Improvement of RSA Algorithm Using Euclidean Technique

R. Felista Sugirtha Lizy[a], V. Joseph Raj[b]

[a]Research Scholar of Computer Science, Kamaraj College, Thoothukudi – 628 003, TamilNadu, India, Affiliated to Manonmaniam Sundaranar University
[b]Associate Professor and Head, Department of Computer Science Kamaraj College, Thoothukudi – 628 003, TamilNadu, India Affiliated to Manonmaniam Sundaranar University
[a]21felistaa@gmail.com, [b] v.jose08@gmail.com

Article History: Received: 10 November 2020; Revised 12 January 2021 Accepted: 27 January 2021; Published online: 5 April 2021

Abstract: Information Security has become an essential concern in the modern world. Encryption is an effective way to prevent an unofficial person from viewing the digital information with the secret key. RSA encryption is often used for digital signatures which can prove the authenticity and reliability of a message. As RSA encryption is less competent and resource-heavy, it is not used to encrypt the entire message. If a message is encrypted with a symmetric-key RSA encryption it will be more efficient. Under this process, only the RSA private key will be able to decrypt the symmetric key. The Euclidean algorithm is attainably one of the most extensively known algorithms. The Euclidean algorithm is used for finding the greatest common divisor of two numbers. The algorithm can also be defined for more general rings than just the integers. This work is very useful to improve the data security in Smart card and Aadhaar card. In this paper, the RSA algorithm is modified using the Euclidean technique to improve its performance. The proposed algorithm shows its better performance in terms of speed, throughput, power consumption, and the avalanche effect. Experimental results and mathematical justification supporting the proposed method are reported.

Index Terms: Decryption, Encryption, Euclidean, RSA, Security

1. Introduction

RSA (Rivest-Shamir-Adleman) is an algorithm used by contemporary computers to encrypt and decrypt messages. It is an asymmetric cryptographic algorithm.

Asymmetric means that there are two different keys: Public Key and Private Key.

The RSA algorithm is the base of a cryptosystem a set of cryptographic algorithms that are used for explicit safety services or resolve which allows public-key encryption and is broadly used to safeguard the data, mainly when in actuality it is directed over an insecure network such as the internet.

RSA was first publicly described in 1977 by Ton Rivest, Adi Shamir, and Leonard Adleman of the Massachusetts Institute of Technology.

Public key cryptography, also well-known as asymmetric cryptography, uses two dissimilar but mathematically related keys one public and one private. The public key is known to everyone, whereas the private key must be kept secret.

In RSA cryptography [1], both the public and private keys can encrypt a message; the opposite key from the one used to encrypt a message is used to decrypt it. This quality is one reason why RSA has become the most extensively used asymmetric algorithm. It provides a method to declare the confidentiality, integrity, authenticity, and non-repudiation of electronic communications and data storage.

2.Related Work

Cryptography [2] is the method of altering data from a human-readable form to an altered form, and then back to its original readable form.

Several symmetric algorithms [3] have been used in the past. These include Blowfish, DES, 3DES (Triple DES), AES.

Blowfish is yet another algorithm planned to replace DES. This symmetric cipher splits messages into blocks of 64 bits and encrypts them independently [4].

Blowfish is known for both its tremendous speed and overall effectiveness as many claims that it has never been defeated.

The Data Encryption Standard (DES) [5] is a secret key encryption scheme implemented as the standard in the USA in 1977.

Triple-DES [6] was intended to substitute the original Data Encryption Standard (DES) algorithm, which hackers ultimately cultured to conquer with relative ease. At one time, Triple DES was the suggested standard and the utmost broadly used symmetric algorithm in commerce.

The Advanced Encryption Standard (AES) is the standard confidential algorithm [7] of the US Government and several administrations.

There are three asymmetric algorithms in use today: Diffie-Hellman, RSA, and Elliptic Curve.

Diffie-Hellman (DH) [8,9] key interchange algorithm is a technique for firmly swapping cryptographic keys over a public communications network. Keys are not swapped – they are together derived. It is named after their originators Whitfield Diffie and Martin Hellman.

