Comparative Analysis of RSA and ElGamal Cryptographic Public-key Algorithms

Andysah Putera Utama Siahaan[1], Elviwani[2], Boni Oktaviana[3]
[1]Faculty of Science and Technology, Universitas Pembangunan Panca Budi, Medan, Indonesia
[2]Faculty of Computer Science and Information Technology, Universitas Sumatra Utara, Medan, Indonesia
[3]Facultyof Engineering and Computer Science, Universitas Harapan Medan, Medan, Indonesia

[1]andiesiahaan@gmail.com, [2]elvialischan@gmail.com, [3]bonioktaviana@yahoo.co.id

ABSTRACT

An asymmetric algorithm is an encryption technique that uses different keys on the process of encryption and decryption. This algorithm uses two keys, public key, and private key. The public key is publicly distributed while the private key is kept confidentially by the user and this key is required at the time of the decryption process. RSA and ElGamal are two algorithms that implement a public key cryptosystem. The strength of this algorithm lies in the bit length used. The degree of difficulty in RSA lies in the factorization of large primes while in ElGamal lies in the calculation of discrete logarithms. After testing, it is proven that RSA performs a faster encryption process than ElGamal. However, ElGamal decryption process is faster than RSA. Both of these algorithms are cryptographic public-key algorithms but have functions in different ways. RSA is a deterministic algorithm while ElGamal is a probabilistic algorithm.

Keywords: *Cryptography, RSA, ElGamal, Public Key, Asymmetric*

1. INTRODUCTION

Data security is very important to note especially if it is in the public network [1]–[5]. The data transmitted over the power grid [6] freely so that cryptography techniques must protect the data. An asymmetric algorithm is an algorithm where the encryption key used is different from the decryption key [7]–[10]. It is different from symmetric keys that use the same two keys during encryption and decryption process [11][12][13]. The asymmetric key uses two keys, public and private key. RSA and ElGamal both use asymmetric key techniques. The fundamental difference lies in the number of variables used. RSA uses two variables during encryption while ElGamal uses three variables. The RSA algorithm's strength is at the difficulty level in factoring the numbers into a prime factor. The public key "n" is the multiplication of two numbers stored in the variables "p" and "q." The factorization process for determining the value of "p" and "q" depends on the value of "n." If "n" is factored into, then determining the value "m" is easy. Although the value "e" is known, the key calculation "d" is not easy because the "m" value is unknown. The advantages of RSA algorithm is the defense system against various attacks, especially brute force attacks. It is because the decryption complexity can be determined by determining the large "p" and "q" values at the time of the key pair generator process [14]. The result "n" is a considerable number that creates a significant space and this makes RSA resistant to attack. However, the size of the private key that is too large will result in a reasonably slow decryption process, especially for large message sizes. Therefore, RSA is commonly used to encrypt small messages such as password encryption and pin number.

Unlike the ElGamal algorithm, this algorithm performs the encryption process on the plaintext blocks which then produces the ciphertext blocks. These hypertext blocks will be decrypted again, and the result is then merged into the original plaintext. The security of the ElGamal algorithm lies in the difficulty of calculating discrete logarithms on large prime modulo [15]. Solving this logarithm problem is a difficult thing to solve. The advantage of the ElGamal algorithm is the generation of keys using discrete logarithms. Encryption and decryption techniques use a large computing process so that the encryption results are twice the size of the original size. The disadvantage of this algorithm is that it requires a tremendous resource because the resulting ciphertext is twice the length of the plaintext and requires a processor capable of performing extensive computations for massive logarithmic calculations. This research tries to analyze which algorithm is better in doing the process of encryption and decryption.

2. LITERATURE REVIEW

2.1 Comparative Study of DES, 3DES, AES and RSA

The exchange of data through the internet and other types of media is beneficial for people in exchanging information [16][17]. Information delivery is speedy. It requires system protection against security attacks [18]. Many methods can be used to send data on time. The authors declare cryptography is a viable method to provide security mechanisms in real-time [19][20][21]. Cryptography is used to conceal information from wild parties. Their study analyze the DES, 3DES, AES and RSA algorithms regarding their ability to secure data protected from attacks. The speed and effectiveness of securing the data will also be tested [22][23].

