Journal of Artificial Intelligence and Engineering Applications

Website: https://ioinformatic.org/

15th October 2025. Vol. 5. No. 1; e-ISSN: 2808-4519

Implementation of Number Theoretic Transform Unit (Ntru) Cryptography to Secure Text Files

Arta Naila Fanaya[1]*, Rahmadani[2] , Marto Sihombing[3]

[1,2,3]*Kaputama STMIK*

artanailafanaya12@gmail.com [1*], rahm4dani@gmail.com [2], martosihombing45@gmail.com [3]

Abstract

This research aims to develop a text file security system using the NTRU cryptographic algorithm. This process begins with understanding the basic principles of NTRU, which uses a polynomial-based mathematical structure to generate key pairs. The public key is used to encrypt text files, while the private key is used to decrypt them. The research stages include generating public-private keys, followed by the process of encrypting text files into *ciphertext* and decrypting to restore the files to their original form. The result of this research is a system capable of effectively securing text files using NTRU encryption. This system is expected to contribute to improving digital information security, especially in protecting the confidentiality of text files. This research report will provide insights and detailed steps regarding the implementation of NTRU. This research is expected to be a guide for developers and researchers in implementing NTRU to strengthen broader data security.

Keywords: *Cryptography, NTRU, Data Security, Encryption, Text Files*

1. Introduction

Text file security in web applications faces complex challenges due to increasing data leaks, the continued prevalence of web application vulnerabilities, the high value of sensitive data in text files, and technological developments that have the potential to undermine conventional cryptographic systems. Therefore, a comprehensive security approach involving strong encryption, strict access control, and continuous security monitoring is required. Given these threats, a cryptographic algorithm is needed that can secure data effectively and efficiently. Although various cryptographic algorithms have been widely used, they have significant weaknesses. These limitations present a challenge for web application developers who require solutions that are not only secure but also efficient and easy to implement.

As a solution to existing data security issues, this study implements the NTRU cryptographic algorithm to protect text files. This process begins with understanding the basic principles of NTRU encryption, which uses a polynomial-based mathematical structure and modulus to generate public and private keys. In the initial stage, keys are generated through the NTRU algorithm. This uses a public key to encrypt a text file, while a private key is used to decrypt it. The securely embedded text file then undergoes an encryption process where the original data is converted into a form that cannot be read without the appropriate key. This process involves selecting appropriate parameters to ensure security and efficiency, such as the key size and the complexity of the polynomial used. NTRU also offers a strong level of security thanks to its *lattice- based structure* , which makes it resistant to attacks. This analysis is expected to provide recommendations for the use of NTRU in various contexts and contribute to the development of secure and efficient encryption solutions.

The NTRU algorithm is an asymmetric algorithm. Asymmetric algorithms have different keys during the encryption and decryption process, namely the public key and the private key. The public key is a key that is published and can be known to everyone, while the private key is a key that is kept secret and can only be understood by one person. The security level of the NTRU algorithm lies in the use of polynomials during the operation process, as well as the difficulty of finding short vectors from a lattice [1] . According to [2] NTRU is an open-source public-key cryptosystem that uses lattice-based cryptography to encrypt and decrypt data. Unlike other popular public-key cryptosystems, this system is resistant to attacks using Shor's Algorithm and its performance has been proven to be much better. According to [3] NTRU is based on a certain polynomial ring algebraic structure.

The implementation of NTRU cryptography in this study will follow several key stages. First, the generation of a public and private key pair. NTRU works by utilizing a truncated polynomial ring, which allows for efficient mathematical operations. The key generation process involves selecting three polynomials, namely f , g , and h , with f and g as the private keys and h as the public key. After the keys are generated, the text file is encrypted with the public key *H.* This encryption process involves multiplying the message polynomial by the public key, followed by a modular operation. Next, the resulting *ciphertext is decrypted with the private keys F* and *G.* to restore the

message to its original *plaintext form* . Examining various NTRU parameters, such as polynomial degree and modulus, to optimize system performance and security according to the specific requirements of text file security.

This research produces a system that can secure text files using NTRU encryption. This system is expected to contribute to improving digital information security, particularly in protecting the confidentiality of text documents. This study will provide a detailed report on the implementation of the NTRU algorithm to protect text files. This report includes the technical steps for generating public and private keys, as well as the encryption and decryption steps. Thus, this report is expected to provide practical insights and recommendations for developers and researchers who wish to apply NTRU in a broader data security context.