The Rivest-Shamir-Adleman (RSA) algorithm is the cryptography system that is used for public-key cryptography. It is normally used for directing safe, sensitive data than an unsecured network like the internet. The RSA algorithm is standard because it permits both public and private keys to encrypt messages. So their secrecy and legitimacy remain complete.

Elliptic Curve Cryptography (ECC) [10,11] is a public key encryption technique based on an elliptic curve system that can be used to make quicker, slighter, and more effective cryptographic keys.

3.Improved Rsa Using Euclidean

Technique

The Euclidean Algorithm is a technique for speedily discovering the GCD of two integers [12,13,14,15,16].

3.1 RSA Algorithm

RSA algorithm is used to hide and retrieve the data in an insecure network environment. The advantage of the RSA algorithm is to increase security and accessibility. The private keys are never required to be transferred or exposed to everybody. In a shared-key cryptographic system, the secret keys must be shared since the same key is used for encryption and decryption. So there may be a chance that an intruder can find the secret key during the transmission. There are so many limitations present in the RSA algorithm.

3.2 Euclidean Technique

GCD of two numbers is the biggest number that divides both of them. A simple way to find GCD is to factorize both numbers and multiply common factors.

3.3 Euclidean Algorithm for GCD

The algorithm is based on the facts below:

1. If we subtract the lesser number from bigger (we decrease bigger number), GCD doesn’t change. So if we retain subtracting frequently the bigger of two, we end up with GCD.
2. Now in its place of subtraction, if we divide the lesser number, the algorithm stops when we find remainder zero.

The Euclidean algorithm is used to improve the RSA algorithm by the modification of enhancing its performance in terms of Avalanche Effect, Speed, Throughput, and Power Consumption.

4. Experimental Results

The research was completed with the input file size varying from 226 bytes to 289 bytes. Each file size is considered ten times and the calculated average of the ten values is accepted. A Laptop with Intel(R) Celeron(R) CPU3865U@1.80GHz 1.80GHZ is used in which the performance data are added.

The performance metrics were the encryption time, decryption time, execution speed, encryption throughput, decryption throughput, execution throughput, and the avalanche effect. The Improved RSA algorithm using Euclidean technique is combined with RSA algorithm and executed using MATLAB.

The investigational results of numerous performance matrices for the Euclidean-RSA algorithm are detailed below.

4.1 Encryption Time

By analyzing, Figure 4.1 shows that the average encryption time for using Euclid techniques in the RSA algorithm is the least which is compared to the RSA algorithm with the GCD technique. The results are given in Table 4.1.

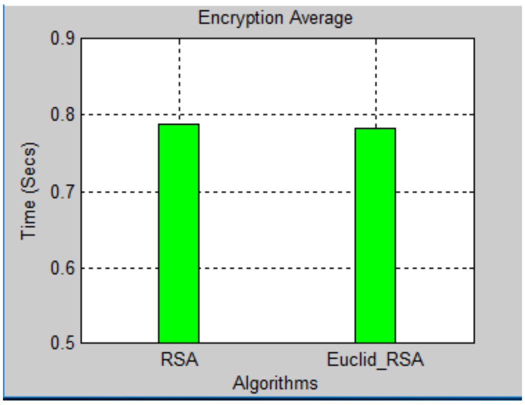

Figure 4.1. Analysis comparison of Average Encryption Time

Table 4.1. Comparative analysis of different input size on Encryption Times (in Secs)

Input Size in Bytes	RSA	Euclid_ RSA
226	0.602258	0.618069
252	0.744034	0.710683
253	0.723753	0.731738
263	0.782120	0.770260
268	0.791476	0.787963
270	0.805340	0.792392
279	0.866912	0.844203
280	0.834226	0.849524
282	0.862093	0.843578
289	0.871534	0.882420
Average Time (Secs)	0.7883746	0.783083

4.2 Decryption Time

According to Figure 4.2, the RSA algorithm with the Euclid technique consumed low memory compared to the RSA algorithm using the GCD technique. The results are as given in Table 4.2.