This section analyzes the symmetric algorithms (DES, 3DES, AES), and RSA algorithms and their performance in encrypting input files of various content and sizes. Some of the factors that influence the results of the analysis are as follows.

- Size. Each algorithm requires different memory capacities to operate. This requirement is determined by the size of the plaintext, the number of rounds, etc. An algorithm is good if by using a small memory, the algorithm can process plaintext smoothly and quickly.

- Time. It is the amount of time required by the algorithm to complete the encryption and decryption process. The speed of the processor and the complexity of the algorithm will affect the performance of the algorithm.

- Throughput-Throughput algorithm on encryption and decryption is obtained by dividing the plaintext by the total time.

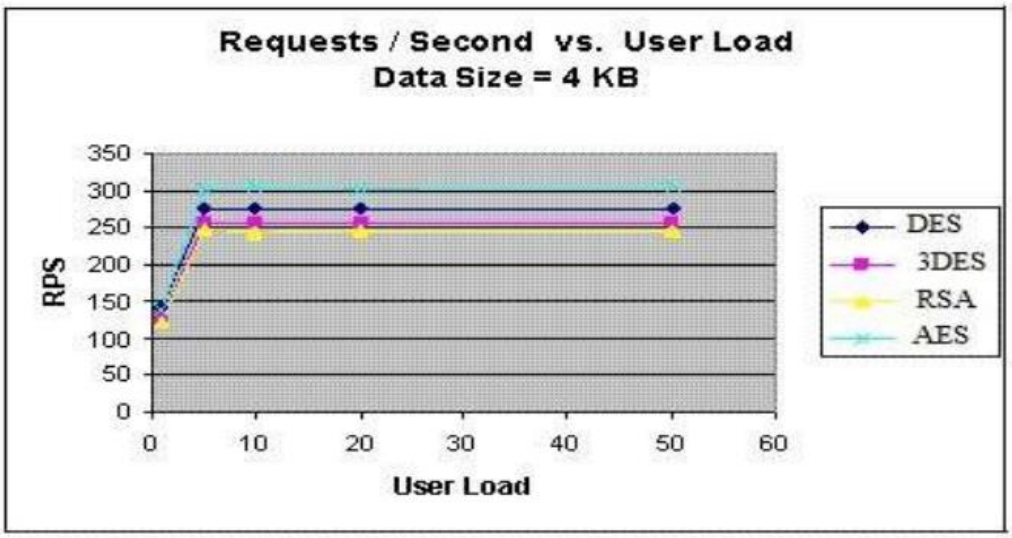

Figure 1. Comparison result Request vs User Load

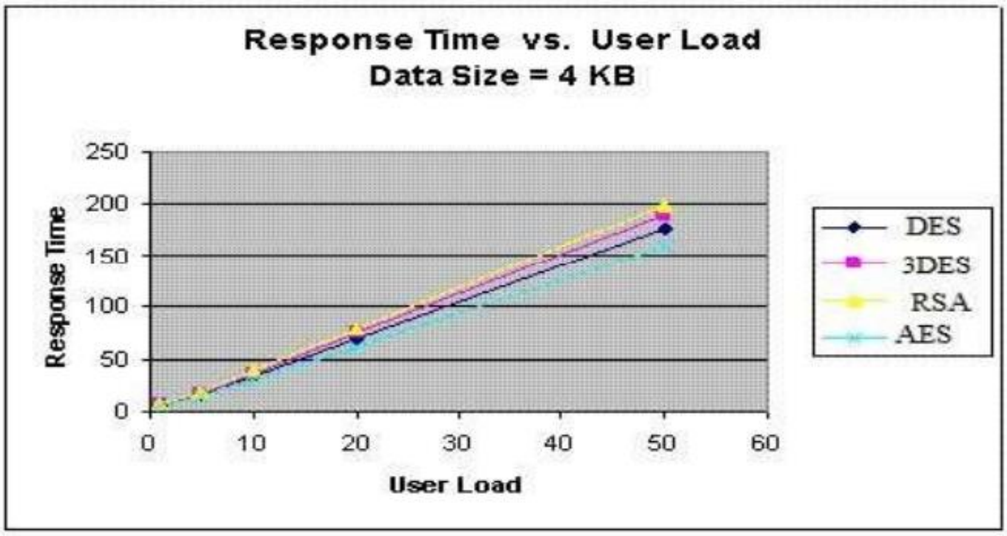

Figure 2. Comparison result Request vs User Load

The test result states that AES is better than other algorithms both in the number of request processes per second in different user loads as well as response times. AES has better performance and safety despite higher power consumption of RSA and Triple DES. DES has less power consumption than AES. The front DES has the most vulnerable security and can be easily solved by brute force attacks in just fifteen hours. A 128-bit AES key has comparable strength with RSA 2600-bit keys. It makes AES the best among the algorithms compared.

2.2 A Review on Public Key Encryption Algorithm

Computer security serves to maintain the integrity, availability, and confidentiality of information systems resources from wild parties [24][25]. The authenticity and correctness of the sent message must be completed so that the recipient receives the message as it is sent. What is worried is that during the sending of messages there is a modification of the message. Data privacy needs to be kept confidential especially in companies that have country data. RSA is an algorithm that can maintain data confidentiality at the time of authentication delivery. RSA has dynamic keys that can vary each time according to the generation of the key [14][26].

