# Comparative Analysis between Elgamal and NTRU Algorithms and their implementation of Digital Signature for Electronic Certificate

Original Scientific Paper

**Bambang Harjito**
Sebelas Maret University,
Faculty of Math and Natural Science, Department of Informatics
Surakarta, Indonesia
bambang_harjito@staff.uns.ac.id

**Tri Setyawati**
Sebelas Maret University,
Faculty of Math and Natural Science, Department of Informatics
Surakarta, Indonesia
trisetyawati11@student.uns.ac.id

**Ardhi Wijayanto**
Sebelas Maret University,
Faculty of Math and Natural Science, Department of Informatics
Surakarta, Indonesia
ardhi.wijayanto@staff.uns.ac.idr

***Abstract*** *– The emergence of electronic certificates, which are official documents in the form of digital files transmitted via the internet, facilitates the exchange of information. However, internet use has risks, such as data theft for fabricating and modifying information. This problem can be solved by applying a digital signature. The main concern in this research is how to perform a comparative analysis between asymmetric cryptographic Elgamal and NTRU (Nth-Degree Truncated Polynomial Ring) algorithms and their implementation of a digital signature as an effort to improve information security in electronic certificates. The stages of the research method are divided into the key generation process, signing, and verification. In the signing and verification process, the SHA-512 hash function is also used for hashing messages to be encrypted-decrypted and QR Code as the signature. Comparison of performance of NTRU with Elgamal algorithms required running at a pdf extension with security levels 80,128,192, 256 bits will be obtained from the templates.office.com website. The results obtained that the El Gamal algorithm is better than the NTRU algorithm, but at a higher security level, the NTRU algorithm is better than the Elgamal algorithm. In the verification experiment that has been carried out, it can be concluded that by using SHA-512 as a hash function, the N parameter used for NTRU must be greater than or equal to 512 to avoid error results from verification.*

***Keywords****: NTRU, Elgamal, Electronic Certificate, Digital Signature, SHA-512, QR Code*

## 1. INTRODUCTION

Electronic certificates are official documents in the form of digital files transmitted via the internet, where the internet itself is vulnerable to theft and falsification of information, such as the fabrication or modification of information [1,2,3]. To increase the security of electronic certificates. A security system in the form of digital signatures is applied to the electronic certificates. The digital signature is a cryptographic value that depends on the message's content and the message's sender. So that different messages with the same sender will have different digital signatures [4,5,6,7].

This study aims to implement a digital signature on an electronic certificate using the NTRU and Elgamal algorithm at security levels 80,128,192, and 256 bits to see whether NTRU is better or Elgamal is better. NTRU algorithm is an asymmetric algorithm. Asymmetric algorithms have different keys during the encryption and decryption process, namely the public and private keys. The public key is the key that is published and may be known by ev-

eryone, while the private key is a key that is kept secret and may only be understood by one person [8,9]. The NTRU algorithm's security level lies in the use of polynomials during the operation process, as well as the difficulty of finding a short vector of a lattice [10,11]. The level of security of the Elgamal algorithm lies in the difficulty of calculating discrete logarithms [12]. For comparison, the Elgamal algorithm is used at security levels 80,128,192, and 256 bits to see which algorithm has better performance. Elgamal's algorithm selection is the comparison because the study [37] shows that Elgamal is a better probabilistic algorithm than RSA and has a difficulty level that lies in discrete logarithm calculations and its ability to solve fundamental distribution problems. Besides that, there are still very few studies comparing the NTRU algorithm with the algorithm Elgamal.

This research aims to implement a digital signature schemes using the NTRU and Elgamal algorithm for electronic certificates, then analyze the running time on the NTRU and Elgamal algorithms based on the process generate keys, signing, and verifying, as well as analyze the results of electronic certificate verification.

## 2. RELATED WORK

The NTRU algorithm is a fast and lightweight public key algorithm to provide end-to-end security that can be used to improve document security standards with better encryption and decryption than the RSA and ECC algorithms [13,14,15]. In another study, to increase document security and facilitate the validation process, the use of digital signatures using the RSA algorithm [16,17] and SHA-512 algorithms can be applied where the method used is to generate a public key and a private key with RSA. The signing process is carried out by encrypting the message digest generated from the message hashing process with SHA-512 and then verifying electronic documents by matching the results of document decryption and SHA-512 hashing of documents [18,19,20,21] in a similar study [3]. With SHA-3 hashing function and super encryption combination of RSA and AES, with QR-Code scheme to accommodate the signature code. In this implementation, the certificate will be signed with the SHA-3 hashing process sequence, encrypted with RSA, encrypted with AES, and ends by embedding the QR-Code that has been generated from the AES encryption results on the electronic certificate.

## 3. THEORY USED

This section discusses the theoretical background to analyze the comparison between the NTRU and Elgamal Algorithms and the implementation of digital signatures on electronic certificates.

### 3.1. DIGITAL SIGNATURE

Digital Signature is a means used to view authentication on digital messages, both messages transmitted through communication channels and electronic documents; what is meant by digital signatures are signatures that have a cryptographic value that depends on the content of the message and the sender of the message, so that the message different ones with the same sender will have different digital signatures [14]. The term direct digital signature refers to a digital signature scheme that only involves the communicating party (sender, receiver). Digital signature schemes are similar to asymmetric cryptographic systems in that they involve public and private keys and run an algorithm that uses these keys to sign and verify [22,23].

### 3.2. NTRU (NTH-DEGREE TRUNCATED POLYNOMIAL RING UNITS) ALGORITHM

NTRU uses addition and multiplication operations in line with Ring, which is an algebraic object that has two operations, addition, and multiplication, which are related via the distributive law [22, 23], NTRU works with rings. An element will be written as a polynomial or vector according to Equation (1).

$$F = \sum_{i=0}^{N-1} F_i x^i = [F_0, F_1, \dots, F_{N-1}] \quad (1)$$

Key Generation: Choose two polynomials $f \in Lf$ $f$ $\in$ and $g \in Lg$. Polynomial f must meet the additional requirement that it has inverse modulo $q$ and inverse modulo $p$. This inverse can be expressed $F_p$ and $F_q$ so that the result is:

$$F_q * f \equiv 1 \ (mod\ q) \quad (2)$$

$$F_p * f \equiv 1 \ (mod\ p) \quad (3)$$

Next calculate $h$ with Equation (4)

$$h \equiv pF_q * g \ (mod\ q) \quad (4)$$

Where $h$ is a polynomial that functions as a public key and a polynomial $f, fp$ as private keys.

