ASEAN Engineering Journal

Full Paper

COMPARATIVE STUDY OF ELGAMAL AND LUC ALGORITHM IN CRYPTOGRAPHIC KEY GENERATION

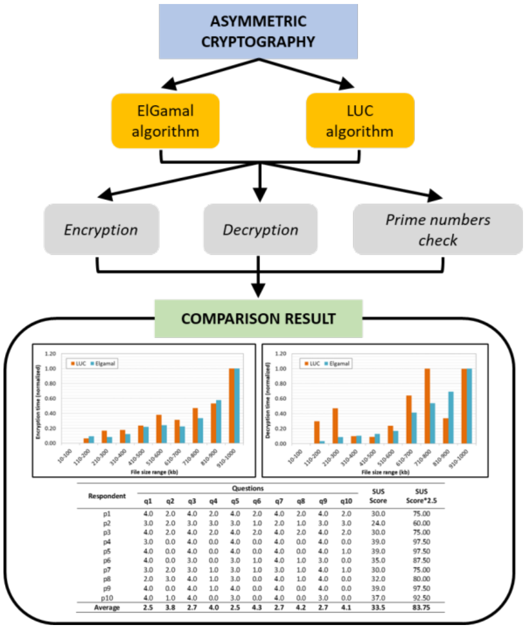
Nur Rochmah Dyah Puji Astuti[a]*, Dimas Panji Setiawan[a], Dhias Cahya Hakika[b]

[a]Informatics Department, Faculty of Industrial Technology, Universitas Ahmad Dahlan, Yogyakarta, Indonesia
[b]Chemical Engineering Department, Faculty of Industrial Technology, Universitas Ahmad Dahlan, Yogyakarta, Indonesia

Article history
Received
12 September 2022
Received in revised form
15 April 2023
Accepted
02 May 2023
Published online
30 November 2023

*Corresponding author
rochmahdyah@tif.uad.ac.id

Graphical abstract



Respondent	Questions										SUS Score	SUS Score*2.5
	q1	q2	q3	q4	q5	q6	q7	q8	q9	q10		
p1	4.0	2.0	4.0	2.0	4.0	2.0	4.0	2.0	4.0	2.0	30.0	75.00
p2	3.0	2.0	3.0	3.0	3.0	1.0	2.0	1.0	3.0	3.0	24.0	60.00
p3	4.0	2.0	4.0	2.0	4.0	2.0	4.0	2.0	4.0	2.0	30.0	75.00
p4	3.0	0.0	4.0	0.0	4.0	0.0	4.0	0.0	4.0	0.0	39.0	97.50
p5	4.0	0.0	4.0	0.0	4.0	0.0	4.0	0.0	4.0	1.0	39.0	97.50
p6	4.0	0.0	3.0	0.0	3.0	1.0	4.0	1.0	3.0	0.0	35.0	87.50
p7	3.0	2.0	3.0	1.0	3.0	1.0	3.0	1.0	4.0	1.0	30.0	75.00
p8	2.0	3.0	4.0	1.0	3.0	0.0	4.0	1.0	4.0	0.0	32.0	80.00
p9	4.0	0.0	4.0	1.0	4.0	0.0	4.0	0.0	4.0	0.0	39.0	97.50
p10	4.0	1.0	4.0	0.0	3.0	0.0	4.0	0.0	3.0	0.0	37.0	92.50
Average	**2.5**	**3.8**	**2.7**	**4.0**	**2.5**	**4.3**	**2.7**	**4.2**	**2.7**	**4.1**	**33.5**	**83.75**

Abstract

In today's era of digital, data security in communication channel becomes important factor to be considered during exchange of information. Cryptography is one of techniques to send and receive information securely through an insecure channel. Based on the number of keys used, encryption methods are categorized as symmetric and asymmetric cryptography. Compared to symmetric cryptography that often suffers from key management issues, asymmetric cryptography delivers higher level of data security. Thus, asymmetric cryptography is more preferred when security if the priority. To determine suitable algorithm, three essential aspects should be considered: security, speed, and prime numbers. This study aims to compare the application of asymmetric cryptographic algorithms between ElGamal and LUC algorithms in the key generation process. A comparative analysis of these two algorithms was conducted by evaluating the processing speed and prime numbers during key generation process to determine the advantages and drawbacks from ElGamal and LUC algorithms. The application in this study was developed using PHP programming language by following the Waterfall Model. Application testing involved two kinds of tests: (i) Black Box test and (ii) System Usability Scale (SUS) test. Results show the application developed from this study successfully performed the encryption, decryption, and checking of prime numbers from ElGamal and LUC algorithms. It displayed ciphertext, plaintext, and the speed of the encryption and decryption process from both methods. The black box test showed that all application functions follow the user's needs, while System Usability Scale (SUS) test obtained an average score of SUS interpretation of 83.75. This value means the adjective ratings was "excellent", the grade scale was "B", and the acceptability range was "acceptable". It is concluded that the LUC algorithm superior to ElGamal in terms of the speed of encryption process. However, during the decryption process the LUC algorithm responded slower than ElGamal.

Keywords: algorithm, asymmetric, cryptography, ElGamal, key generator, LUC.