Hung-Min Sun's [27] research tries to modify RSA using a dual system. This system serves to reduce the need for key storage. The author says that the disadvantage of RSA dual-systems is the computational complexity of the key generation algorithms are also optimized.

Taher ElGamal proposed a signature scheme based on discrete logarithms. He has implemented Diffie-Hellman key distribution scheme to generate the public key for encryption and decryption processes. The strength depends on the difficulty of computing discrete logarithms over finite fields. The larger the number used, the harder the discrete logarithms is solved [15].

3. RESULT AND DISCUSSION

This section of the researcher tries to compare the two algorithms and find out which algorithm is faster and look for the advantages of each algorithm.

3.1 Key Generation

RSA produces six variables (P, Q, N, φ, E, D) at the time of key generation. Variables "N" and "E" are keys used for encryption and "N," and "D" are keys used for decryption. ElGamal produces four variables (P, G, X, Y) at the time of key generation. Variables "P," "G," and "Y" are used during the encryption process while variables "P" and "X" are used during the decryption process. The following example is RSA and ElGamal key generation.

RSA

P	=	5062283
Q	=	6515623
N	=	P.Q
	=	32983927547309
Φ	=	(P-1).(Q-1)
	=	32983915969404
E	=	287
D	=	11952359793791

ElGamal

P	=	6062429
G	=	1628134
X	=	660876
Y	=	G^X % P
		5809535

RSA and ElGamal have relatively the same time in the key generation. Generating a key does not take long for a number that is not so large. RSA and ElGamal take longer to generate 2048 bit keys because the calculation result must have modular expression.

3.2 Encryption

In the encryption section, the plaintext tested is "UNIVERSITY." This word will be encrypted according to the key to being raised. Several keys are made with different key lengths.