2. Research methodology

To assist in the preparation of this research, a clear framework with clear stages is needed. This framework outlines the steps to be taken to resolve the problem under discussion. The research methods used are :

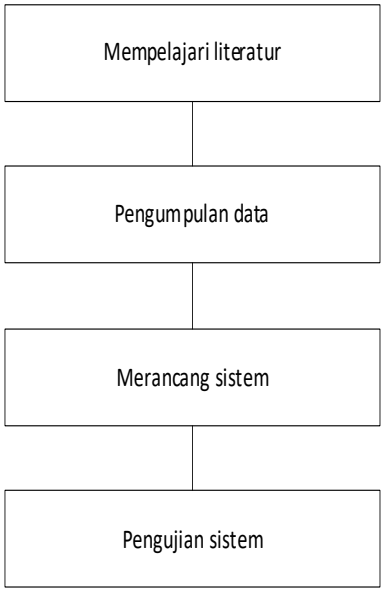


Fig. 1: Research Methodology

From the flow diagram in the image above, the stages of this research can be described as follows: following:

1. Studying Literature: At this stage, a search is carried out for theoretical foundations obtained from various books and the internet to complete the vocabulary of concepts and theories, so that it has a good and appropriate foundation and knowledge.
2. Data Collection: Data collection by using or collecting written sources , by reading, studying and noting down important things from books, journals and the internet that are related to the problem being discussed in order to obtain a theoretical overview.
3. Designing the System: At this stage, the system is designed using *Python 3* with *the Django framework.*
4. System Testing: At this stage, the previously designed application has been completed and the system testing stage is carried out to determine whether there are *any errors* or damage to the system that has been designed.

3. Results and Discussion

3.1. Discussion

A Number Theoretic Transform Unit (NTRU) cryptographic implementation designed to secure text files. The goal of this implementation is to address data security challenges, particularly text files that are vulnerable to leaks. Based on testing results, the system successfully encrypted and decrypted text files .

3.2. Implementation

This research aims to implement a system to secure text files using NTRU cryptography. Based on a structured methodology, the system was successfully implemented using the Python 3 programming language with the Django *framework* . The implementation of Number Theoretic Transform Unit (NTRU) cryptography is carried out to secure text files with three main stages. First, key generation, where a public and private key pair is generated from a polynomial structure. The public key is used to encrypt messages, while the private key is used to decrypt them. Second, encryption, where the original text file is converted into *ciphertext* through a polynomial multiplication

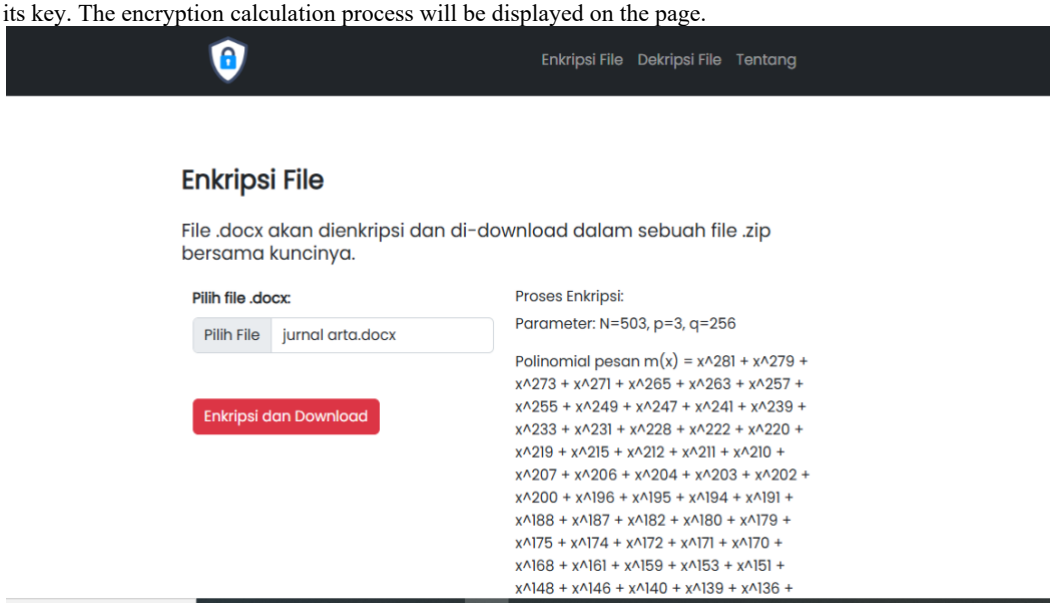
process with the public key. Finally, decryption, where *the ciphertext* is returned to *plaintext* (the original text) using the generated private key.

3.3. System Trial

After implementing the system, the next step is to test the developed system. This testing aims to ensure that each feature in the system functions correctly and as expected. The following are the results of the testing: Account Registration Test

1. Encryption Process Trial

encryption process trial is the testing stage for To ensure that the NTRU cryptographic system can take a text file and convert it into a secure, unreadable encrypted form. The user can select the file to be encrypted, in this example the arta.doc journal file. The system will then process the file to be encrypted. Once the process is complete, a .zip file will be downloaded containing the encrypted file and its key. The encryption calculation process will be displayed on the page.



x^134 + x^132 + x^131 + x^129 + x^128 + x^127 + x^124 + x^123 + x^120 + x^118 + x^116 + x^115 + x^113 + x^112 + x^108 + x^107 + x^105 + x^102 + x^100 + x^99 + x^98 + x^96 + x^92 + x^91 + x^89 + x^86 + x^84 + x^83 + x^78 + x^76 + x^75 + x^73 + x^72 + x^71 + x^67 + x^60 + x^59 + x^57 + x^54 + x^52 + x^51 + x^49 + x^48 + x^47 + x^44 + x^43 + x^41 + x^38 + x^35 + x^28 + x^27 + x^23 + x^20 + x^19 + x^16 + x^14 + x^12 + x^11 + x^10 + x^7 + x^4 + x^3 + x^2 + 1