NTRU encryption: Selects m as messages from a set of plaintexts $L_m$. Then randomly choose the polynomial $L_\phi$ and use the public key $h$ to compute the encrypted message $e$ by Equation (5)

$$e \equiv \Phi * h + m \ (mod\ q) \quad (5)$$

Polynomial $e$ is an encrypted message that will be sent to the recipient of the message.

NTRU decryption: In the decryption of a received e-message, the process is carried out using the private key $f$ to calculate the value of a with Equation (6).

$$a \equiv f * e \ (mod\ q) \quad (6)$$

Where the coefficient a is in the interval from $q/2$ to $-q/2$. Now with $a$ as a polynomial with integer coefficients and a private key, $F_p$, which can be used to recover $m$ messages with Equation (7)

$$m \equiv F_p * a \ (mod\ p) \quad (7)$$

### 3.3. ELGAMAL ALGORITHM

Key Generation: Elgamal has a parameter of key size, which will later be used to determine positive prime numbers and integers that are primitive roots of $p$. To generate a public key and a private key is done by choosing a random number $x$, provided that than calculate the value of $y$ with Equation (8)

$$y = g^x \ (mod\ p) \tag{8}$$

The result of key generation is in the form of private key $x$ and public key $y$, $g$, and $p$.

Elgamal Encryption: Before performing the encryption process, first declare the message as an integer $m$ and must lie in the range [0, $p$-1]. For large $m$, divide $m$, into smaller blocks so that each block represents a value in the range [0, $p$-1]. The encryption steps are as follows:

1. Choose a random number $k$, provided that
   $1 \leq k \leq (p-1)$
2. Encrypt message m into value pairs ($a$, $b$) with the Equation:

$$a = g^k \ (mod\ p) \tag{9}$$

$$b = y^k m \ (mod\ p) \tag{10}$$

The pairs $a$ and $b$ are the ciphertext for message $m$. So, the ciphertext size is twice the size of the plaintext.

Elgamal's description: For decryption, the private key $x$ is used to decrypt a and b into plaintext $m$ with Equation [11].

$$m = b(a^x)^{-1} \ (mod\ p) \tag{11}$$

### 3.4. SHA-512 (SECURE HASHING ALGORITHM)

SHA (Secure Hashing Algorithm) is designed by the National Security Agency (NSA). SHA security is based on the fact that a birthday attack on a digest of n bits results in a collision with a work factor of about 2n/2 [24]. SHA-512 is one of the results of the revision of the FIPS standard in 2002, which defines three new versions of SHA, with hash values of 256, 384, and 512-bits long, known as SHA-256, SHA-384, and SHA-512. Collectively, these hash algorithms are known as SHA-2. This new version has the same basic structure and uses the same types of binary logical operations and modular arithmetic as SHA-1 [12].

### 3.5. QR CODE

QR Code is a two-dimensional matrix symbology with a position detection pattern at three angles initially designed for very high-speed reading and Omnidirectional reading. The QR Code was developed to increase the speed of reading complex structured 2D barcodes. Other QR Code features are bulk data capacity, high data density, and selectable levels of error correction capability [25, 26,27,28]. QR codes store data using a graphical representation. The essence of this representation is based on the arrangement of several simple geometric shapes on a fixed space [29,30,31,32].

## 4. PROPOSED WORK

This section provided an overview of our solution comparative analysis of the Elgamal and NTRU Algorithm, and the Implementation of digital signatures on an electronic certificate.

Fig.1 shows the comparative analysis model of the digital signature in the NTRU and Elgamal algorithms in carrying out the signing and verifying process. The comparative analysis model consists of two processes: (1) the signing process using both NTRU and Elgamal Algorithms and (2) the verifying process using both NTRU and Elgamal Algorithm. The process begins with the user who selects the electronic certificate with a pdf extension file. This file can be called plaintext.

a) The signing process using both NTRU and Elgamal Algorithms

The signing process begins with (1) calculating the hash value of the certificate file with the SHA-512 hash function, which produces a message digest (m). (2) these results are then encrypted with the Private Key and produce a message digest cipher (3) The cipher message digest is then stored in the form of a file pythons. (4) Generate a QR Code with the cipher message digest file address as the data. QR Code generation is done using a QR Code Generator. (5) Embed the QR Code as a signature into the certificate file. (6) Sending certificate file + QR Code to recipient.

b) The In the verifying process using both NTRU and Elgamal Algorithm

The verifying process is executed with (1) Look for the signature in the form of a QR Code, which is contained in the certificate file, then separate the QR Code from the certificate. (2) Calculate the hash value of the certificate file, and generate a message digest (m'). (3) Decode the QR Code to get the data in it, which is the address of the cipher message digests file. (4) Based on the address obtained, then look for the cipher message digest file to get the cipher message digest. (5) Decrypt the cipher message digest with the Public Key, which results in a message digest ($m$). (6) Comparing $m$ and $m'$. If $m = m'$, then it can be concluded that the certificate file is "Valid". Meanwhile, if it is not the same, it can be concluded that the certificate file is "Invalid

Testing is done by running the system to see if the system has running according to its function or not, by doing singing and verifying experiments document. The certificate document will be subject to a signing process and generate a new certificate document complete with a digital signature in it. This document will then be subject to two treatments, (1) The certificate document does not subject to any content changes, so

when the verification process is carried out with the system, it will display a result indicating that the certificate is correct or valid. (2) The certificate document is subject to content changes so that when the verification process is carried out with the system, it will display the following results indicating that the certificate is incorrect or invalid. After testing, system test results data in the form of running time during the key generation, sign, and verify processes. The data is then analyzed to see whether NTRU is better or Elgamal is better.