U	N	I	V	E	R	S	I	T	Y
85	78	73	86	69	82	83	73	84	89

```
RSA
P     =  6713911561289923
Q     =  8067467447266457
N     =  P.Q
      =  54164262964532367864523210012811
Φ     =  (P-1).(Q-1)
      =  54164262964532353083144201456432
E     =  733
D     =  47292125917190594779280066755957

C1    =  $85^{733}$ % 54164262964532367864523210012811
      =  20096929491328173590938043104042
C2    =  $78^{733}$ % 54164262964532367864523210012811
      =  48801761437637915480947952618010
C3    =  $73^{733}$ % 54164262964532367864523210012811
      =  48227725082732325579008683930221
C4    =  $86^{733}$ % 54164262964532367864523210012811
      =  11754012436905151520593852085384
C5    =  $69^{733}$ % 54164262964532367864523210012811
      =  51805072138259574569488165852517
C6    =  $82^{733}$ % 54164262964532367864523210012811
      =  8010444548914103342943171918866
C7    =  $83^{733}$ % 54164262964532367864523210012811
      =  29052294401379937407723425977319
C8    =  $73^{733}$ % 54164262964532367864523210012811
      =  48227725082732325579008683930221
C9    =  $84^{733}$ % 54164262964532367864523210012811
      =  39031922711229174544925519098765
C10   =  $89^{733}$ % 54164262964532367864523210012811
      =  1702407206289746953392490725740

Ciphertext:
20096929491328173590938043104042 48801761437637915480947952618010
48227725082732325579008683930221 11754012436905151520593852085384
51805072138259574569488165852517 8010444548914103342943171918866
29052294401379937407723425977319 48227725082732325579008683930221
39031922711229174544925519098765 1702407206289746953392490725740
```

```
Time: 0.0033971 second.

ElGamal
--------------------------------------------------------------------------------
P     =  76481
G     =  15442
X     =  30951
Y     =  $G^{X}$ % P
         $15442^{30951}$ % 76481
         26297

K[0]  =  68490
K[1]  =  42064
K[2]  =  70103
K[3]  =  25789
K[4]  =  39183
K[5]  =  54400
K[6]  =  61237
K[7]  =  73115
K[8]  =  48942
K[9]  =  44474

A[0]  =  ($15442^{68490}$) % 76481
      =  50157
B[0]  =  (($26297^{68490}$) * 85) % 76481
      =  49769
A[1]  =  ($15442^{42064}$) % 76481
      =  68957
B[1]  =  (($26297^{42064}$) * 78) % 76481
      =  24976
A[2]  =  ($15442^{70103}$) % 76481
      =  17835
B[2]  =  (($26297^{70103}$) * 73) % 76481
      =  26125
A[3]  =  ($15442^{25789}$) % 76481
      =  23423
B[3]  =  (($26297^{25789}$) * 86) % 76481
      =  50298
A[4]  =  ($15442^{39183}$) % 76481
      =  53509
B[4]  =  (($26297^{39183}$) * 69) % 76481
      =  335
A[5]  =  ($15442^{54400}$) % 76481
      =  56506
B[5]  =  (($26297^{54400}$) * 82) % 76481
      =  62508
A[6]  =  ($15442^{61237}$) % 76481
      =  43167
B[6]  =  (($26297^{61237}$) * 83) % 76481
      =  34850
A[7]  =  ($15442^{73115}$) % 76481
      =  56559
B[7]  =  (($26297^{73115}$) * 73) % 76481
```

= 71675

A[8] = (15442^{48942}) % 76481

= 32727

B[8] = $((26297^{48942}) * 84)$ % 76481

= 48351

A[9] = (15442^{44474}) % 76481

= 41457

B[9] = $((26297^{44474}) * 89)$ % 76481

= 65154

<u>Ciphertext:</u>
50157 49769 68957 24976 17835 26125 23423 50298 53509 335 56506 62508 43167 34850 56559 71675 32727 48351 41457 65154

Time: 1.2034075 second.

3.3 Decryption

The decryption process will return ciphertext to plaintext. The following is the decryption process of the RSA and ElGamal algorithms.

RSA

P = 6713911561289923

Q = 8067467447266457

N = P.Q

= 54164262964532367864523210012811

Φ = (P-1).(Q-1)

= 54164262964532353083144201456432

E = 733

D = 47292125917190594779280066755957

P1 = $20096929491328173590938043104042^{47292125917190594779280066755957}$ % 54164262964532367864523210012811

= 85

P2 = $48801761437637915480947952618010^{47292125917190594779280066755957}$ % 54164262964532367864523210012811

= 78

P3 = $48227725082732325579008683930221^{47292125917190594779280066755957}$ % 54164262964532367864523210012811

= 73

P4 = $11754012436905151520593852085384^{47292125917190594779280066755957}$ % 54164262964532367864523210012811

= 86

P5 = $51805072138259574569488165852517^{47292125917190594779280066755957}$ % 54164262964532367864523210012811

= 69

P6 = $8010444548914103342943171918866^{47292125917190594779280066755957}$ % 54164262964532367864523210012811

= 82

P7 = $29052294401379937407723425977319^{47292125917190594779280066755957}$ % 54164262964532367864523210012811

= 83

P8 = $48227725082732325579008683930221^{47292125917190594779280066755957}$ %
54164262964532367864523210012811
= 73
P9 = $39031922711229174544925519098765^{47292125917190594779280066755957}$ %
54164262964532367864523210012811
= 84
P10 = $17024072062897469533924907257740^{47292125917190594779280066755957}$ %
54164262964532367864523210012811
= 89

Plaintext:
85 78 73 86 69 82 83 73 84

Time: 0.0320240 second.