Polinomial acak r(x) = x^502 + x^501 − x^490 − x^482 + x^478 − x^470 − x^468 + x^467 − x^465 − x^464 + x^454 − x^452 − x^439 + x^420 − x^419 + x^414 − x^412 − x^409 − x^408 + x^397 + x^394 − x^390 + x^385 − x^374 + x^369 − x^368 − x^362 − x^361 − x^359 + x^358 + x^354 − x^353 − x^350 − x^345 − x^344 + x^343 + x^342 − x^336 − x^333 + x^326 − x^325 − x^324 − x^316 + x^315 + x^314 − x^313 − x^306 + x^304 − x^301 − x^298 − x^289 − x^286 − x^274 − x^272 + x^271 − x^269 − x^268 + x^264 + x^260 − x^252 + x^245 + x^244 +

x^127 - x^118 + x^116 - x^111 - x^86 - x^82 -
x^76 + x^70 + x^68 + x^67 + x^66 - x^55
+ x^50 + x^44 + x^38 + x^33 + x^26 +
x^25 + x^20 - x^18 + x^9 + x^4 - 1

Kunci publik h(x) = 105*x^502 - 70*x^501
- 58*x^500 + 36*x^499 - 104*x^498 +
113*x^497 + 69*x^496 + 87*x^495 -
134*x^494 - 39*x^493 - 28*x^492 +
46*x^491 - 5*x^490 + 119*x^489 - x^488 +
10*x^487 + 135*x^486 + 23*x^485 -
100*x^484 + 127*x^483 + 11*x^482 -
28*x^481 - 63*x^480 + 190*x^479 +
49*x^478 - 2*x^477 + 123*x^476 -
96*x^475 + 80*x^474 - 155*x^473 -
44*x^472 - 47*x^471 + 66*x^470 +
115*x^469 + 108*x^468 - 114*x^467 -
78*x^465 - 180*x^464 + 111*x^463 -
97*x^462 - 98*x^461 - 54*x^460 -
16*x^459 + 171*x^458 - 69*x^457 -
21*x^456 + 30*x^455 + 74*x^454 -
116*x^453 + 13*x^452 + 33*x^451 -
63*x^450 - 51*x^449 + 133*x^448 +
115*x^447 + 178*x^446 - 113*x^445 -
204*x^444 + 142*x^443 + 49*x^442 +

3*x^26 + 26*x^25 - 116*x^24 + 44*x^23 +
68*x^22 - 34*x^21 - 96*x^20 + 52*x^19 -
169*x^18 + 217*x^17 - 209*x^16 - 167*x^15 -
51*x^14 - 189*x^13 + 207*x^12 - 155*x^11 -
32*x^10 + 36*x^9 + 183*x^8 + 50*x^7 +
4*x^6 + 141*x^5 + 150*x^4 + 88*x^3 -
115*x^2 - 32*x - 6

Mengalikan r(x) dengan h(x):

r(x) * h(x) = 105*x^1004 + 35*x^1003 -
128*x^1002 - 22*x^1001 - 68*x^1000 +
9*x^999 + 182*x^998 + 156*x^997 -
47*x^996 - 173*x^995 - 67*x^994 +
18*x^993 - 64*x^992 + 184*x^991 +
176*x^990 - 27*x^989 + 249*x^988 +
45*x^987 - 146*x^986 - 60*x^985 +
167*x^984 + 92*x^983 - 5*x^982 +
45*x^981 + 453*x^980 - 255*x^979 -
5*x^978 - 34*x^977 - 121*x^976 +
54*x^975 - 2*x^974 - 177*x^973 -
226*x^972 + 121*x^971 + 212*x^970 -
21*x^969 - 211*x^968 - 292*x^967 -
246*x^966 - 266*x^965 + 258*x^964 +
100*x^963 + 95*x^962 - 409*x^961 -
111*x^960 - 185*x^959 - 159*x^958 +

- 625*x^40 + 56*x^39 - 52*x^38 +
252*x^37 + 3*x^36 - 665*x^35 + 292*x^34
+ 21*x^33 + 335*x^32 + 457*x^31 +
22*x^30 + 532*x^29 + 132*x^28 -
364*x^27 + 65*x^26 - 184*x^25 - x^24 -
96*x^23 - 691*x^22 + 338*x^21 - 159*x^20
- 219*x^19 + 160*x^18 - 223*x^17 +
466*x^16 + 16*x^15 + 160*x^14 + 375*x^13
+ 64*x^12 + 90*x^11 + 4*x^10 + 99*x^9 -
33*x^8 + 38*x^7 - 119*x^6 - 173*x^5 -
156*x^4 - 88*x^3 + 115*x^2 + 32*x + 6

Reduksi hasil perkalian (mod q = 256):