1.0 INTRODUCTION

Security is an essential aspect of securing important data when information is shared during this fast-paced communication and digital era. Cryptography is one of the techniques to maintain the confidentiality of messages. Cryptography can be interpreted as the art of hiding specific message from information so that the arrangement of the message becomes random and could not be understood by unauthorized people. This tool is used to send and receive information and data securely through an insecure channel. Based on the number of keys that are used, encryption methods are categorized as symmetric key cryptography and asymmetric key cryptography [1]. Symmetric key cryptography is often called conventional cryptographic algorithms as it uses the same secret keys for encryption and decryption processes. This approach is the inverse of asymmetric cryptography, which uses different secret keys for encryption and decryption processes [2]. The benefits of symmetric cryptography are its simplicity and flexibility to be implemented. However, it has concern with "key distribution" issues, because if a third party intercepts during the exchange of the keys, the message can be decrypted as the same key is used to encrypt and decrypt the message [3]. This problem can present trust problem related to the integrity of secure communication as the secret key have to remain secure. Thus, symmetric cryptography may not feasible to be implemented due to risk and inconvenience [4], [5]. To overcome this problem, asymmetric key cryptography is developed as an alternative mode. Asymmetric cryptography, often called public key cryptography, uses one called the 'public key' and another called the 'private key'. The strength of asymmetric cryptography is increased data security, because there is no need to securely transmit a secret key [6], [7].

There are various types of asymmetric cryptography like RSA, Diffie-Helman, ECC, DSA, ElGamal, and LUC. In this study, two modern asymmetric algorithms, El Gamal and LUC, is chosen to be studied, because they are more efficient and used smaller amount of memory than other old algorithms. Furthermore, the development of various new programs with new technologies that need more security issues is increasing [8], [9]. Thus, these algorithms have potential to be further elaborated. ElGamal is a block cipher algorithm that performs the encryption process on ciphertext blocks, which are then decrypted, and the results are combined into a complete and understandable message [10], [11]. The strength of the ElGamal algorithm lies in the difficulty of calculating discrete logarithms on large modulo prime numbers, making it difficult to be solved. The ElGamal algorithm consists of three processes: (i) key generation, (ii) encryption, and (iii) decryption. Meanwhile, LUC algorithm is a cryptographic method using two different keys in the cryptosystem. This algorithm is mostly used in cryptography for data encoding, signatures, and key generation. The stages of LUC algorithm include three parts similar to ElGamal: (i) key generation, (ii) encryption process, and (iii) decryption process. The mathematical operation of LUC's cryptography is the Lucas series. From the large numbers in the Lucas series, explicitly testing the primacy and the results are used as an efficient algorithm for the implementation of LUC. In the LUC algorithm, the initial process is to determine two prime numbers and then calculate N=pq. The next step is to determine the public key e by selecting one of the numbers that prime relative to (p-1), (p+1), (q-1), (q+1) [12], [13].

ElGamal and LUC algorithms are cryptographic algorithms with a relatively high level of security. ElGamal algorithm provided bidirectional identity authentication between both communications sides which prevents the attacker sending a forged message. This algorithm also includes information to track the origins of messages during communication, allowing message recipients to properly verify the message's validity [14]. Meanwhile, LUC algorithm are not formulated in terms of exponentiation. This would make LUC unsusceptible to some attacks that jeopardize the security of other algorithms such as RSA or Diffie-Helman [15]. Thus, these two algorithms would provide security far superior than other traditional asymmetric algorithms.

This level of security can be used to secure the contents of information that would be shared. However, the algorithm speed could be a disadvantage of public-key cryptography as there is a trade-off between efficiency and security. Thus, in order to consider the strength against cryptoanalysis and Brutforse attacks, a suitable algorithm needs to be carefully chosen. Algorithms with a high level of security but a slow provisioning process would not be the primary choice for users [16]–[18]. Apart from the speed of prime numbers as a key generator, conducting secure encryption is also a concern for users. It is necessary to use optimal prime numbers for an algorithm.

Some researchers have conducted study to compare various cryptographic methods to determine its effectiveness during certain tasks. For example, Iswari [11] compared RSA and ElGamal in terms of key generation, while Mawengkang [19] compared ElGamal and LUC algorithm in terms of file security. Sann [20] also performed comparison of ElGamal and RSA algorithm for the encryption and decryption process for mail services. This study aims to compare the application of asymmetric cryptographic algorithms between ElGamal and LUC algorithms in terms of key generation process. A comparative study of these two algorithms will be conducted by analyzing the processing speed and prime numbers during key generation process to determine the advantages and the drawbacks from each algorithms.

2.0 METHODOLOGY

2.1 Research Object

An experimental research strategy and quantitative approach were used in this study. Three properties: (i) speed of the encryption, (ii) speed of decryption process, and (ii) prime numbers of key generators were evaluated using an application which processing text type file. The key generation, encryption, and decryption process would then produce a private key, public key, ciphertext, plaintext, and the speed of the encryption and decryption process from the input of ElGamal and LUC algorithms by the user. The data attributes used are prime numbers and the message that are to be hidden or displayed.

2.2 Stages of System Development

To develop the application for comparing asymmetric cryptography algorithms between the ElGamal and LUC algorithms, System Development Life Cycle (SDLC) method with the Waterfall model was used in this study. Figure 1 depicts the methodology of SDLC Waterfall model. This model comprises five phases: (i) requirements analysis and definition, (ii) system and software design, (iii), implementation and unit testing, (iv) integration and system testing, and (v) operation and maintenance.