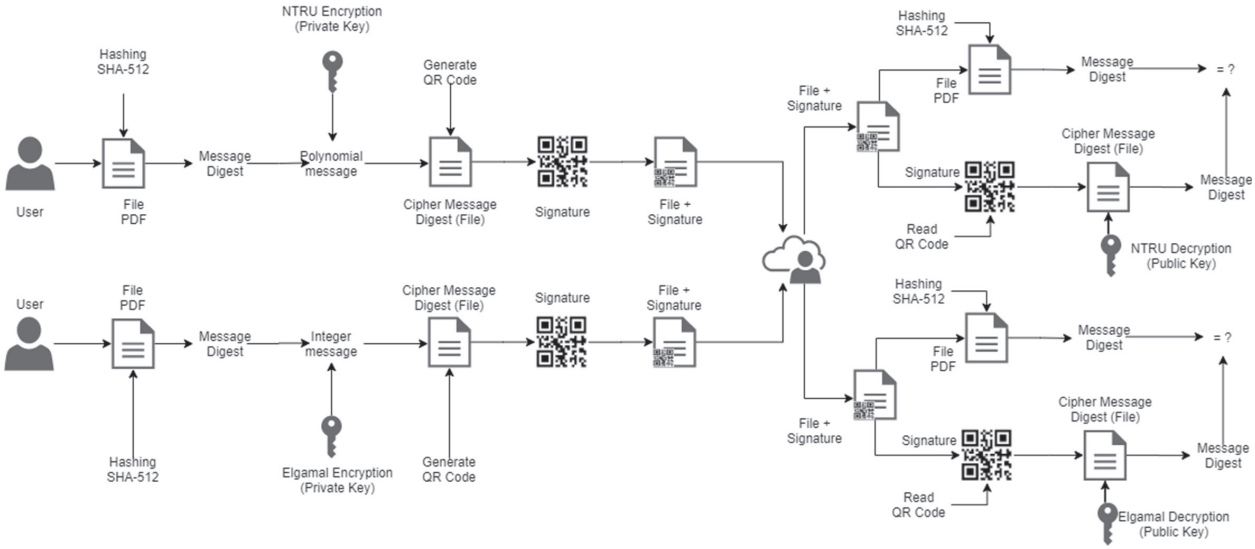


**Fig. 1.** Comparative Analysis of Elgamal and NTRU Algorithm and Implementation in the electronic certificate

## 5. RESULT AND DISCUSSION

In this section, perform a comparative analysis of the NTRU and ElGamal Algorithm and an Implementation in the electronic certificate.

### 5.1. DATA COLLECTION

The data for this research is a certificate with a pdf extension obtained from the templates.office.com website. The certificate can be depicted in Fig 2.

The performance of NTRU with ElGamal required data showing that the NTRU with such parameters will be comparable to ElGamal with so many bits. The data which can be used for this purpose is the security level of the NTRU algorithm with the ElGamal algorithm [33, 34, 35, 36]. The data can be seen in Table 1.


CERTIFICATE OF TRAINING

This certifies that

Tengiz Kharatishvili

has successfully completed training in video publishing

Rowan Murphy, Sr. Videographer

June 04, 20XX

Date

Fig. 2. Certificate Data with a pdf extension

**Table 1.** Security Level Elgamal and NTRU

| Security Level (bits) | NTRU | Elgamal (bits) |
|---|---|---|
| 80 | 251 | 1024 |
| 128 | 397 | 3072 |
| 192 | 587 | 7680 |
| 256 | 787 | 15360 |

### 5.2. NTRU IMPLEMENTATION OF THE DIGITAL SIGNATURE SCHEME

NTRU Key Generation: The main parameters of the NTRU algorithm are integers $N$, $p$, and $q$, and the four sets $L_f$, $L_g$, $L_m$, $L_\Phi$ polynomial of degree N-1 with integer coefficient. The integers $p$ and $q$ do not have to be prime, but provided that gcd($p$, $q$) = 1, and $q$ will always be much greater than $p$. NTRU works with rings polynomial $R = Z[X]/ (XN – 1)$.

For example, the low parameter is chosen to facilitate the ease of its calculation. The selected parameters are $N = 251$, $p = 3$, and $q = 2048$. Then randomly determine the polynomials $f$ and $g$ as follows:

$$g = x^{247} + x^{245} - x^{244} + x^{240} + x^{235} + x^{225} + x^{221} + x^{220} + x^{219} - x^{217} + \ldots - x^{36} - x^{34} + x^{31} - x^{30} + x^{29} + x^{27} + x^{21} + x^{19} - x^{16} + x^{15} - x^{13} - x^{7} - x^{2} - x - 1$$

$$f = x^{250} + x^{249} - x^{248} - x^{246} + x^{244} - x^{243} - x^{242} + x^{241} - x^{239} + x^{238} - x^{237} - \ldots - x^{17} + x^{16} - x^{15} + x^{14} - x^{13} - x^{12} - x^{9} - x^{8} + x^{6} + x^{5} + x^{4} - x^{3} + x^{2} - 1$$

Then calculate the inverse of $f$ mod $p$ and $f$ mod $q$. The results obtained are

$$f_p^{-1} = x^{250} + x^{249} + x^{247} + x^{245} + x^{244} + x^{243} - x^{241} - x^{240} - x^{239} + x^{236} - x^{235} + x^{234} - \ldots + x^{22} + x^{21} + x^{19} - x^{18} + x^{17} + x^{15} - x^{14} + x^{10} + x^{8} + x^{7} + x^{6} - x^{5} + x^{4} - x^{3} - x$$

$$f_q^{-1} = -919x^{250} - 141x^{249} + 376x^{248} + 556x^{247} + 275x^{246} + 150x^{245} + 201x^{244} + \ldots + 249x^{9} - 199x^{8} + 805x^{7} + 384x^{6} + 216x^{5} + 864x^{4} + 819x^{3} - 696x^{2} + 686x + 1019$$

The last step is the public key calculation

$$h = 1013*x**250 + 684*x**249 + 737*x**248 + 355*x**247 - 113*x**246 - 991*x**245 - 652*x**244 + 473*x**243 - \ldots - 571*x**8 - 531*x**7 - 414*x**6 - 466*x**5 + 847*x**4 - 214*x**3 - 926*x**2 - 1014*x + 718$$

The result of generating the NTRU key is stored in a python file, as shown in Fig. 3, and is used in the signing and verifying process as input.


**Fig. 3.** NTRU Key File

NTRU Signing: The signing process requires the input of a certificate file with extension pdf and a key file with extension npz. The message in the certificate file will be hashed with the SHA-512 hash function and produce a message digest. These results are then encrypted using the NTRU algorithm with completion $e \equiv r * h + m_i \ (mod\ q)$ so that the results are in the form of ciphertext. The resulting ciphertext will be stored in a file and the address of the file is then used as data in QR Code generation, by utilizing the QR Code module from Python and acts as a signature.