(r(x) * h(x)) mod 256 = 105*x^1004 +
35*x^1003 + 128*x^1002 - 22*x^1001 -
68*x^1000 + 9*x^999 - 74*x^998 -
100*x^997 - 47*x^996 + 83*x^995 -
67*x^994 + 18*x^993 - 64*x^992 -
72*x^991 - 80*x^990 - 27*x^989 -
7*x^988 + 45*x^987 + 110*x^986 -
60*x^985 - 89*x^984 + 92*x^983 -
5*x^982 + 45*x^981 - 59*x^980 + x^979 -
5*x^978 - 34*x^977 - 121*x^976 +
54*x^975 - 2*x^974 + 79*x^973 +
30*x^972 + 121*x^971 - 44*x^970 -

56*x^99 + 56*x^98 + 69*x^97 + 37*x^96 −
47*x^95 − 50*x^94 + 23*x^93 + 40*x^92 −
51*x^91 + 105*x^90 − 82*x^89 + 87*x^88 −
81*x^87 + 38*x^86 + 117*x^85 + 57*x^84 −
8*x^83 − 40*x^82 − 46*x^81 + 115*x^80 +
72*x^79 − 50*x^78 + 70*x^77 − 13*x^76 +
26*x^75 + 127*x^74 − 73*x^73 + 33*x^72 −
122*x^71 + 41*x^70 + 89*x^69 − 12*x^68 +
82*x^67 − 126*x^66 − 113*x^65 − 58*x^64 −
29*x^63 − 102*x^62 + 18*x^61 + 80*x^60 +
78*x^59 + 42*x^58 + 69*x^57 − 69*x^56 −
45*x^55 + 126*x^54 − 98*x^53 + 50*x^52
+ 15*x^51 + 19*x^50 + 95*x^49 + 43*x^48
+ 122*x^47 − 49*x^46 − 74*x^45 − 110*x^44
− 69*x^43 + 65*x^42 + 80*x^41 − 113*x^40
+ 56*x^39 − 52*x^38 − 4*x^37 + 3*x^36 +
103*x^35 + 36*x^34 + 21*x^33 + 79*x^32 −
55*x^31 + 22*x^30 + 20*x^29 − 124*x^28 −
108*x^27 + 65*x^26 + 72*x^25 − x^24 −
96*x^23 + 77*x^22 + 82*x^21 + 97*x^20 +
37*x^19 − 96*x^18 + 33*x^17 − 46*x^16 +
16*x^15 − 96*x^14 + 119*x^13 + 64*x^12 +
90*x^11 + 4*x^10 + 99*x^9 − 33*x^8 +
38*x^7 − 119*x^6 + 83*x^5 + 100*x^4 −
88*x^3 + 115*x^2 + 32*x + 6

Menambahkan polinomial pesan m(x):

e'(x) = (r(x) * h(x)) mod q + m(x)

e'(x) = 105*x^1004 + 35*x^1003 +
128*x^1002 − 22*x^1001 − 68*x^1000 +
9*x^999 − 74*x^998 − 100*x^997 −
47*x^996 + 83*x^995 − 67*x^994 +
18*x^993 − 64*x^992 − 72*x^991 −
80*x^990 − 27*x^989 − 7*x^988 +
45*x^987 + 110*x^986 − 60*x^985 −
89*x^984 + 92*x^983 − 5*x^982 +
45*x^981 − 59*x^980 + x^979 − 5*x^978 −
34*x^977 − 121*x^976 + 54*x^975 −
2*x^974 + 79*x^973 + 30*x^972 +
121*x^971 − 44*x^970 − 21*x^969 +
45*x^968 − 36*x^967 + 10*x^966 −
10*x^965 + 2*x^964 + 100*x^963 +
95*x^962 + 103*x^961 − 111*x^960 +
71*x^959 + 97*x^958 + 86*x^957 −
111*x^956 + 44*x^955 − 58*x^954 +
115*x^953 − 118*x^952 − 41*x^951 −
78*x^950 − 25*x^949 − 56*x^948 +
68*x^947 − 123*x^946 + 6*x^945 −
95*x^944 − 65*x^943 − 52*x^942 −

96*x^18 + 33*x^17 − 45*x^16 + 16*x^15 −
95*x^14 + 119*x^13 + 65*x^12 + 91*x^11 +
5*x^10 + 99*x^9 − 33*x^8 + 39*x^7 −
119*x^6 + 83*x^5 + 101*x^4 − 87*x^3 +
116*x^2 + 32*x + 7

Reduksi akhir (mod x^503−1) dan (mod q
= 256):

e(x) = e'(x) mod (x^503−1) mod 256

Hasil Akhir Polinomial Terenkripsi e(x) =
2*x^502 − 126*x^501 + 48*x^500 −
88*x^499 + 68*x^498 + 63*x^497 −
44*x^496 − 95*x^495 + 58*x^494 −
97*x^493 + 120*x^492 − 100*x^491 −
12*x^490 + 39*x^489 + 39*x^488 +
98*x^487 + 108*x^486 + 18*x^485 +
54*x^484 + 122*x^483 + 27*x^482 −
41*x^481 + 33*x^480 + 20*x^479 −
42*x^478 − 26*x^477 − 86*x^476 −
117*x^475 + x^474 + 40*x^473 + 83*x^472
+ 65*x^471 − 82*x^470 − 54*x^469 +
54*x^468 + 55*x^467 + 119*x^466 +
29*x^465 − 70*x^464 + 114*x^463 +
80*x^462 − 73*x^461 + 72*x^460 +