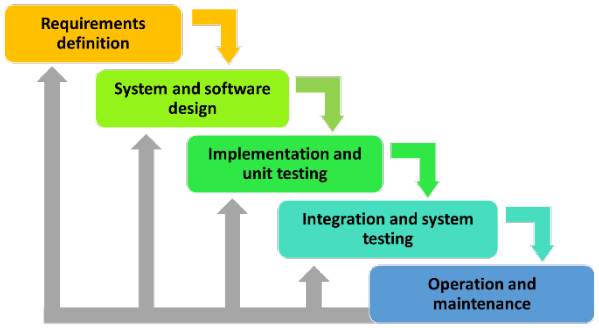

Requirements definition
System and software design
Implementation and unit testing
Integration and system testing
Operation and maintenance

Figure 1 The SDLC Waterfall model methodology [21]

2.3 Experimental Procedure

Research procedure in this study consists of three steps:

1. System design

At this stage, user determined the system services, constraints, and objectives. These components were defined in detail and served as system specifications. After obtaining the system requirements specifications, the system design stage was carried out, which allocates hardware and software requirements by forming the overall system architecture. Software design involved identifying and delineating the basic system abstractions of software and their relationships. The flowcharts of encryption and decryption process for ElGamal and LUC algorithm were shown by Figure 2 and Figure 3, respectively.

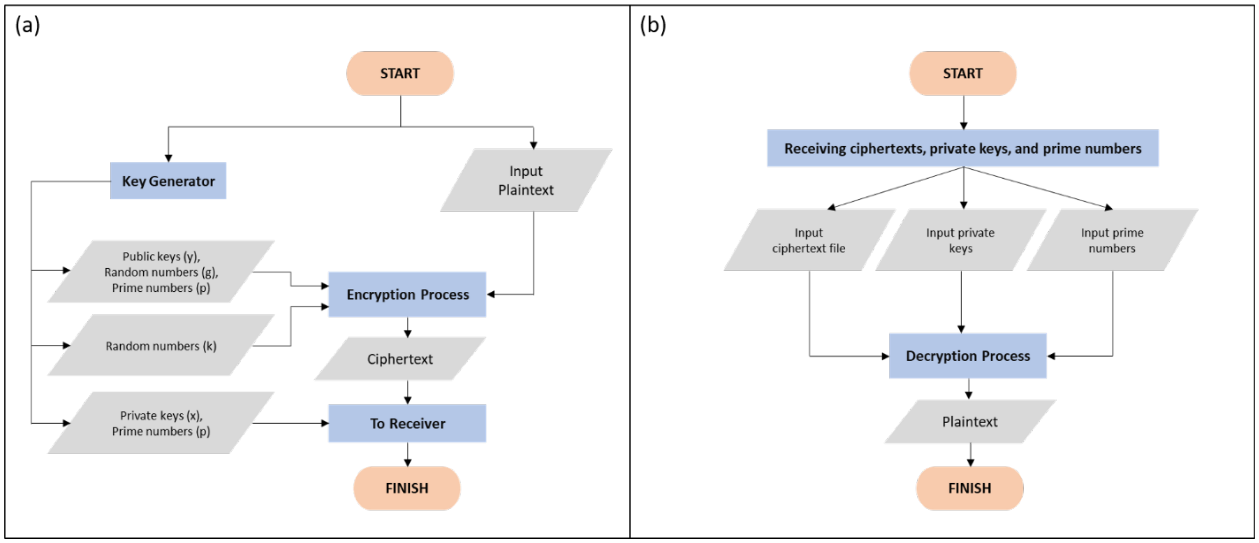


Figure 2 Flowchart for the: (a) encryption and (b) decryption procedure of ElGamal algorithm

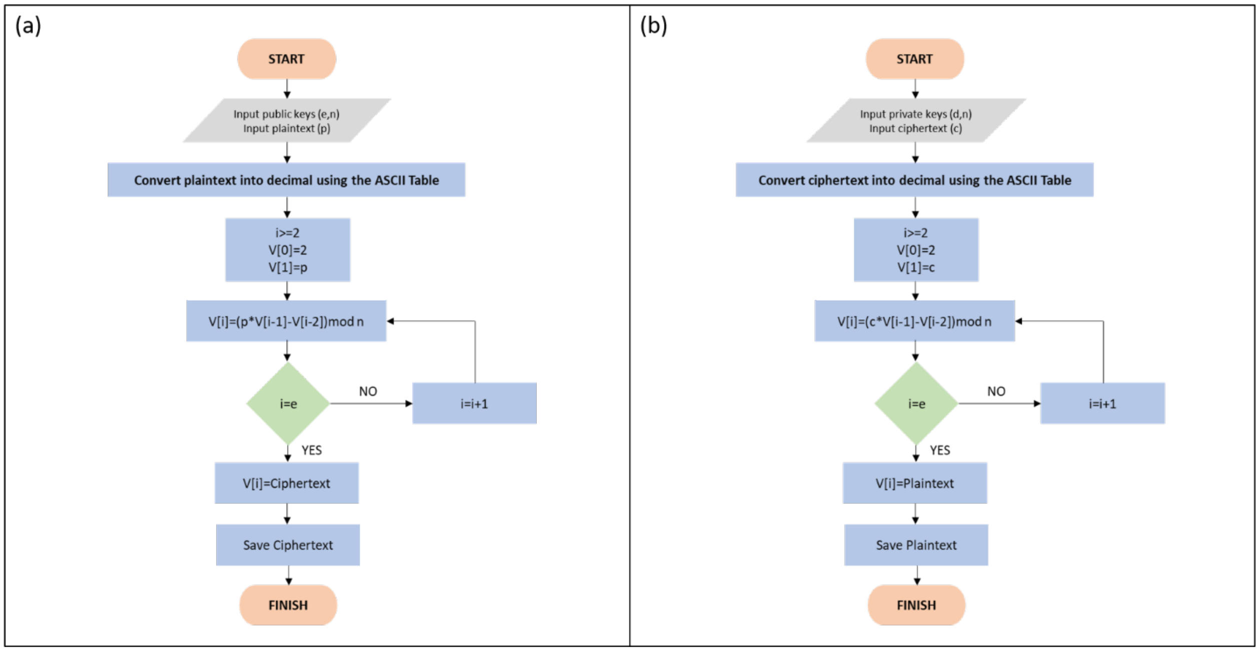


Figure 3 Flowchart for the: (a) encryption and (b) decryption procedure of LUC algorithm

2. System development and implementation

The system was then developed based on the design that previously made by applying a programming language per existing rules using the PHP programming language.