To be clear, assume that the message is already in the form of polynomial.

$$m = x**509 + x**507 + x**506 + x**505 + x**504 + x**503 + x**502 + x**499 + x**498 + x**493 + x**490 + x**489 + \ldots + x**38 + x**36 + x**33 + x**31 + x**28 + x**26 + x**14 + x**12 + x**11 + x**7 + x**6 + x**5 + 2*x + 2$$

Then calculate a random polynomial r of degree 251. Assume that the result is:

$$r = x**202 + x**186 + x**183 + x**180 + x**176 + x**169 + x**167 + x**163 + x**151 + x**146 + x**135 + x**127 + x**126 + x**123 + x**121 + x**114 + x**112 + x**109 + x**107 + x**100 + x**97 + x**94 + x**93 + x**92 + x**88 + x**84 + x**83 + x**82 + x**80 + x**75 + x**67 + x**66 + x**64 + x**60 + x**59 + x**58 + x**50 + x**46 + x**45 + x**38 + x**37 + x**18 + x**16 + x**13 + x**5$$

Based on the encryption formula, the encrypted message value is obtained with the public key in the previous example.

$$e = -39*x**250 + 981*x**249 + 124*x**248 - 90*x**247 + 238*x**246 - 129*x**245 - 147*x**244 - 217*x**243 - \ldots - 173*x**9 + 808*x**8 - 981*x**7 - 4*x**6 + 952*x**5 + 962*x**4 + 472*x**3 + 935*x**2 + 185*x - 946$$

The final result of the signature process (with parameters N = 587, $p = 3$, and $q = 2048$) is a pdf certificate file, as shown in Fig 4.


**Fig. 4.** Signatured Certificate File

During the signing process, the data, which was originally a plaintext message, was processed to become a signature. An example of the process of changing the data can be seen in Table 2.

Table 2. Data Changes During NTRU Signing

| Process | Results |
|---|---|
| Read the contents of the certificate file | CERTIFICATE OF TRAINING<br>This certifies that Tengiz Kharatishvili has successfully completed training in video publishing<br>Rowan Murphy, Sr. Videographer<br>June 04, 20XX<br>Date |
| Hashing messages with SHA – 512 | b0f741a08c914065d146114e6f946b50449ca05a87753cf014cb572eec68dbb5172457fa9e49a9f188a1a377ed9047eb1be64d1d61c27f4327fc56f89d85fba4 |
| Converting message to polynomial form | Poly(x**511 + x**510 + x**509 + x**508 + … + x**2 + 2*x + 2, x, domain='ZZ') |
| Encryption result | Poly(38*x**586 - 28*x**585 - … - 26*x**3 - 38*x**2 - 29*x + 9, x, domain='ZZ' ) |
| Save the encryption result into file | encrypted_220512_191310.npz |
| Generate QR-Code | |

Verifying NTRU: The verification process is initiated by inputting the signed certificate file and key file. The certificate file will then be carried out by two different processes, namely (a) the process of hashing the contents of the certificate file, which produces a message digest, and (b) the process of decrypting the ciphertext using the completion of the NTRU algorithm $d \equiv f_p{}^{-1} * [f * e]_q \ (mod\ p)$. The decryption process can be done by first decoding the QR Code to obtain the ciphertext file address.

More specifically, take the values of *e*, *f*, and *fp*-1 using the previous encryption calculations. Then use the private key f to calculate the value of d with the formula until you get the result:

$$d = x**250 + x**248 + x**247 + x**244 + x**242 + x**241 + x**240 + x**239 + x**238 + x**236 + x**234 + x**233 + \ldots + x**22 + x**21 + x**17 + x**14 + x**13 + x**12 + x**11 + x**9 + x**8 + x**6 + x**5 + x**4 + x**3 + x**2$$

An example of the process of changing the data can be seen in Table 3.

**Table 3.** Data Changes During NTRU Verifying

| Process | File | Signature |
|---|---|---|
| Processed data | CERTIFICATE OF TRAINING<br>This certifies that<br>Tengiz Kharatishvili has successfully completed training in video publishing<br>Rowan Murphy, Sr. Videographer<br>June 04, 20XX<br>Date | QR Code Decode Result: encrypted_220512_191310.npz |

| | Message Digest: | Decryption Result: |
|---|---|---|
| Results | b0f741a08c914065d<br>146114e6f946b5044<br>9ca05a87753cf014cb<br>572eec68dbb517245<br>7fa9e49a9f188a1a37<br>7ed9047eb1be64d1<br>d61c27f4327fc56f89<br>d85fba4 | b0f741a08c914065<br>d146114e6f946b50<br>449ca05a87753cf01<br>4cb572eec68dbb51<br>72457fa9e49a9f188<br>a1a377ed9047eb1b<br>e64d1d61c27f4327f<br>c56f89d85fba4 |
| Verification result | Valid Certificate | |

### 5.3. ELGAMAL IMPLEMENTATION OF THE DIGITAL SIGNATURE SCHEME

**Elgamal Key Generation**: Elgamal's algorithm uses a parameter in the form of a key size, which will later be used to determine the positive prime number p and the integer $q$, which is the primitive root of $p$. More specifically, for example, a low parameter will be chosen so that calculations can be carried out easily. The selected parameter is a key length of 1024 bits.

Determine the positive prime number $p$ and the primitive root integer (of $p$) $g$, as follows:

*p :*

*12489857221665811157220874083547072009042450549499155485955947034777566992936482827403733844778047412215234604025630607003873323702143405841378919941167032148636454799401302956338429661580291454927374628123190632069773463945113846463535827792773210978304435023477920819130196803951971300258281822235019146787*

*g :*

*63418710995387172241403192069805538197164554016167091263919199424389687867175103116405042351638104831030066812158610889170947692265070654665995197920497500022013878682010454919367498931390498515528551610334103183863686102471492733297900719100600995239227544949867236010406029917652700377892076852541591702248*

After that, a random number x that meets the conditions can be determined:

*x:*

*91491006234848614245179977624682077925335318344922709975556087966952052496013141292670470834735786780210541158300926247902230874565092117625576634443276708565512300768601795841838503235296900503438256131228239255238826117592836186931503759908005375323734223307625329561031993013698760733136076161345164483 07*

Then calculate the value of the public key y with the formula until its get the result:

*y:*

*65135285886376233423156589926619645695596911824928693298285511568454874973191493243330433838732322620497760174310043241085102670692205950165076922093086441318339707278937587061797663872550861404298605449593244319051941058320010389322519896297712589672532523234837800990051284605923946962568451274849688823 32*

The result of the Elgamal key generation is stored in a python file, as shown in Fig. 5, and is used in the signing and verifying process as input.