37*x^35 + 125*x^34 + 67*x^33 + 65*x^32 −
14*x^31 − 46*x^30 − 21*x^29 + 13*x^28 +
42*x^27 + 55*x^26 + 73*x^25 + 7*x^24 +
87*x^23 + 32*x^22 − 47*x^21 + 75*x^20 +
95*x^19 − 70*x^18 + 70*x^17 − x^16 +
72*x^15 − 106*x^14 + 34*x^13 + 127*x^12 −
92*x^11 + 61*x^10 − 99*x^9 − 114*x^8 −
45*x^7 − 35*x^6 − 111*x^5 − 41*x^4 −
117*x^3 + 55*x^2 − 126*x + 49

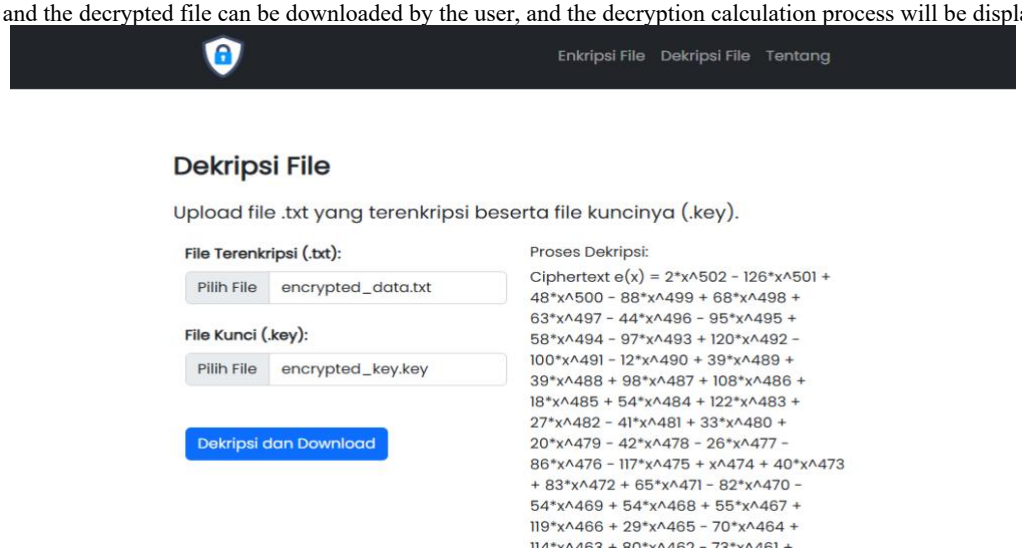
Koefisien e(x) = [2, −126, 48, −88, 68, 63,
−44, −95, 58, −97, 120, −100, −12, 39, 39, 98,
108, 18, 54, 122, 27, −41, 33, 20, −42, −26, −86,
−117, 1, 40, 83, 65, −82, −54, 54, 55, 119, 29,
−70, 114, 80, −73, 72, 122, 107, −29, −118, −39,
−86, 8, 51, −52, −97, 76, −63, 15, 112, 48, 39,
79, −82, 55, −39, 63, −82, 31, −17, 12, −96, −29,
76, −81, −108, −28, 55, −40, 93, 109, 107, 30,
111, 65, −68, 76, −76, −20, −58, 3, 75, −76,
−109, 89, 87, 120, −57, −24, 45, 10, −56, −101,
−105, 31, −62, 102, 71, 8, 101, −30, −83, −54, 116,
52, 62, 127, 97, −94, −68, 108, 27, −77, −84, 16,
117, −122, −79, −14, −42, −124, −29, 95, −88,
−13, −103, 112, 38, 96, 106, 40, −43, −47, −81,
53, −94, −107, −90, 94, −37, −84, 51, −19, −36,

−121, −64, −13, 120, −26, −60, 116, 12, 13, 117, 27,
99, 60, 76, 30, 41, −37, −7, 91, −94, −44, −117,
−4, 105, −3, 18, −12, −80, 60, 15, 16, 82, 7, −105,
63, −86, −63, −14, 84, 16, 18, −25, −79, 114,
−59, −55, 3, −104, −89, 82, 59, −85, 15, −127,
31, −76, −18, −65, −94, 126, 73, −16, 116, −77,
86, 102, −122, 34, −72, −3, −76, 125, 98, 25, 78,
125, −63, −23, 4, −20, 117, −31, −23, 104, 102, 16,
56, 32, 87, 63, 128, −107, 3, −69, 25, 50, −80,
−84, 127, −92, 28, 63, 53, −111, −95, −107, 113,
106, −100, −5, −16, 109, 125, 28, −107, 111, −88,
−67, −99, −116, −13, 122, −21, −54, −103, −16,
−30, −49, −89, −12, −21, 20, −22, 108, 116, −111,
−84, 17, −34, −79, 123, −42, 11, 79, 120, 83, −31,
124, −19, 28, 12, 87, 76, 94, 112, −104, 76, 22,
85, −127, −115, 24, 5, 18, −6, 31, 94, 116, 109,
−90, 61, 31, −107, 19, 62, −59, −96, 109, −74, 111,
23, 69, 14, 5, 21, −61, 103, −120, −76, 34, −56,
−11, −82, 98, −24, 111, −25, 106, −73, 35, −70,
62, 18, 40, −33, 119, −44, 96, −31, −62, −125,
−127, 75, −14, 62, 11, −72, 65, 17, 91, −117, 63,
−37, 125, 67, 65, −14, −46, −21, 13, 42, 55, 73,
7, 87, 32, −47, 75, 95, −70, 70, −1, 72, −106, 34,
127, −92, 61, −99, −114, −45, −35, −111, −41, −117,
55, −126, 49]