3. System testing

After completing the application, the system was tested to evaluate the errors in the system and improve the results. Two methods, Black Box test and System Usability Scale (SUS), are applied for the system testing in this study.

(a) Black Box test

Black Box Testing is a software testing technique to discover the functional specifications of the application being developed. It works by matching the outputs value based on the software input value without knowing the program code is used [22]. In this study, this test was carried out on three stories (encryption, decryption, and prime numbers check).

(b) SUS test

The SUS is well known as a reliable tool for evaluating products and systems [23]. In this study, the usability testing is done by using a questionnaire as a test measuring tool. A questionnaire was given to 10 students of Informatics Department at Universitas Ahmad Dahlan. The value scale conversion (0-4) from each respondent's questions for the odd statement value will be reduced by one value scale given by the respondent. Furthermore, for an even statement, five is subtracted from the scale value given by the respondent. The total number of scale values was multiplied by 2.5 to get the System Usability Scale (SUS) average value. Figure 4 shows the guide for grade rankings of SUS score.

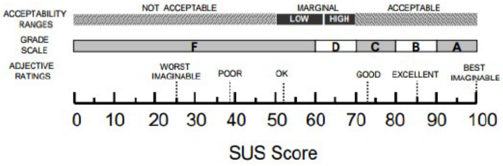


Figure 4 Grade rankings of SUS ccore [24]

3.0 RESULTS AND DISCUSSION

The analysis of ElGamal and LUC asymmetric cryptographic algorithms has been successfully developed and evaluated using an application. The application was created by PHP programming language.

3.1 Application Interface

The interface of this application, as shown in Figure 5, depicts three properties: (i) encryption, (ii) decryption, and (iii) prime numbers check. The encryption interface provides a form to input prime numbers and plaintext to perform the encryption process from ElGamal and LUC algorithms.

Prime numbers and plaintext input were executed by the encryption and decryption button to start the process. It would generate the private key, public key, ciphertext, and the speed of the encryption process. After being executed, the result generated from the application is shown in Figure 6 which includes public key, private key, ciphertext, and the speed of the encryption process.

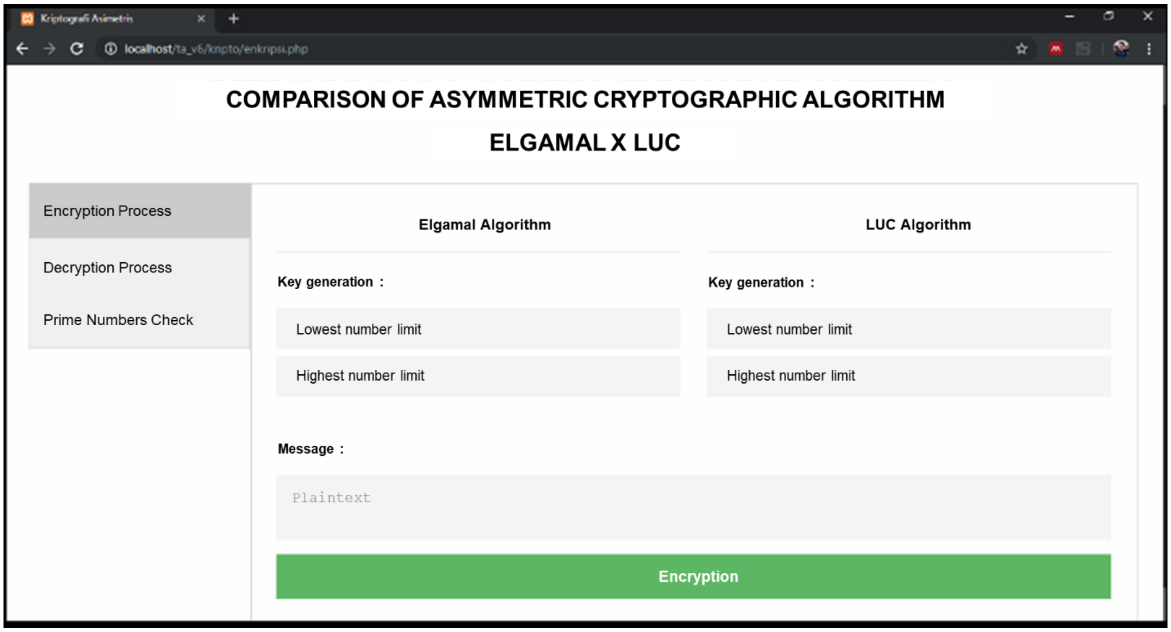
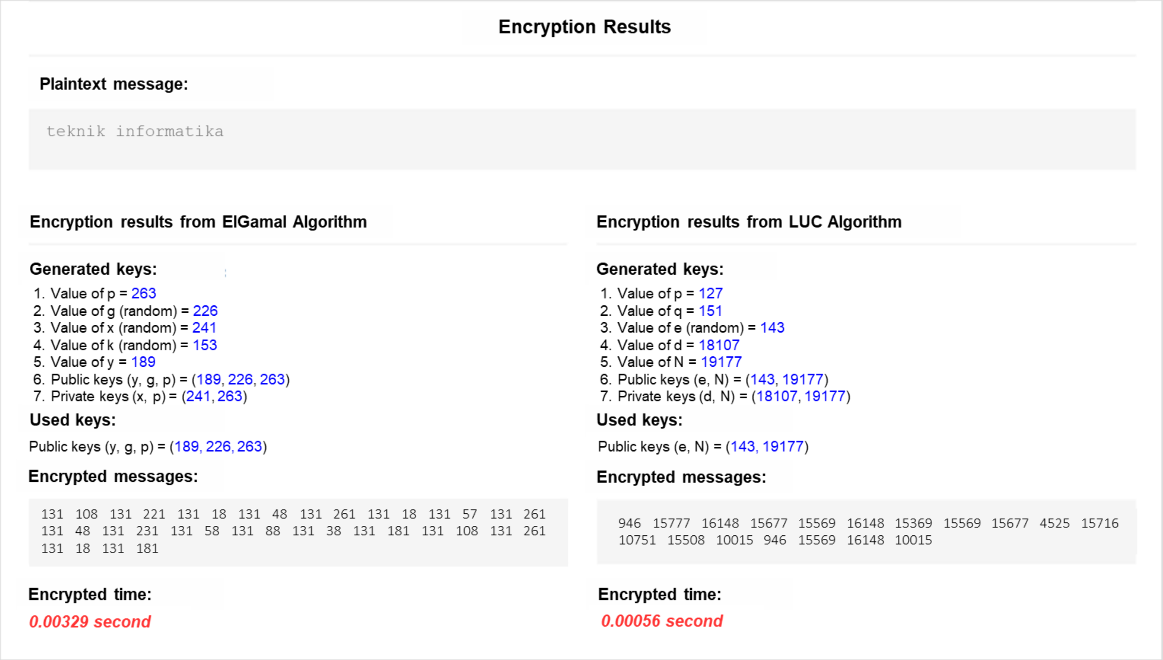