**Fig. 5.** Key File

**Elgamal Signing**: The signing process requires the input of a certificate file with a pdf extension and a key file with npz extension. Messages in the certificate file are hashed with SHA-512 and generate a message digest. These results are then encrypted using the Elgamal algorithm with completion $a = g^k \pmod p$ and $b = y^k m \pmod p$.

So that the results are in the form of ciphertext. The resulting ciphertext will be stored in a file and the address of the file is then used as data in QR Code generation by utilizing the QRcode module from Python and acts as a signaturere.

More specifically, for example, assume that the message you want to send is already in the form of an integer:

*m:*

*[14538442250038144105231320892152646169614843557636457374901122892002070321499286938540856146613390051004182394475048657577252138832684349648204124476643815169212696928541858932923968677050098169102877787229500894085499216436358018904406817756050209176272038926760434163580356302597168837638856495582818655, 27705160943300941789610854972146584381060479022655085427330167228125638887616690918970704138783319628359874481019133326273393774170300068315654083835662344198168585128631502674887272242620366028390917958811935679447774090212292845437859079857887075774843267331850791258686344274144911278596297922552299521 01, 51]*

Then determine the random number k:

*k:*

*41359562320441767377634283288850322412547276803159461149548477024128442005584475007172779670001604162003715286024328117858087334908022747810832477109268627896844574612695853240981237510154371251553070029152566278109253481414377771793265148953331162313443731795588514539317270078899192970341452959520836779932 1*

By using the values of $g$, $p$, and $y$ obtained in the previous generate key, then calculating the values of $a$ and $b$ with the formula until the following results are obtained.

*a:*

*70479883181290373076759284085036204725930413600173204582302048923766707934263351040953786274254348456281394440796452980960199763720656329447552526439670077738634270971906420126270709077063852313937425705610528258510766266386304210223963436311307917371153782572535907053966206171964146582994168138422591559688*

*b:*

*38146908137094241042048652122577800334802193925017266170183760286767827207220163037256842810239840048015914643159927683135244502378470278563399040005854722260201849018724348577119582737543234061937118153287164993158A39017723232940081333614556789700270444823795050233554392401223836825760672707159682353497160*

The final result of the signature process is a pdf certificate file, as shown in Fig. 6.


**Fig. 6.** Signatured (Elgamal) Certificate File

During the signing process, the data, which was initially a plaintext message, was processed to become a signature. An example of the process of changing the data can be seen in Table 4.

**Table 4.** Data Changes During Elgamal Signing

| Process | Results |
|---|---|
| Read the contents of the certificate file | CERTIFICATE OF TRAINING<br>This certifies that Tengiz Kharatishvili has successfully completed training in video publishing<br>Rowan Murphy, Sr. Videographer<br>June 04, 20XX<br>Date |
| Hashing messages with SHA – 512 | b0f741a08c914065d146114e6f946b5044<br>9ca05a87753cf014cb572eec68dbb51724<br>57fa9e49a9f188a1a377ed9047eb1be64d<br>1d61c27f4327fc56f89d85fba4 |
| Convert message digest to integer | 2688241…950207, 2660811…905845, 52 |
| Determine the random number k | 6791467875486908727…724749591809191<br>63072209829196994526 |
| Encryption result | 82875883331547…83931630431734,<br>56002328522083…40235165644727,<br>79852418918538…58497092617061,<br>31666415267698…84650783518716,<br>11618727619395…39185616194775,<br>37640656195438…72911613317981 |
| Save the encryption result into file | keyElgamal_220512_203003.npz |
| Generate QR-Code | |

**Verifying Elgamal**: The verification process begins with inputting a signed certificate file and a key file. The certificate file will then be carried out by two different processes, namely (a) the process of hashing the contents of the certificate file, which produces a message digest, and (b) the process of decrypting the ciphertext using the completion of the NTRU algorithm $m = b(a^x)^{-1} (mod\ p)$. The decryption process can be done by first decoding the QR Code to obtain the ciphertext file address..

More specifically, use the private keys $x$, ciphertext [$a$, $b$], and $p$ from the previous calculation, then calculate the value of $m$ with the formula until the results are obtained

*m:*
*145384422500381441052313208921526461696148435576364573749011228920020703214992869385408561*
*46613390051004182394475048657577252138832684349648204124476643815169212696928541858932923 9*
*68677050098169102877787229500894085499216436358018904406817756050209176272038926760434163 5*
*80356302597168837638856495582851865 5,*
*27705160943300941789610854972146584381060479022655085427330167228125638887616690918970704 1*
*38783319628359874481019133326273393774170300068315654083835662344198168585128631502674887 2*
*72242620366028390917958819356794477740902122928454378590798578870757748432673318507912586 8*
*63442741449112785962979225522995210 1*

An example of the process of changing the data can be seen in Table 5.

**Table 5.** Data Changes During Elgamal Verifying

| Process | File | Signature |
|---|---|---|
| Processed data | CERTIFICATE OF TRAINING<br>This certifies that Tengiz Kharatishvili has successfully completed training in video publishing<br>Rowan Murphy, Sr. Videographer<br>June 04, 20XX<br>Date | QR Code Decode Result:<br>keyElgamal<br>_220512<br>_203003.npz |
| Results | Message Digest:<br>b0f741a08c914065d14<br>6114e6f946b50449ca0<br>5a87753cf014cb572ee<br>c68dbb5172457fa9e49<br>a9f188a1a377ed9<br>047eb1be64d1d61c27<br>f4327fc56f89d85fba4 | Decryption Result:<br>b0f741a08c91406<br>5d146114e6f946<br>b50449ca05a877<br>53cf014cb572eec<br>68dbb5172457fa<br>9e49a9f188a1a37<br>7ed9047eb1be64<br>d1d61c27f4327fc<br>56f89d85fba4 |
| Verification result | Valid Certificate | |

## 5.4. COMPARISON OF NTRU AND ELGAMAL ALGORITHM

Key Generation: Longer keys will provide higher security but consume more computational time, so the value of safety and speed will be inversely related. Table 6 shows the running time results for key generation on the Elgamal and NTRU algorithms.