Fig. 2: Decryption process trial

2. Decryption Process Trial

This test is to ensure that the NTRU cryptographic system can restore previously encrypted files to their original form (text files), using the appropriate private key. Users can select the file encryption option then users can select the encrypted file and the key file that has been downloaded during file encryption, users can directly click the decryption and download button then the system will process the file and the decrypted file can be downloaded by the user, and the decryption calculation process will be displayed



125*x^47 − 127*x^46 + 75*x^45 − 14*x^44 + 62*x^43 + 11*x^42 − 72*x^41 + 65*x^40 + 17*x^39 + 91*x^38 − 117*x^37 + 63*x^36 − 37*x^35 + 125*x^34 + 67*x^33 + 65*x^32 − 14*x^31 − 46*x^30 − 21*x^29 + 13*x^28 + 42*x^27 + 55*x^26 + 73*x^25 + 7*x^24 + 87*x^23 + 32*x^22 − 47*x^21 + 75*x^20 + 95*x^19 − 70*x^18 + 70*x^17 − x^16 + 72*x^15 − 106*x^14 + 34*x^13 + 127*x^12 − 92*x^11 + 61*x^10 − 99*x^9 − 114*x^8 − 45*x^7 − 35*x^6 − 111*x^5 − 41*x^4 − 117*x^3 + 55*x^2 − 126*x + 49

Kunci Privat f(x) = −x^502 + x^501 + x^500 + x^498 + x^497 + x^496 + x^495 + x^494 − x^493 + x^492 − x^491 − x^490 + x^489 + x^487 − x^486 − x^485 + x^484 + x^483 + x^482 + x^480 − x^479 + x^478 + x^477 − x^476 − x^475 − x^473 − x^472 + x^471 − x^470 − x^469 − x^468 − x^467 + x^466 − x^465 − x^464 + x^463 + x^461 + x^460 + x^459 + x^458 + x^457 + x^455 − x^454 − x^453 + x^452 − x^451 − x^449 − x^448 + x^447 + x^446 + x^445 + x^444 + x^443 − x^442 + x^441 − x^440 + x^439 − x^438 +

x^70 − x^69 + x^68 − x^67 + x^66 − x^65 + x^63 + x^62 + x^61 − x^59 − x^58 + x^57 − x^56 − x^55 + x^54 + x^53 + x^52 − x^51 + x^50 + x^48 − x^47 − x^46 − x^45 − x^44 − x^43 − x^42 + x^40 + x^38 + x^37 + x^35 − x^34 + x^33 − x^32 − x^30 − x^29 − x^28 + x^27 − x^26 − x^24 − x^22 − x^20 + x^19 − x^18 + x^16 + x^14 − x^13 − x^10 + x^9 − x^8 + x^7 − x^6 + x^5 + x^4 + x^3 + x^2 + x + 1

Invers f_p(x) = 2*x^502 + 2*x^501 + x^500 + 2*x^497 + x^496 + 2*x^495 + 2*x^488 + x^487 + 2*x^484 + 2*x^483 + x^482 + 2*x^480 + 2*x^479 + x^478 + x^477 + x^474 + x^473 + x^471 + 2*x^469 + x^467 + 2*x^466 + 2*x^465 + 2*x^464 + x^463 + 2*x^461 + 2*x^460 + 2*x^459 + 2*x^458 + x^457 + 2*x^456 + x^455 + x^454 + 2*x^453 + 2*x^452 + 2*x^451 + x^450 + x^448 + 2*x^445 + 2*x^444 + 2*x^443 + x^442 + x^441 + x^438 + x^437 + x^436 + x^435 + 2*x^432 + 2*x^431 + x^430 + x^429 + 2*x^428 + x^427 + 2*x^424 + 2*x^422 + 2*x^421 + x^420 + x^417 + x^416 + x^415 + 2*x^414 + x^412 +

x^45 + x^44 + x^43 + 2*x^41 + 2*x^40 + 2*x^39 + 2*x^38 + 2*x^37 + x^36 + 2*x^33 + x^32 + x^30 + 2*x^29 + 2*x^28 + x^27 + x^26 + x^25 + x^23 + 2*x^22 + 2*x^18 + x^15 + 2*x^14 + 2*x^13 + 2*x^12 + x^11 + 2*x^10 + 2*x^9 + 2*x^8 + 2*x^6 + x^5 + x^4 + x + 2