ElGamal

P = 76481
G = 15442
X = 30951
Y = G^{X} % P
15442^{30951} % 76481
26297

D[0] = (49769 * (50157^{45529})) % 76481
= 85
D[1] = (24976 * (68957^{45529})) % 76481
= 78
D[2] = (26125 * (17835^{45529})) % 76481
= 73
D[3] = (50298 * (23423^{45529})) % 76481
= 86
D[4] = (335 * (53509^{45529})) % 76481
= 69
D[5] = (62508 * (56506^{45529})) % 76481
= 82
D[6] = (34850 * (43167^{45529})) % 76481
= 83
D[7] = (71675 * (56559^{45529})) % 76481
= 73
D[8] = (48351 * (32727^{45529})) % 76481
= 84
D[9] = (65154 * (41457^{45529})) % 76481
= 89

Plaintext:
85 78 73 86 69 82 83 73 84

Time: 0.0278580 second.

4. CONCLUSION

The encryption and decryption time of the RSA algorithm is better than the ElGamal algorithm. Ciphertext RSA has fewer numbers than ElGamal algorithm. The ElGamal algorithm has a ciphertext pair. Each encrypted plaintext will generate two ciphertext values. RSA algorithm and ElGamal algorithm are asymmetric algorithms which have different formulas for encryption and decryption. RSA algorithm is faster than ElGamal algorithm. Regarding security, the ElGamal algorithm will be more challenging to solve than the RSA algorithm because ElGamal has a complicated calculation to solve discrete logarithms.

REFERENCES

[1] R. Rahim *et al.*, “Searching Process with Raita Algorithm and its Application,” *J. Phys. Conf. Ser.*, vol. 1007, no. 1, p. 012004, Apr. 2018.

[2] R. Meiyanti, A. Subandi, N. Fuqara, M. A. Budiman, and A. P. U. Siahaan, “The recognition of female voice based on voice registers in singing techniques in real-time using hankel transform method and macdonald function,” *J. Phys. Conf. Ser.*, vol. 978, no. 1, p. 012051, Mar. 2018.

[3] R. Rahim, M. Dahria, M. Syahril, and B. Anwar, “Combination of the Blowfish and Lempel-Ziv-Welch algorithms for text compression,” *World Trans. Eng. Technol. Educ.*, vol. 15, no. 3, pp. 292–297, 2017.

[4] R. Rahim, D. Hartama, H. Nurdiyanto, A. S. Ahmar, D. Abdullah, and D. Napitupulu, “Keylogger Application to Monitoring Users Activity with Exact String Matching Algorithm,” *J. Phys. Conf. Ser.*, vol. 954, no. 1, p. 012008, 2018.

[5] H. Nurdiyanto and R. Rahim, “Enhanced pixel value differencing steganography with government standard algorithm,” in *2017 3rd International Conference on Science in Information Technology (ICSITech)*, 2017, pp. 366–371.

[6] S. Aryza, M. Irwanto, Z. Lubis, A. P. U. Siahaan, R. Rahim, and M. Furqan, “A Novelty Design Of Minimization Of Electrical Losses In A Vector Controlled Induction Machine Drive,” *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 300, p. 012067, 2018.

[7] A. Putera, U. Siahaan, and R. Rahim, “Dynamic Key Matrix of Hill Cipher Using Genetic Algorithm,” *Int. J. Secur. Its Appl.*, vol. 10, no. 8, pp. 173–180, Aug. 2016.

[8] H. Nurdiyanto, R. Rahim, and N. Wulan, “Symmetric Stream Cipher using Triple Transposition Key Method and Base64 Algorithm for Security Improvement,” *J. Phys. Conf. Ser.*, vol. 930, no. 1, p. 012005, Dec. 2017.

