An examination of public key ‘e’ in the RSA algorithm

Sachin Agrawal,
Assistant Professor, BCA Department
DPBS College Anupshahr
Mayank Sharma,
Assistant Professor, BCA Department
DPBS College Anupshahr
Dr. P.K. Tyagi,
Professor, Statistics Department
DPBS College Anupshahr

How to Cite this Article:
Agrawal, S. & Sharma, M. (2026). An examination of public key ‘e’ in the RSA algorithm. International Journal of Creative and Open Research in Engineering and Management, 02(05). https://doi.org/10.55041/ijcope.v2i5.504

https://doi.org/10.55041/ijcope.v2i5.504

Abstract

RSA Algorithm is a public-key cryptography method vulnerable to factorization attacks that leverage the public key ‘e’ and modulus ‘n’. This research focused on altering an RSA Algorithm by generating various values for the public key ‘e’ and demonstrating different outcomes based on the original RSA Algorithm equation.

Introduction

The advancement of internet applications has progressed rapidly, yet it requires a secure cryptosystem for data transmission. To achieve user satisfaction, it is necessary to create high-speed algorithms with improved security .

Public key cryptosystem offers a more effective solution for secure data communication, encompassing encryption, decryption, and authentication, designed to enhance data security and privacy . There are two types of encryption to choose from: symmetric and asymmetric. Los algoritmos de cifrado simétrico son AES [2], DES [4], RC5 [5], RC6 [6] y Blowfish [7]. Asymmetric encryption methods include RSA [8], ECC [9], and Diffie-Hellman [10].

The RSA algorithm stands out as the optimal choice in asymmetric cryptography, not just due to its widespread use but also because of its extensive range of applications[11]. It enhances security by utilizing large prime numbers for generating the public key, private key 'd,' and modulus 'n' [12]. The Rivest Shamir Adleman (RSA) algorithm relies on public-key cryptography and is regarded as one of the significant advancements in IT security because of its robustness against various attacks [13].
Der RSA-Algorithmus ist die am häufigsten verwendete asymmetrische Verschlüsselung, und die Nachfrage nach besserer Sicherheit stellt eine der Herausforderungen dar, um das Vertrauen der Nutzer aufrechtzuerhalten [14]. Despite its advantages, RSA has issues with slow performance, a requirement for key storage, and is often unsuitable for various systems [3]. Due to the speed and security concerns, researchers are urged to suggest new algorithms with various parameters for improved performance in secure network communication [1].
In the RSA Algorithm indicates that the public key 'e' can effectively factor n accordingly [15].

Boundaries and Constraints

The research confines its focus to the alteration of the RSA Algorithm using the public key 'e'. The comparison will focus solely on the Original RSA Algorithm and the Modified RSA Algorithm. The aim of the research is to highlight the significance of the public key 'e' in the alteration of the RSA Algorithm.

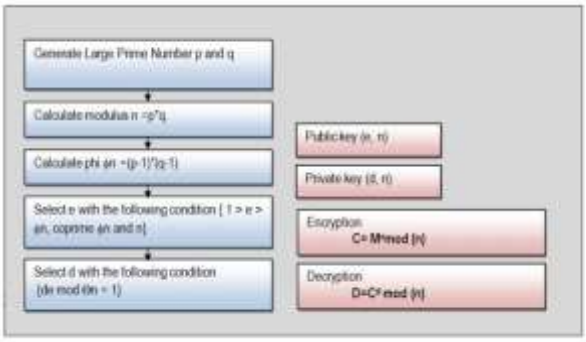

Figure 1

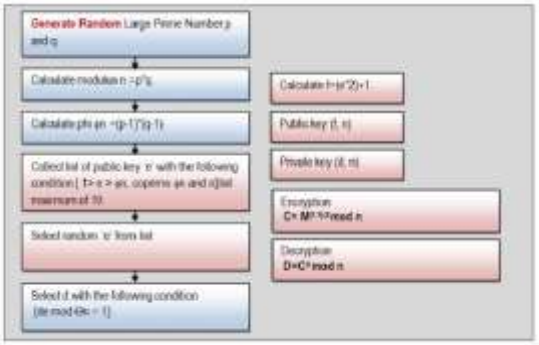

Figure 2

Figure 1 illustrates the process flow of the Original RSA Algorithm. Figure 2 illustrates the Revised RSA Algorithm from the research. The public key is first altered, and then the encryption formula is applied.