**Table 6.** Running Time of NTRU and Elgamal Key Generation

| Security Level | Algoritma | Trial (seconds) | | | | | |
|---|---|---|---|---|---|---|---|
| | | 1 | 2 | 3 | 4 | 5 | Average |
| Low | NTRU | 97.977 | 95.85 | 95.599 | 95.867 | 114.09 | 99.8766 |
| | Elgamal | 5.913 | 1.324 | 2.698 | 25.212 | 119.116 | 30.8526 |
| Standard | NTRU | 274.4 | 282.074 | 275.681 | 250.869 | 252.186 | 267.042 |
| | Elgamal | 133.610 | 4143.273 | 384.657 | 1712.502 | 282.778 | 1331.364 |
| High | NTRU | 532.952 | 527.532 | 539.384 | 684.279 | 532.874 | 563.4042 |
| | Elgamal | 3875.925 | 6818.925 | 7651.898 | 2647.856 | 2641.153 | 4727.151 |
| Highest | NTRU | 1255.205 | 1154.33 | 1106.095 | 1024.373 | 1026.911 | 1113.3828 |
| | Elgamal | >9 Hours (32400) | >9 Hours (32400) | >9 Hours (32400) | >9 Hours (32400) | >9 Hours (32400) | > 9 Hours (32400) |

From the Table 6, it can be described in graphical form, as shown in Fig. 7.

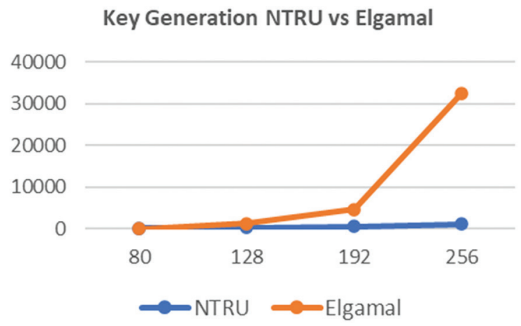


**Fig. 7.** Running Time of NTRU and Elgamal Key Generation

From Fig. 7, the computation time required to generate an NTRU key at a low-security level is 3x slower than Elgamal, but at a standard and high NTRU security level it is almost 5x and 8x faster than Elgamal

**Signs**: The time required to sign the file using the two algorithms is compared to evaluate the performance of the proposed system. The running time of the signing process with the NTRU and Elgamal Algorithms in five trials, results are shown in Table 7.

**Table 7**. Running Time Signing NTRU and Elgamal

| Security Level | Algoritma | Trial (seconds) | | | | | |
|---|---|---|---|---|---|---|---|
| | | 1 | 2 | 3 | 4 | 5 | Average |
| Low | NTRU | 6.969 | 6.242 | 6.138 | 5.967 | 6.242 | 6.3116 |
| | Elgamal | 0.62 | 0.606 | 0.649 | 0.565 | 0.527 | 0.5934 |
| Standard | NTRU | 11.975 | 12.004 | 11.829 | 10.601 | 10.353 | 11.3524 |
| | Elgamal | 2.505 | 2.281 | 2.226 | 2.218 | 2.123 | 2.2706 |
| High | NTRU | 20.529 | 20.401 | 20.5 | 24.45 | 20.725 | 21.321 |
| | Elgamal | 28.23 | 27.072 | 31.958 | 34.558 | 31.604 | 30.6844 |
| Highest | NTRU | 50.471 | 45.614 | 46.38 | 40.799 | 40.158 | 44.6844 |
| | Elgamal | - | - | - | - | - | - |

Table 7 shows graphs of running time for the signing process using the Elgamal and NTRU algorithms, as shown in Fig. 8. Experimental running time on Elgamal with the highest security cannot be carried out due to a failure in the key generation process which cannot generate a public key and a private key, so the process cannot be continued.

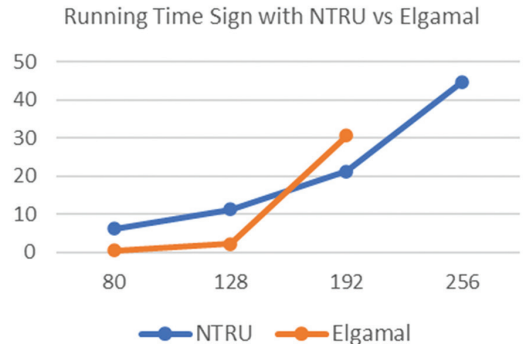


**Fig. 8.** Running Time Signing NTRU and Elgamal

From Fig. 8, the time for signing using Elgamal is faster than NTRU when the experiment is carried out at a low-security level and standard security. At the same time, at a higher security level, namely high safety, and highest security, the signing process using the NTRU algorithm requires faster time if compared to Elgamal's algorithm. So, at a higher level of protection, it can be said that the signing process using the NTRU algorithm is faster and safer than using the Elgamal algorithm. From Fig 8, the computation time required to sign NTRU at a high-security level is almost 1.5x faster than Elgamal.

**Verify**: The NTRU cryptosystem significantly produces faster average speeds than Elgamal when the key size is increased. The time required to sign the file using the two algorithms is compared to evaluate the performance of the proposed system. Table 8 shows the results of running time verifying NTRU and Elgamal

**Table 8**. Running Time Verifying NTRU and Elgamal

| Security Level | Algoritma | Trial (seconds) | | | | | |
|---|---|---|---|---|---|---|---|
| | | 1 | 2 | 3 | 4 | 5 | Average |
| Low | NTRU | 8.578 | 7.815 | 7.538 | 7.619 | 8.121 | 7.9342 |
| | Elgamal | 0.527 | 0.648 | 0.555 | 0.54 | 0.531 | 0.5602 |
| Standard | NTRU | 20.843 | 20.985 | 22.962 | 19.06 | 18.214 | 20.4128 |
| | Elgamal | 3.614 | 3.610 | 3.458 | 3.443 | 3.509 | 3.5268 |
| High | NTRU | 39.782 | 39.462 | 40.5 | 50.229 | 40.109 | 42.0164 |
| | Elgamal | 64.338 | 70.588 | 59.873 | 61.728 | 61.132 | 63.5318 |
| Highest | NTRU | 101.122 | 83.753 | 84.22 | 77.998 | 79.882 | 85.395 |
| | Elgamal | - | - | - | - | - | - |

Table 8 shows graphs of running time Verifying NTRU and Elgamal Cryptosystem, as shown in Fig. 9.