Menghitung a(x) = (f(x) * e(x)) mod (x^503−1) mod 256

Hasil a(x) = −19*x^502 + 19*x^501 + 4*x^500 − 48*x^499 + 18*x^498 + 27*x^497 − 4*x^496 − 26*x^495 − 20*x^493 + 11*x^492 + 37*x^491 + 7*x^490 + 19*x^489 − 12*x^488 − 25*x^487 + 20*x^486 + 38*x^485 − 14*x^484 − 31*x^483 + 32*x^482 − 25*x^481 − 10*x^480 − 24*x^479 − 25*x^478 + 6*x^477 − 18*x^476 + 3*x^475 + 35*x^474 + 29*x^473 − 5*x^472 + 25*x^471 + 20*x^470 + 24*x^469 − 8*x^468 − 26*x^467 − 11*x^466 − 5*x^465 − 15*x^464 − 13*x^463 + 13*x^462 + 22*x^461 + 12*x^460 + 9*x^459 − 23*x^458 +

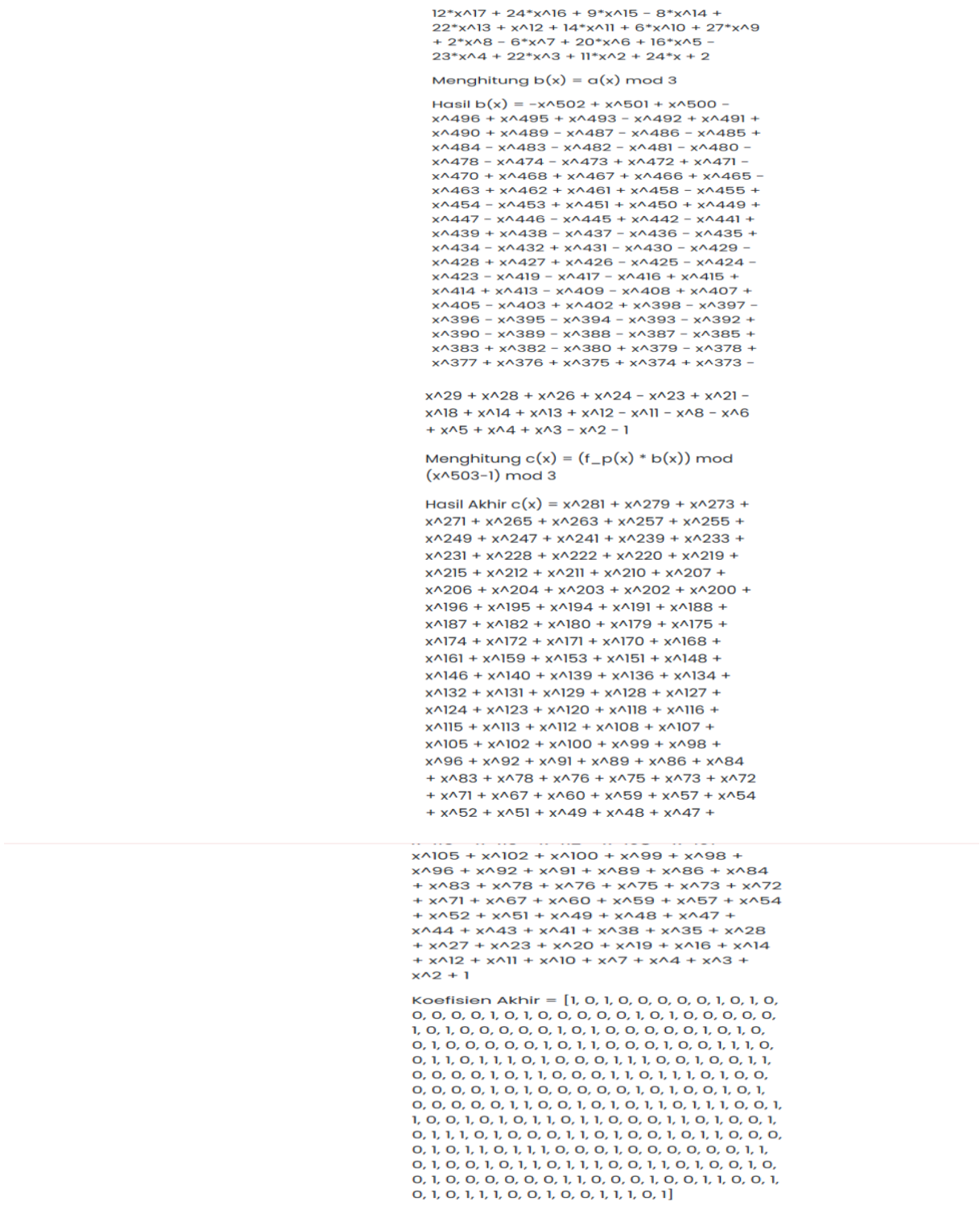


Fig 3: Decryption process trial

3. About Page Test

In this section, users can select the about menu in the system navigation, then the page will display information from the creator of this system.