[9] R. Rahim, “Man-in-the-middle-attack prevention using interlock protocol method,” *ARPN J. Eng. Appl. Sci.*, vol. 12, no. 22, pp. 6483–6487, 2017.

[10] E. Kartikadarma, T. Listyorini, and R. Rahim, “An Android mobile RC4 simulation for education,” *World Trans. Eng. Technol. Educ.*, vol. 16, no. 1, pp. 75–79, 2018.

[11] B. Oktaviana and A. P. U. Siahaan, “Three-Pass Protocol Implementation on Caesar Cipher in Classic Cryptography,” *IOSR J. Comput. Eng.*, vol. 18, no. 4, pp. 26–29, 2016.

[12] R. Rahim *et al.*, “Combination Base64 Algorithm and EOF Technique for Steganography,” *J. Phys. Conf. Ser.*, vol. 1007, no. 1, p. 012003, Apr. 2018.

[13] D. Abdullah, R. Rahim, D. Apdilah, S. Efendi, T. Tulus, and S. Suwilo, “Prime Numbers Comparison using Sieve of Eratosthenes and Sieve of Sundaram Algorithm,” in *Journal of Physics: Conference Series*, 2018, vol. 978, no. 1, p. 012123.

[14] D. Kurnia, H. Dafitri, and A. P. U. Siahaan, “RSA 32-bit Implementation Technique,” *Int. J. Recent Trends Eng. Res.*, vol. 3, no. 7, pp. 279–284, 2017.

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

[16] M. Iqbal, Y. Sahputra, and A. P. U. Siahaan, “The Understanding of GOST Crytography Technique,” *Inter Natl. J. Eng. Trends Technol.*, vol. 39, no. 3, pp. 168–172, 2016.

[17] A. P. U. Siahaan, “Blum Blum Shub in Generating Key in RC4,” *Int. J. Sci. Technoledge*, vol. 4, no. 10, pp. 1–5, 2016.

[18] I. Sumartono, A. P. U. Siahaan, and Arpan, “Base64 Character Encoding and Decoding Modeling,” *Int. J. Recent Trends Eng. Res.*, vol. 2, no. 12, pp. 63–68, 2016.

[19] W. Fitriani, R. Rahim, B. Oktaviana, and A. P. U. Siahaan, “Vernam Encypted Text in End of File Hiding Steganography Technique,” *Int. J. Recent Trends Eng. Res.*, vol. 3, no. 7, pp. 214–219, Jul. 2017.

[20] A. P. U. Siahaan, “Rail Fence Cryptography in Securing Information.”

[21] A. P. U. Siahaan, “Rabin-Karp Elaboration in Comparing Pattern Based on Hash Data,” *Int. J. Secur. Its Appl.*, vol. 12, no. 2, pp. 59–66, 2018.

[22] G. Singh and Supriya, “A Study of Encryption Algorithms (RSA, DES, 3DES and AES) for Information Security,” *Int. J. Comput. Appl.*, vol. 6, no. 19, pp. 33–38, 2013.

[23] Y. Kumar, R. Munjal, and H. Sharma, “Comparison of Symmetric and Asymmetric Cryptography with Existing Vulnerabilities and Countermeasures,” *Int. J. Comput. Sci. Manag. Stud.*, vol. 11, no. 3, pp. 60–63, 2011.

[24] A. P. U. Siahaan, “Genetic Algorithm in Hill Cipher Encryption,” *Am. Int. J. Res. Sci. Technol. Eng. Math.*, vol. 15, no. 1, pp. 84–89, 2016.

[25] A. P. U. Siahaan, “Three-Pass Protocol Concept in Hill Cipher Encryption Technique.”

[26] S. Garg and M. K. Rana, “A Review on RSA Encryption Algorithm,” *Int. J. Eng. Comput. Sci.*, vol. 5, no. 7, pp. 17148–17151, 2016.

[27] Hung-Min Sun, Mu-En Wu, Wei-Chi Ting, and M. J. Hinek, “Dual RSA and Its Security Analysis,” *IEEE Trans. Inf. Theory*, vol. 53, no. 8, pp. 2922–2933, Aug. 2007.