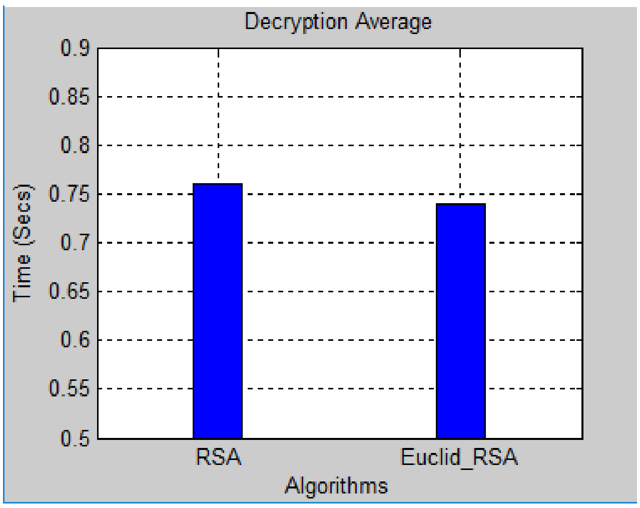

Figure 4.2. Analysis comparison of Average Decryption Time

Table 4.2. Comparative analysis of different input size on Decryption Times (in Secs)

Input Size in Bytes	RSA	Euclid_ RSA
226	0.560304	0.569633
252	0.693223	0.682798
253	0.782367	0.685634
263	0.775782	0.770414
268	0.764340	0.722343
270	0.769396	0.763860
279	0.834860	0.810662
280	0.809466	0.798579
282	0.794328	0.790252
289	0.824921	0.799859
Average Time (Secs)	0.7608987	0.7394034

4.3 Execution Time

According to Figure 4.3, the RSA algorithm with the Euclid technique, clearly defines that the execution time for the RSA algorithm with the Euclid technique is the least when compared to the RSA algorithm with the GCD technique. The results are detailed as shown in Table 4.3.

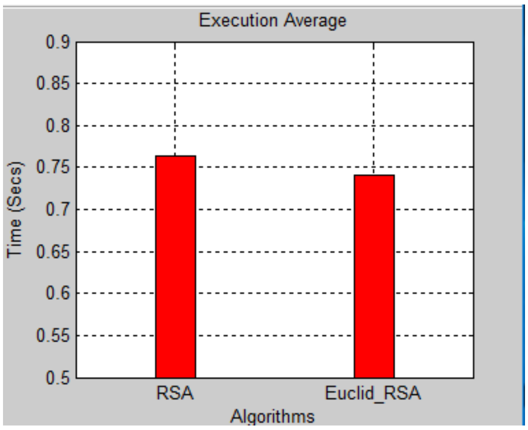

Figure 4.3. Analysis comparison of Execution Time

Table 4.3. Comparative analysis of different input size on Execution Times (in Secs)

Input Size in Bytes	RSA	Euclid_ RSA
226	0.573907	0.570459
252	0.694055	0.683565
253	0.783417	0.686687
263	0.776696	0.771315
268	0.765315	0.723363
270	0.772589	0.764962
279	0.836048	0.811499
280	0.810452	0.799561
282	0.795308	0.791311
289	0.826452	0.801670
Average Time (Secs)	0.7634239	0.7404392

4.4 Throughput

Figure 4.4, clearly shows that the Euclid-RSA algorithm has the highest encryption Throughput when compared to the RSA algorithm with GCD. The results are shown in Table 4.4.