Figure 5 Application interface



Encryption Results

Plaintext message:

```
teknik informatika
```

Encryption results from ElGamal Algorithm

Generated keys:

1. Value of p = 263
2. Value of g (random) = 226
3. Value of x (random) = 241
4. Value of k (random) = 153
5. Value of y = 189
6. Public keys (y, g, p) = (189, 226, 263)
7. Private keys (x, p) = (241, 263)

Used keys:

Public keys (y, g, p) = (189, 226, 263)

Encrypted messages:

```
131  108  131  221  131  18  131  48  131  261  131  18  131  57  131  261
131  48  131  231  131  58  131  88  131  38  131  181  131  108  131  261
131  18  131  181
```

Encrypted time:

0.00329 second

Encryption results from LUC Algorithm

Generated keys:

1. Value of p = 127
2. Value of q = 151
3. Value of e (random) = 143
4. Value of d = 18107
5. Value of N = 19177
6. Public keys (e, N) = (143, 19177)
7. Private keys (d, N) = (18107, 19177)

Used keys:

Public keys (e, N) = (143, 19177)

Encrypted messages:

```
946  15777  16148  15677  15569  16148  15369  15569  15677  4525  15716
10751  15508  10015  946  15569  16148  10015
```

Encrypted time:

0.00056 second

Figure 6 Encryption results from: ElGamal (left) and LUC (right) algorithms

3.2 Performance of ElGamal and LUC Algorithms

The performance of both algorithms was evaluated from their execution time during the encryption and decryption process. The ElGamal and LUC algorithms used 100 files with size range from 10 kb to 1000 kb with the same key generator (three-digit prime numbers) and the same file in each process. Table 1 shows the average time of the encryption and decryption process by ElGamal and LUC algorithm. The profiles of this experimental output are shown in Figure 7 and Figure 8, while Figure 9 compares the average time values from both algorithms.

Table 1 Average time of ElGamal and LUC algorithm during encryption and decryption process

File Size	Encryption Time(s)		Decryption Time(s)	
Range (kb)	Elgamal	LUC	Elgamal	LUC
10-100	13.05	0.17	7.71	751.43
110-200	48.02	0.92	14.48	1076.55
210-300	45.17	2.11	25.32	1266.86
310-400	59.52	2.21	28.54	861.37
410-500	96.13	2.89	33.63	849.63
510-600	104.95	4.57	41.66	1012.88
610-700	98.18	3.79	90.41	1453.53
710-800	139.45	5.59	115.64	1844.29
810-900	231.44	6.33	146.30	1120.58
910-1000	391.08	11.71	207.54	1842.60
Average	122.70	4.03	71.12	1207.97

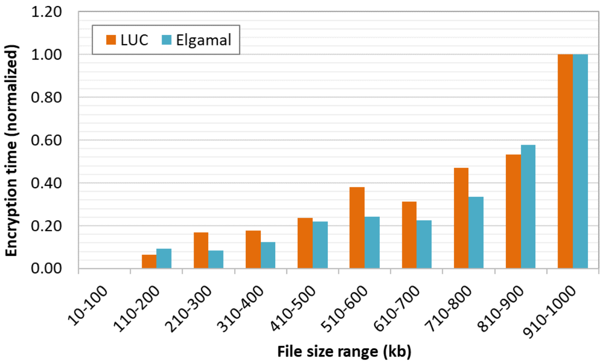


Figure 7 Average time profile of encryption process from ElGamal and LUC algorithm (normalized data)