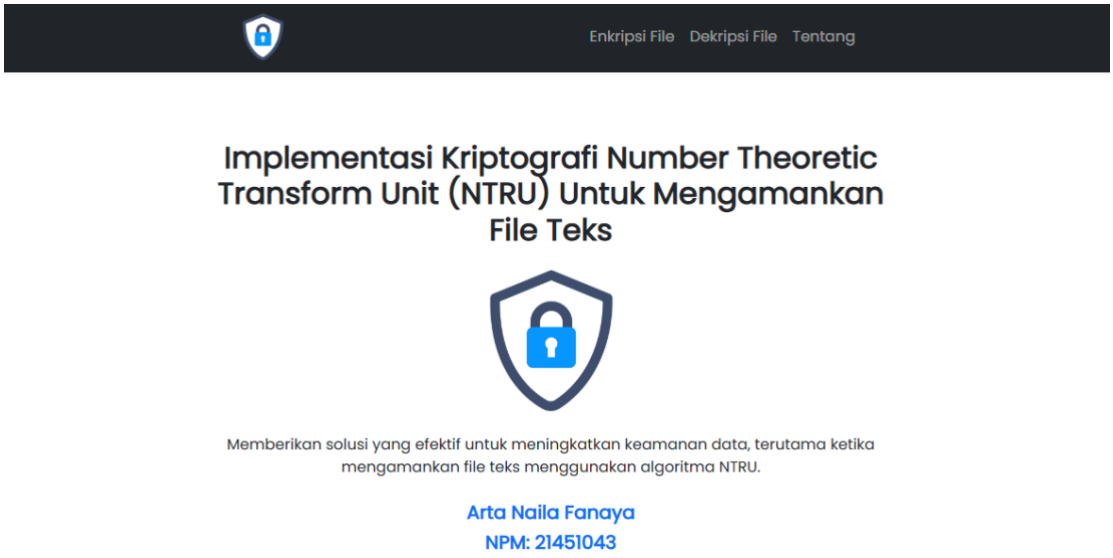


Fig. 4: Encrypted message test

4. Conclusion

Based on all stages of system design, implementation, and testing that have been carried out, it can be concluded that the implementation of NTRU cryptography has been successfully applied to secure a text file. This success is proven by system testing that shows the results of the encryption and decryption process on the text file, so that the security of the text file can be maintained from unauthorized access. System testing shows that all features such as about, encryption and decryption of files function properly .

Confession

To all those who have provided support, guidance, and prayers, I express my deepest gratitude. May all your kindness be repaid with even better.

Reference

[1.] B. Harjito, T. Setyawati, and A. Wijayanto, "Comparative Analysis between Elgamal and NTRU Algorithms and their implementation of Digital Signature for Electronic Certificate," *Int. J. Electr. Comput. Eng. Syst.*, vol. 13, no. 9, pp. 729–739, 2022, doi: 10.32985/ijeces.13.9.1.

[2.] A. Bhowmik and U. Menon, "Enhancing the NTRU Cryptosystem," *Int. J. Comput. Appl.*, vol. 176, no. 29, pp. 46–53, 2020, doi: 10.5120/ijca2020920320.

[3.] L. C. Hardiaman Hulu, William, D. Sipayung, M. Hafis, and Christnatalis, "Analysis A parallel combination between the Number Theorists aRe Us, Riverst Shamir Adleman and Triple Data Encryption Standard methods for measuring the speed of document security," *J. Phys. Conf. Ser.*, vol. 1230, no. 1, 2019, doi: 10.1088/1742-6596/1230/1/012093.

[4.] A. Karmakar, S. S. Roy, F. Vercauteren, and I. Verbauwhede, "Efficient finite field multiplication for isogeny based post quantum cryptography," 2017, doi: 10.1007/978-3-319-55227-9_14.

[5.] A. M. H. Pardede, M. Zarlis, and H. Mawengkang, "Optimization of Health Care Services with Limited Resources," *Int. J. Adv. Sci. Eng. Inf. Technol.*, vol. 9, no. 4, pp. 1444–1449, 2019, doi: 10.18517/ijaseit.9.4.8348.

[6.] A. M. H. Pardede, Y. Maulita, and R. Buaton, "Application modeling ipv6 (internet protocol version 6) on e-id card for identification number for effectiveness and efficiency of registration process identification of population," in *Journal of Physics: Conference Series*, 2018, vol. 978, . 1, doi: 10.1088/1742-6596/978/1/012017.

[7.] P. Mohanty, U. Choppali, and E. Kougianos, "Everything you wanted to know about smart cities," *IEEE Consum. Electron. Mag.*, vol. 5, no. 3, pp. 60–70, 2016, doi: 10.1109/MCE.2016.2556879.

[8.] W. A. Jabbar, W. K. Saad, and M. Ismail, "MEQSA-OLSRv2: A multicriteria-based hybrid multipath protocol for energy-efficient and QoS-aware data routing in MANET-WSN convergence scenarios of IoT," *IEEE Access*, 2018, doi: 10.1109/ACCESS.2018.2882853.

[9.] D. Niyigena, C. Habineza, and T. S. Ustun, "Computer-based smart energy management system for rural health centers," 2016, doi: 10.1109/IRSEC.2015.7455005.

[10.] F.-Z. Younsi, A. Bounnekar, D. Hamdadou, and O. Boussaid, "SEIR-SW, Simulation Model of Influenza Spread Based on the Small World Network," *Tsinghua Sci. Technol.*, vol. 20, no. 5, pp. 460–473, 2015.