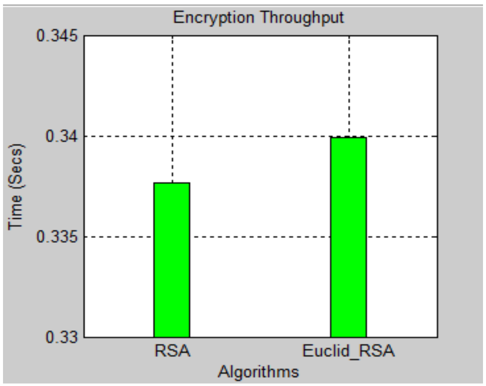

Figure 4.4. Analysis comparison of Encryption Throughput

Figure 4.5 shows that the Euclid-RSA algorithm has the highest decryption Throughput when compared to the GCD in the RSA algorithm. The results are detailed as given in Table 4.4.

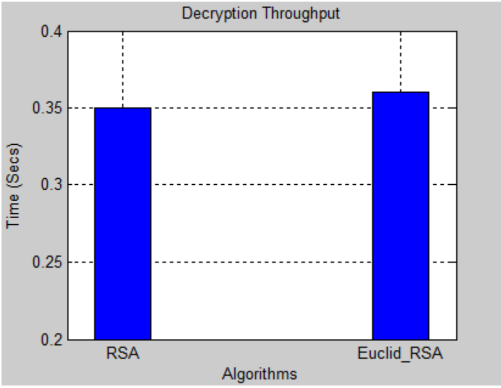

Figure 4.5. Analysis comparison of Decryption Throughput

Table 4.4. Performance Metrics on RSA and Euclid-RSA Algorithm

Input Size in Bytes	**RSA**			**Euclid_RSA**		
	ET	**DT**	**EXT**	**ET**	**DT**	**EXT**
226	0.602258	0.560304	0.573907	0.618069	0.569633	0.570459
252	0.744034	0.693223	0.694055	0.710683	0.682798	0.683565
253	0.723753	0.782367	0.783417	0.731738	0.685634	0.686687
263	0.782120	0.775782	0.776696	0.770260	0.770414	0.771315
268	0.791476	0.764340	0.765315	0.787963	0.722343	0.723363
270	0.805340	0.769396	0.772589	0.792392	0.763860	0.764962
279	0.866912	0.834860	0.836048	0.844203	0.810662	0.811499
280	0.834226	0.809466	0.810452	0.849524	0.798579	0.799561
282	0.862093	0.794328	0.795308	0.843578	0.790252	0.791311
289	0.871534	0.824921	0.826452	0.882420	0.799859	0.801670
Average	0.7883746	0.7608987	0.7634239	0.783083	0.7394034	0.7404392
Throughput (KB/Secs)	0.337656743	0.349849461	0.348692253	0.3399388423	0.360019984	0.359516352

Figure 4.6 shows clearly that the Euclid-RSA algorithm has the highest execution Throughput when compared to the RSA algorithm. The results are in Table 4.4.

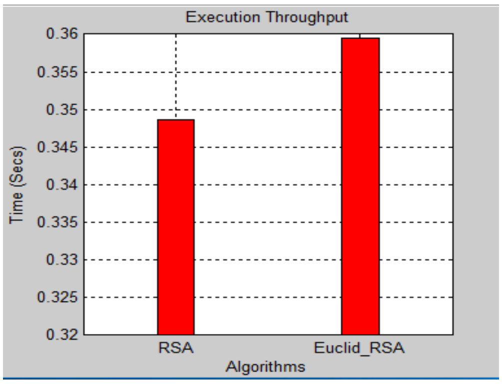

Figure 4.6. Analysis comparison of Execution Throughput

4.5 Power Consumption

The higher the Throughput the lesser will be the power consumption. So from the above findings, it is clear that the power consumption will be the least for the Euclid-RSA algorithm which has the highest Execution Throughput when compared to the RSA algorithm with the GCD technique.

4.6 Avalanche Effect

Figure 4.7 shows that the RSA algorithm has the lowest Avalanche effect when compared to the RSA algorithm using the Euclid technique. The results are detailed in Table 4.5.