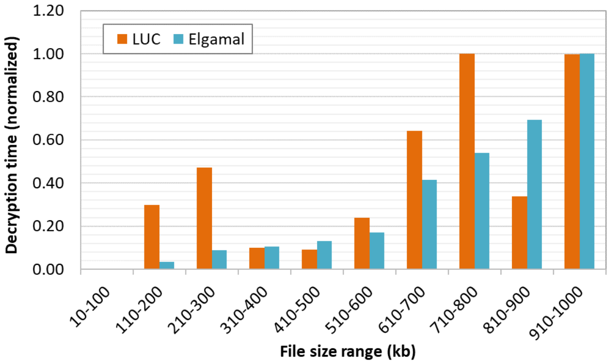


Figure 8 Average time profile of decryption process from ElGamal and LUC algorithm (normalized data)

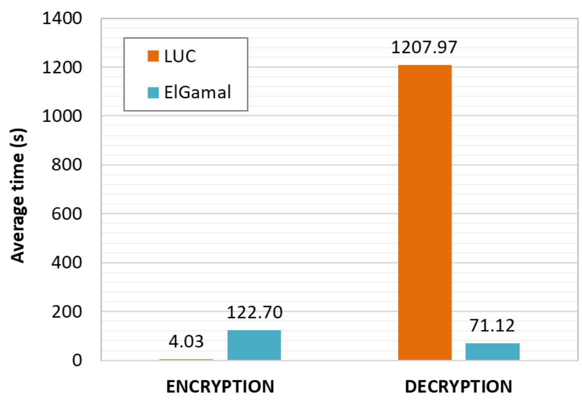


Figure 9 Comparison of the total average time of ElGamal and LUC algorithm

From Table 1, it can be seen that the larger file size range that will be used for the encryption and decryption process, the longer time needed for the algorithm to process it. As shown in Figure 9, during encryption process the average time of LUC algorithm is faster than ElGamal. Figure 7 shows that from the normalized data during encryption process, LUC indicated more consistent speed than ElGamal. In contrast with encryption process, from Figure 9 it can be seen that the average time of LUC algorithm during decryption is significantly slower than ElGamal. Based on its consistency during the decryption process, ElGamal provided more steady profile, while LUC exhibited fluctuation as represented in Figure 8.

The difference profile during both process (encryption and decryption) of these two algorithms is strongly influenced by the size of public and private keys, where the data length being processed is important here [25], [26]. It can be seen that during encryption, LUC executed data faster than ElGamal because smaller number of data was processed. While during decryption, LUC took longer time than ElGamal to execute the data. LUC has been acknowledged as strong algorithm, as it is not formulated in terms of exponentiation. However, this unavailability of sub-exponential logarithm leads LUC algorithm required longer time to break the ciphertext and execute the decryption process [27]. The slow speed during decryption process in LUC is caused by longer time is needed to return data from ciphertext to plaintext. This is why asymmetric algorithm is often used for privacy because encryption would be more important than decryption [28].

Meanwhile, ElGamal is computed based on the discrete logarithm, so this algorithm takes longer time during encryption process as large computing resources is needed and the process is more complicated [29]. However, ElGamal has some interesting properties such as homomorphic properties that make it useful in specific applications. It also leads to a fairly simple threshold cryptosystem [30]. Based on the performance comparison, it is considered to choose LUC algorithm if user's priority is to encrypt messages of any size faster. On the contrary, if user's priority is to decrypt messages faster, it is preferred to use ElGamal algorithm.

3.3 Application Testing with the Black Box and SUS Method

The performance of application was evaluated using Black Box Testing. This technique is working on the functional specifications of the application used. In addition, to measure its usability, the application was also tested using SUS for evaluating its functionality. Results of the Black Box and SUS test can be seen in Table 2 and Table 3, respectively.

Table 2 Acceptance test results from Black Box method

Stories	Action	Expected results	Output
Encryption	1. Input prime numbers as ElGamal key generator 2. Input prime numbers as LUC key generator 3. Input plaintext to be encrypted 4. Click the encryption button	1. The prime number has been successfully inputted and fulfills the elgamal encryption process equation 2. The prime number has been successfully inputted and satisfies the LUC encryption process equation 3. Plaintext has been inputted successfully and is ready to be encrypted 4. The encryption results appear in the ciphertext and the speed of the encryption process for each algorithm.	*IN ACCORDANCE*
Decryption	1. Enter ElGamal private key 2. Enter LUC private key 3. ElGamal ciphertext input 4. LUC ciphertext input 5. Click the decryption button	1. Elgamal private key has been successfully inputted and meets the elgamal decryption process equation 2. LUC private key was successfully inputted and satisfied LUC decryption process equation 3. Ciphertext has been inputted successfully and is ready to be decrypted 4. Ciphertext has been inputted successfully and is ready to be decrypted 5. The decryption results appear in the form of plaintext, and the speed of the decryption process for each algorithm	*IN ACCORDANCE*
Prime numbers check	1. Input the number to be checked 2. Click the check number button	1. Numbers have been successfully inputted and are ready to be checked whether they are prime numbers or not 2. Display information on whether the number entered is a prime number or not	*IN ACCORDANCE*