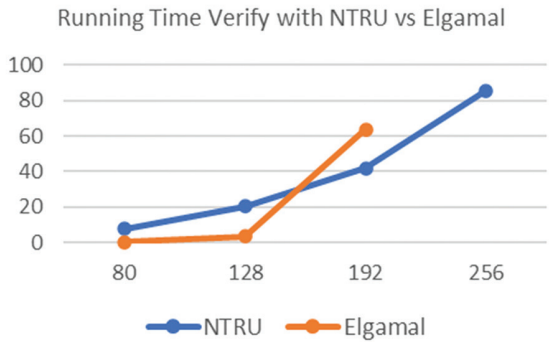


**Fig. 9.** Running Time Verifying NTRU and Elgamal

From Fig. 9, the verification process with the Elgamal algorithm for low-security levels, Elgamal is superior to NTRU. However, the security level is at a higher security level the NTRU is found to be faster than the Elgamal algorithm.

### 5.5. DIGITAL SCHEMATIC TESTING

NTRU: The test is carried out by running a digital signature scheme using the NTRU algorithm, against the same certificate file, with the final result as an "Invalid" or "Valid" certificate statement. It can be shown in Table 9.

Elgamal: The test is carried out by running a digital signature scheme using the Elgamal algorithm, against the same certificate file, with the final result in the form of an "Invalid" or "Valid" certificate statement. . It can be shown in Table 9.

Table 9 shows that in the Elgamal scheme, the verification results on certificates without changes at all security levels have "Valid" verification results, and certificates with changes have "Invalid" results. In the NTRU scheme, the verification results on the certificate with changes in all N-parameter tests result “Invalid”, this happens because a difference in the contents of the certificate will result from a new message digest m (from the hashing process) which during the comparison process the value of m and m’ (message digest from the decryption process) will be different and cause the verification results to be invalid.

**Table 9.** Testing the Digital Signature Scheme with NTRU

| Security level (bits) | NTRU | | | Elgamal | | |
|---|---|---|---|---|---|---|
| | N | Certificate Verification Results (Without Changes) | Certificate Verification Results (With Changes) | bits | Certificate Verification Results (Without Changes) | Certificate Verification Results (With Changes) |
| 80 | 251 | Invalid | Invalid | 1024 | Valid | Invalid |
| 128 | 397 | Invalid | Invalid | 3071 | Valid | Invalid |
| 192 | 587 | Valid | Invalid | 7680 | Valid | Invalid |
| 256 | 787 | Valid | Invalid | 15360 | - | - |

However, Table 9 shows that the results of certificate verification without changes with parameter values NTRU N – 587 and N-787 show "Valid" results. The verification results do not change because the length of the polynomial ring $R$ can include the size of the original message polynomial with the highest degree of 511. When decrypted, the decrypted polynomial will not be truncated because of the polynomial ring rule $R$. So that from the beginning to the end of the decryption process, the polynomial length of the processed message will not be truncated and intact. Meanwhile, the N – 251 and N -397 tests show "Invalid" results, which can occur due to the use of the SHA-512 hash function and the $N$ parameter value that affects the length of the polynomial ring $R = \mathbb{Z}[X]/(X^N - 1)$. When the original message is hashed with SHA-512, which is then converted into polynomial form, the resulting polynomial will have a maximum length of 511 degrees. Meanwhile, during the encryption and decryption process, the size of the polynomial will follow the rules $R = \mathbb{Z}[X]/(X^N - 1)$, If the N parameter value used is 251, then the highest degree of the applicable polynomial is N-1 or 250. When the original message polynomial has the highest degree of 511 while the message decryption polynomial only has the highest degree of 250 because it follows the ring polynomial $R$. Then the message digest generated from the decryption process is different from the message digest from hashing the original message. So that the verification results show the same "Invalid" results, namely "Invalid" for the same reason.

## 6. CONCLUSION

From the problems encountered, the proposed problem-solving solutions, as well as the experiments carried out. It can be concluded that with the application of the NTRU and Elgamal Algorithms in the digital signature scheme, based on the comparison of running time in the key generation, sign, and verify processes, it shows that when the security level is low, NTRU is slower than Elgamal, but at the high-security level, NTRU is faster than Elgamal, which is 1.4x faster in the signing process and 1.5x faster in the verification process. So it can be said that NTRU, at a higher level of security, has faster when compared to Elgamal. In addition, the results of the certificate verification test with NTRU and Elgamal on the digital signature have been tested to be safe. This is proven by the testing process where data modification is carried out in the certificate document, and the program manages to find out and shows the results "Invalid ". In the certificate document without modification, the program shows the result Valid", but this result does not apply to NTRU N-251 and NTRU N-397 and shows the result "Invalid," which should be "Valid" this can happen because of the role of bit length The SHA used is SHA-512.

## 7. REFERENCES:


[1] N. Yanti et al. Implementation of Advanced Encryption Standard (AES) and QR code algorithm on digital legalization system. in The 3rd International Conference on Energy, Environmental and Information System. Semarang August 14-15, Vol 73 2018. EDP Sciences.

[2] M. Kang, V. A. Lemieux, "decentralized identity-based blockchain solution for privacy-preserving licensing of individual-controlled data to prevent unauthorized secondary data usage", Ledger. Vol 6, pp: 126-151

[3] A. Hakami, A. Al-Omary, "Secure Transaction Framework based on Encrypted One-time Password and Multi-factor", Proceedings of the International Conference on Data Analytics for Business and Industry, Bahrain, 25-26 October 2021, pp. 677-682.

[4] R. Bernardini, "Cryptography - Recent Advances and Future Developments", IntechOpen, 2021.

[5] A. Mittelbach, M. Fischlin, "The Theory of Hash Functions and Random Oracles : An Approach to Modern Cryptography", 1st Edition, Springer-Verlag Berlin Heidelberg 2021:

[6] H. Mukhtar, "Kriptografi Untuk Keamanan Data" Edisi pertama, Deepublish, Yogyakarta, 2018.

[7] R. Munir, "Kriptografi" Edisi Kedua", Bandung: Informatika, 2019.

[8] J. Zhou et al. "Applied cryptography and network security workshops", Springer-Verlag Berlin Heidelberg, 2021.

[9] R. Chaudhary et al. "Lattice-based public key cryptosystem for internet of things environment: Challenges and solutions", IEEE Internet of Things Journal, Vol. 6, No 3, 2018. pp. 4897-4909.