Table 4.5. Analysis comparison of Avalanche Effect

Encryption Technique	**Avalanche Effect**
RSA	50
Euclid_RSA	54

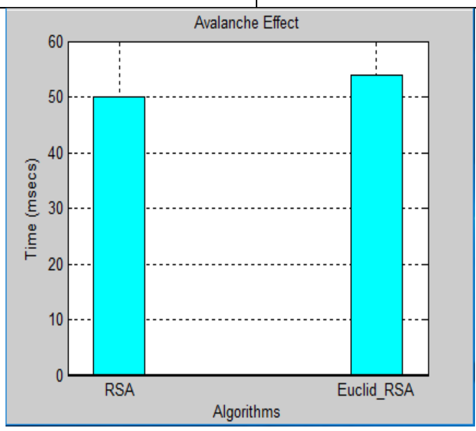

Figure 4.7. Analysis comparison of Avalanche effect

5. Conclusion

The better performance of the Euclid-RSA is compared to the RSA algorithm with the GCD technique. This paper, clearly indicates that the Avalanche effect, Encryption, Decryption, Throughput, and power consumption of Euclid-RSA are more efficient than the RSA algorithm with the GCD technique. It is for the future to speed up the RSA with reduced exponents and the most favorable parameters of the new variant to get good performance and security

References

Dilbag Singh and Ajit Singh, “A Secure Private Key Encryption Technique for Data Security in Model Cryptosystem”, BIJIT Journal, ISSN 0973-5658, Vol. 2, BIJIT 2010, pp. 251-254, 270.

Nehha Mishra, Shahid Siddiqui, and Jitwst P. Tripathi, “A Compendium Over Cloud Computing Cryptographic Algorithms and Security Issues”, BIJIT Journal, ISSN 0973-5658, Vol. 7, BIJIT 2015, pp.810-814.

Ramesh G and Umarani R, “A Comparative Study of Six most Common Symmetric Encryption Algorithms across Different Platforms”, International Journal of Computer Applications, Vol.46, No.13, May 2012.

J.C. Butcher, “A History of Runge-Kutta Methods”, Elsevier, Applied Numerical Mathematics, Vol. 20, 1996, pp. 247-260.

U.S. National Bureau of Standards, “Data encryption standard”, U.S. Fed. Inform. Processing Standards Pub., FIPS PUB 46, January 1977, pp. 2-27.

A. Nadeem, “A performance comparison of data encryption algorithms”, IEEE information and communication technologies, pp.84-89, 2006.Bn.

Atul Kahte, “Cryptography and Network Security”, Tata Mcgraw Hill, 2007.

William Stallings, “Cryptography and Network Security”, Fifth Edition, Pearson Education, 2011, pp. 119-120.

M.K.Jain, S.R.K. Iyengar and R.K.Jain, “Numerical Methods for Scientific and Engineering Computation”, Fifth Edition, New Age International Publishers, 2007, pp. 438-445.

L.F. Shampine and H.A.Watts, “Comparing Error Estimators for Runge-Kutta Methods”, Mathematics of Computation, Vol. 25, Number 115, July 1971, pp.445-455.

S.S.Sastry, “Introductory Methods of Numerical Analysis”, Fourth Edition, 2009, pp. 304-306.

E. Balagurusamy, “Numerical Methods”, Tata McGraw-Hill Education Private Limited, pp. 436-437.

S.R.K.Iyengar and R.K.Jain, “Numerical Methods”, First Edition, New Age International Publishers, 2009, pp. 200-203.

Ashok Kumar and T. E.Unny, “Application of Runge-Kutta method for the solution of non-linear partial differential equations”, Applied Mathematical Modelling, Elsevier, Vol.1, Issue4, March 1977, pp. 199-204.

V. Josephraj and B. Shamina Ross, “Enhancing the Performance of Blowfish Encryption Algorithm in Terms of Speed and Security By Modifying its Function”, International Journal of Applied Engineering Research, ISSN 0973-4562 Vol. 10 No. 79, 2015, pp. 621-624.

V. Josephraj and B.Shamina Ross, “Security Evaluation of Blowfish and Its Modified Version Using GT’s One-Shot Category of Nash Equilibrium”, International Journal of Control Theory and Applications, ISSN 0974-5572, 2016, pp.4771-4777