Table 3 SUS test results

Respondent	Questions										SUS Score	SUS Score*2.5
	q1	q2	q3	q4	q5	q6	q7	q8	q9	q10		
p1	4.0	2.0	4.0	2.0	4.0	2.0	4.0	2.0	4.0	2.0	30.0	75.00
p2	3.0	2.0	3.0	3.0	3.0	1.0	2.0	1.0	3.0	3.0	24.0	60.00
p3	4.0	2.0	4.0	2.0	4.0	2.0	4.0	2.0	4.0	2.0	30.0	75.00
p4	3.0	0.0	4.0	0.0	4.0	0.0	4.0	0.0	4.0	0.0	39.0	97.50
p5	4.0	0.0	4.0	0.0	4.0	0.0	4.0	0.0	4.0	1.0	39.0	97.50
p6	4.0	0.0	3.0	0.0	3.0	1.0	4.0	1.0	3.0	0.0	35.0	87.50
p7	3.0	2.0	3.0	1.0	3.0	1.0	3.0	1.0	4.0	1.0	30.0	75.00
p8	2.0	3.0	4.0	1.0	3.0	0.0	4.0	1.0	4.0	0.0	32.0	80.00
p9	4.0	0.0	4.0	1.0	4.0	0.0	4.0	0.0	4.0	0.0	39.0	97.50
p10	4.0	1.0	4.0	0.0	3.0	0.0	4.0	0.0	3.0	0.0	37.0	92.50
Average	**2.5**	**3.8**	**2.7**	**4.0**	**2.5**	**4.3**	**2.7**	**4.2**	**2.7**	**4.1**	**33.5**	**83.75**

Table 2 shows that based on the Black Box Testing, the outputs from three stories (encryption, decryption, and prime numbers check) were all in accordance with the action given by user. From the conversion of the SUS test data in Table 3, the average score of the SUS interpretation was 83.75. Based on the grade rankings of SUS score as shown in Figure 4, it means that the adjective rating of the application was "excellent"; the grade scale was "B"; and the acceptability range was "acceptable".

4.0 CONCLUSION

In this work, a comparative study of ElGamal and LUC algorithm in cryptographic key generation process has been conducted. Results show that ElGamal and LUC algorithm successfully performed encryption, decryption, and prime numbers check. The larger file size range that will be used for the encryption and decryption process, the longer time needed for the algorithm to process it. For the encryption process, LUC algorithm performed faster average time than ElGamal, while during decryption process the average time of LUC was notably slower than ElGamal. From the application evaluation, Black Box test show that the outputs were all in accordance with the action, while System Usability Scale (SUS) test resulted the average score of 83.75. This score represents that the adjective rating of the application was "excellent"; the grade scale was "B"; and the acceptability range was "acceptable".

Acknowledgement

Authors would like to thank Institute of Research and Community Service Universitas Ahmad Dahlan (LPPM UAD) for the financial support for this study through Internal Research Scheme.

References

[1] Sharma, D.K.; Singh, N.C.; Noola, D.A.; Doss, A.N.; Sivakumar, J. 2022. A review on various cryptographic techniques & algorithms. *Materials Today Proceedings* 51: 104–109

[2] J. Yashaswini, 2015 "Key Distribution for Symmetric Key Cryptography: A Review," *International Journal of Innovative Research in Computer and Communication Engineering*, 3(5): 4327–4331, Jun. DOI: https://doi.org/10.15680/ijircce.2015.0305047.

[3] H. I. Hussein, R. J. Mstafa, A. O. Mohammed, and Y. M. Younis, 2022, "An Enhanced ElGamal Cryptosystem for Image Encryption and Decryption," in 2022 *International Conference on Computer Science and Software Engineering (CSASE)*, Duhok, Iraq: IEEE, Mar. 337–342. DOI: https://doi.org/10.1109/CSASE51777.2022.9759643.

[4] M. A. Panhwar, S. A. Khuhro, G. Panhwar, and K. A. Memon, 2019, "SACA: A Study of Symmetric and Asymmetric Cryptographic Algorithms," presented at the IJCSNS *International Journal of Computer Science and Network Security*, 19: 48–55.

[5] L. Shinder and M. Cross, 2008, "Understanding Cybercrime Prevention," in Scene of the Cybercrime, *Elsevier,* 505-554. DOI: https://doi.org/10.1016/B978-1-59749-276-8.00012-1

[6] E. Conrad, S. Misenar, and J. Feldman, 2016, "Domain 3: Security Engineering (Engineering and Management of Security)," in CISSP Study Guide, *Elsevier,* 103-217. DOI: https://doi.org/10.1016/B978-0-12-802437-9.00004-7

[7] A. Dutta, 2022. "Comparison of Modern Cryptography Methods," Mathematics & Computer Science, preprint, DOI: https://doi.org/10.20944/preprints202207.0389.v1

[8] H. I. Hussein and W. M. Abduallah, 2021. "An efficient ElGamal cryptosystem scheme," *International Journal of Computers and Applications,* 43(10): 1088–1094, DOI: https://doi.org/10.1080/1206212X.2019.1678799

[9] T. Elgamal, 1985."A public key cryptosystem and a signature scheme based on discrete logarithms," *IEEE Transactions on Information Theory,* 31(4): 469-472, DOI: https://doi.org/10.1109/TIT.1985.1057074

[10] N. M. S. Iswari, 2016. "Key Generation Algorithm Design Combination of RSA and ElGamal Algorithm," *The 8th International Conference on Information Technology and Electrical Engineering,* 1-5, DOI: https://doi.org/10.1109/ICITEED.2016.7863255