[10] G. Mittal, S. Kumar, S. Kumar, "Novel public-key cryptosystems based on NTRU and algebraic structure of group rings". Journal of Information and Optimization Sciences, Vol. 42, No. 7, 2021. pp 1507-1521.

[11] H. R. Yassein, A.A. Abidalzahra, N. M. Al-Saidi, "A new design of NTRU encryption with high security and performance", Proceedings of the 4th International Conference of Mathematical Sciences, Istanbul, Turkey, 17-21 June 2020, p. 080005

[12] K. Daimi et al., "Computer and network security essentials", Springer Verlag, Berlin Heidelberg, 2018.

[13] A. K. Sharma, S. Mittal. "Cryptography & network security hash function applications, attacks and advances: A review", Proceedings of the 3rd International Conference on Inventive Systems and Control, Coimbatore, India, 10-11 January 2019, pp. 177-188.

[14] B. A. Forouzan, D. Mukhopadhyay, "Cryptography and network security", Mc Graw Hill Education (India) Private Limited New York, NY, USA, 2015.

[15] S. Ghosh, S. Sampalli, "A survey of security in SCADA networks: Current issues and future challenges", IEEE Access,Vol 7, 2019, pp 135812-135831.

[16] E. V. Waruwu, N. B. Nugroho, F. Sonata, "Penerapan Digital Signature Menggunakan Metode RSA Untuk Verifikasi Surat Keterangan Keaslian Ijazah SMA Swasta Bina Artha", Jurnal Cyber Tech, Vol. 1, No. 1, 2021. pp. 37-47.

[17] Nuraeni, F., Y.H. Agustin, and I.M. Muharam, "Implementasi Tanda Tangan Digital Menggunakan RSA dan SHA-512 Pada Proses Legalisasi Ijazah", Proceedings of the Konferensi Nasional Sistem Informasi, Pangkalpinang, 8-9 March 2018, pp. 864-869.

[18] B. Triand et al. "Digital Document Security on Legalize Higher Education Diplomas with Digital Signature and SHA-1 Algorithm", Proceedings of the 7th International Conference on Cyber and IT Service Management, 6-8 November 2019, pp. 1-5.

[19] P. A. W. D. Putro, "Arumsari. Designing and Building Disposition-EL Application by Applying AES-256 and RSA-2048", Proceedings of the International Conference on Informatics, Multimedia, Cyber and Information System, 24-25 October 2019, pp. 163-168.

[20] T. Yuniati, M. F. Sidiq, "Literature Review: Legalisasi Dokumen Elektronik Menggunakan Tanda Tangan Digital sebagai Alternatif Pengesahan Dokumen di Masa Pandemi". Jurnal RESTI (Rekayasa Sistem Dan Teknologi Informasi), Vol. 4 No. 6, 2020, pp. 1058-1069.

[21] W. Pramusinto et al. "Implementation of AES-192 Cryptography and QR Code to Verify the Authenticity of Budi Luhur University Student Certificate", Jurnal Pendidikan Teknologi Kejuruan, Vol 3, No. 6, 2020, pp. 209-215.

[22] J. Katz, Y. Lindell, "Introduction to modern cryptography", CRC press, 2020.

[23] J. S. Kraft, L.C. Washington, "An introduction to number theory with cryptography", Chapman and Hall/CRC, 2018.

[24] A. J. Menezes, P. C. Van Oorschot, S.A. Vanstone, "Handbook of applied cryptography", CRC press, 2018.

[25] X. Yu, "Design of Aerospace QR Ticketing System Based on Mobile Devices", Proceedings of the 4th International Conference on Information Systems and Computer Aided Education, 24 September 2021, pp.2285-2287.

[26] A. I. Chowdhury, M. S. Rahman, N. Sakib, "A study of multiple barcode detection from an image in business system", International Journal of Computer Applications, Vol. 181, No. 37, 2019, pp. 30-37.

[27] Z. Azuan et al. "Mobile Advertising via Bluetooth and 2D Barcodes", Proceedings of the International Conference on Data Engineering 2015, Singapore, 10 August 2019. pp. 443-456.

[28] E. Hari Charan, et al. "Electronic toll collection system using barcode technology in Nanoelectron-

ics, Circuits and Communication Systems", Singapore, 2 August 2018. pp. 549-556.

[29] D. D. Vo et al. "Barcode Image Restoration for Recognition of Product Information", European Journal of Engineering and Technology Research, Vol.4, No. 9, 2019. pp. 93-100.

[30] R. Focardi, F. L. Luccio, H. A. Wahsheh, "Usable security for QR code", Journal of information security and applications, Vol. 48, No. 1, 2019, p. 102369.

[31] R. Palomäki, "A distance-aware 2D barcode for mobile computing applications", Aalto University School of Science, Communication and Information Science, Finland, Master Thesis, 2018.

[32] N. G. Kaziyeva, G. Kukharev, Y. Matveev. "Barcoding in biometrics and its development", Proceedings of the International Conference on Computer Vision and Graphics, Warsaw, Poland, 17 September 2018, pp. 464-471.

[33] C. Guo, C.-C. Chang, S.-C. Chang, "A Secure and Efficient Mutual Authentication and Key Agreement Protocol with Smart Cards for Wireless Communications", International Journal of Network Security, Vol.20, No.2, 2018, pp. 323-331.

[34] M. Qi, J. Chen, Y. Chen, A "secure authentication with key agreement scheme using ECC for satellite communication systems", International Journal of Satellite Communications and Networking, Vol. 37, No. 3, 2019, pp. 234-244.

[35] M. Jiaqing, Z. Hu, H. Chen, W. Shen, "An Efficient and Provably Secure Anonymous User Authentication and Key Agreement for Mobile Cloud Computing", Wireless Communications and Mobile Computing, Vol. 2019, p. 4520685.

[36] H. Loriya, A. Kulshreshta, D. Keraliya, "Security analysis of various public key cryptosystems for authentication and key agreement in wireless communication network", International Journal of Advanced Research in Computer and Communication Engineering, Vol. 6, No. 2, 2017, pp. 267-274.

[37] A. P. Siahaan, B. O. Elviwani, B. Oktaviana, "Comparative analysis of rsa and elgamal cryptographic public-key algorithms", Proceedings of the Joint Workshop KO2PI and The 1st International Conference on Advance & Scientific Innovation, 4 July 2018, pp. 162-171.