Public Key Cryptography Part 1: Randomized Algorithms Miller-Rabin and Solovay-Strassen Tests*. University of Pennsylvania

[8] Menezes, Alfred J and friends. (1996). *Handbook of Applied Cryptography*. Massachusetts Institute of Technology.

[9] Jajodia, Sushil. (2011). *Encyclopedia of Cryptography and Security*. Springer Science+Business Media.

[10] Stallings, William. (2017). *Cryptography and Network Security. Priciples and Practice. Seventh Edition Global Edition*. England : Pearson.

[11] Schneier, Bruce. (2015). *Applied Cryptography: Protocols, Algorithms, and Source Code in C*. Indianapolis: John Wiley & Sons.

[12] Li, Frank. (2011). *An Overview of Elliptic Curve Primality Proving*.

[13] Parama, Rizki Ihza. (2016). *Aplikasi Teori Bilangan dalam Kriptografi untuk Keamanan Teknologi Informasi dalam Bentuk Algoritma DSA*. Institut Teknologi Bandung.

[14] Severina, Verena. (2016). *Penggunaan Teori Bilangan dan Kriptografi dalam Peningkatan Keamanan Aplikasi Personal and Group Messaging*. Institut Teknologi Bandung.

[15] Shukla, Ashutosh; Jay Shah; Nikhil Prabhu. (2013). *Image Encryption using Elliptic Curve Cryptography*. St Francis Institute of Technology.

[16] Abdelfatah, Roayat Ismail. (2020). *Secure Image Transmission Using Chaotic-Enhanced Elliptic Curve Cryptography*. Tanta University.

[17] Singh, Laiphrakpam Dolendro; Khumanthem Manglem Singh. (2015). *Image Encryption using Elliptic Curve Cryptography*. India National Institute of Technology.

[18] Goldwasser, Shafi; Joe Kilian. (1999). *Primality Testing using Elliptic Curves*. Massachusetts Institute of Technology.

[19] Savas, Erkay; Thomas A. Schmidt; Cetin K. Koc. *Generating Elliptic Curves of Prime Order*. Oregon State University.

[20] Setyobudi, Febrina Mediawati. (2013). *Penggunaan Kriptografi Kurva Eliptik pada Proses Penyandian ElGamal*. Universitas Islam Negeri Maulana Malik Ibrahim.

[21] Damanik, Putri S E A. (2019). *Implementasi Algoritma Elliptic Curve Cryptography (ECC) untuk Penyandian Pesan pada Aplikasi Chatting Client Server Berbasis Desktop*. STMIK Budi Darma.

Pernyataan

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 3 Desember 2020

Hokki Suwanda (13519143)