[11] M. Othman, E. M. Abulhirat, Z. M. Ali, M. R. M. Said, and R. Johari, 2008. "A New Computation Algorithm for a Cryptosystem Based on Lucas Functions," *Journal of Computational Science.* 4(12): 1056-1060, DOI: https://doi.org/10.3844/jcssp.2008.1056.1060

[12] Z. M. Ali, M. Othman, M. R. Muhd, and M. N. Sulaiman, 2010,"Computation of Cryptosystem based on Lucas Functions using Addition Chain," in *2010 International Symposium on Information Technology, Kuala Lumpur,* Malaysia, 1082-1086. DOI: https://doi.org/10.1109/ITSIM.2010.5561514

[13] A. N. El-Kassar and R. Haraty, 2005. "ElGamal Public-Key cryptosystem in multiplicative groups of quotient rings of polynomials over finite fields," *Computer Science and Information Systems*, 2(1): 63–77, DOI: https://doi.org/10.2298/CSIS0501063E.

[14] D. Bleichenbacher, W. Bosma, and A. K. Lenstra, "Some Remarks on Lucas-Based Cryptosystems," 1995 in Advances in Cryptology — CRYPT0' 95, D. Coppersmith, Ed., in Lecture Notes in Computer Science. 963: 386–396. Springer Berlin Heidelberg Berlin, Heidelberg, DOI: https://doi.org/10.1007/3-540-44750-4_31.

[15] S. Mahajan and M. Singh, 2014. "Analysis of RSA Algorithm using GPU Programming," *International Journal of Network Security & Its Applications*. IJNSA, 6(4): 1-14 DOI: https://doi.org/10.5121/ijnsa.2014.6402

[16] S. Singh and R. Maini, 2011. "Comparison of data encryption algorithms," *International Journal of Computer Science & Communication*., 2(1): 125-127,

[17] Z. M. Ali, M. Othman, M. R. M. Said, and M. N. Sulaiman, 2008, "An efficient computation technique for Cryptosystems based on Lucas Functions," in 2008 *International Conference on Computer and Communication Engineering,* Kuala Lumpur, Malaysia, 187-190. DOI: https://doi.org/10.1109/ICCCE.2008.4580593

[18] H. Mawengkang, A. F. Siregar, and S. Efendi, 2018. "Combination analysis of ElGamal algorithm and LUC algorithm in file security," *IOP Conference Series: Materials Science and Engineering.*, 420(012130): 1-6, DOI: https://doi.org/10.1088/1757-899X/420/1/012130

[19] Z. Sann, T. thi Soe, K. W. M. Knin, and Z. M. Win, 2019. "Performance comparison of asymmetric cryptography (case study-mail message)," *Aptikom Journal on Computer Science and Information Technologies.* 4(3): 105-111, DOI: https://doi.org/10.11591/APTIKOM.J.CSIT.147

[20] B. Youssef, 2012. "A Simulation Model for the Waterfall Software Development Life Cycle," *International Journal of Engineering & Technology* 2(5): 1-7. DOI: https://doi.org/10.48550/arXiv.1205.6904

[21] Joosten, 2021. "The Black Box Testing and Loc Method Approach in Testing and Streamlining The Patient Registration Program,*" Jurnal Riset Informatika*, 3(2): 137-144. DOI: https://doi.org/10.34288/jri.v3i2.188

[22] B. Klug, 2017."An Overview of the System Usability Scale in Library Website and System Usability Testing," *Weave Journal of Library User Experience.* 1(6). DOI: https://doi.org/10.3998/weave.12535642.0001.602

[23] J. Brooke, 2013 "SUS: A Retrospective," *Journal of Usability Studies*, 8(2): 29-40.

[24] T. J. Wong, L. F. Koo, F. H. Naning, A. F. N. Rasedee, M. M. Magiman, and M. H. A. Sathar, 2021 “A Cubic El-Gamal Encryption Scheme Based On Lucas Sequence And Elliptic Curve,” *Advances in Mathematics: Scientific Journal,* 10(11): 3439–3447, doi: 10.37418/amsj.10.11.5.

[25] A. Mousa, 2005. "Security and performance of elgamal encryption parameters," *Journal of Applied Sciences.* 5(5): 883-886, DOI: https://doi.org/10.3923/jas.2005.883.886

[26] S. Okyere-Gamfi, J. B. H. Acquah, and V. Akoto-Adjepong, 2020. "An Enhanced Asymmetric Cryptosystem using Multiple Key System," *International Journal of Computers and Applications*, 176(15): 18-26. DOI: https://doi.org/10.5120/ijca2020920017

[27] D. Coppersmith, Ed., Advances in Cryptology --2003. CRYPT0' 95: *15th Annual International Cryptology Conference Santa Barbara, California,* USA, August 27-31, 1995 Proceedings. Berlin, Heidelberg: Springer Berlin Heidelberg, DOI: https://doi.org/10.1007/3-540-44750-4

[28] P. P. Sari, E. B. Nababan, and M. Zarlis, 2020 *"Comparative Study of LUC, ElGamal and RSA Algorithms in Encoding Texts*,", 148-151. DOI: https://doi.org/10.1109/MECnIT48290.2020.9166586

[29] M. A. Will and R. K. L. Ko, 2015, "A guide to homomorphic encryption," in The Cloud Security Ecosystem, *Elsevier.* 101-127. DOI: https://doi.org/10.1016/B978-0-12-801595-7.00005-7