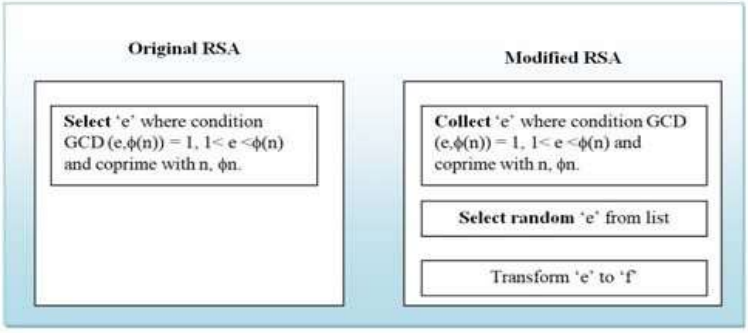

Figure 3

Figure 3.TheModificationonPublicKey'e'

A. KeyGeneration

1. Selectlargeprimenumberspandq.
2. Computen=p*q.
3. Computetheϕ(n)=(p-1)x(q-1).
4. Selectpublickeye.

GCD(e,ϕ(n))=1,1<e<ϕ(n)andcoprimewithn,ϕn.

5. Computewherecondition“de(mod(phin))=1”

B. Encryption

C=Memodn

C. Decryption

$D=C^{d}mod(n)$

The steps below will show the modification process.

ModifiedRSAAlgorithmBasedonPublicKey‘e’

TheproposedstudywastomodifythePublicKey‘e’valueformoresecureRSAAlgorithm.

A. KeyGeneration

1. Selectlargeprimenumberspandq.
2. Computen=p*q.
3. Compute ϕn=(p-1)*(q-1)
4. Collect e with the following condition { p > e >ϕn, coprime ϕn and n}(maximum of tenvalues)
5. Selectrandom‘e’fromlist
6. Calculatef=(e*2)+1.
7. Selectdwiththefollowingcondition{demodn=1}
8. SendPublickey(f,n)

The‘f’serveasanewpublickeywhichwillhidetheoriginalevalue.

9. SendPrivatekey(d,n)

B. Encryption

$C = M^{((f-1)/2)}mod(n)$

C. *Decryption*

$D = C^{d}mod(n)$

Sample

A. KeyGeneration

1. Select large prime numbers p and q . p=7 q=11
2. Compute modulus n=p*q.n

=7*11

=77

3. Compute ϕn =(p-1)*(q-1)

=(7-1)*(11-1)

=6*10

=60

4. Collect list of public key ‘e’ with the following condition { p > e >ϕn, coprime ϕn and n}e={13,17,23,27}(maximum of ten values)
5. Select Random‘e’from the list.

e=23

6. Calculatef=(e*2)+1.

=(23*2)-1

= 46+1

=47

7. Select d with the following condition {de mod n = 1} d=47
8. Send Public key(f,g) Public Key=(47,77)
9. Send Private key (d,g)Private Key=(47,77)

B. Encryption M=4

$C = M^{(f-1)/2} \bmod(n)$

$= 4^{(47-1)/2} \bmod 77$

$= 4^{23} \bmod 77$

$= 70,368,744,177,664 \bmod 77$

$= 9$

D. Decryption

$D = C^{d} \bmod(n)$

$= 9^{47} \bmod 77$

$= 7.0696504901e+44 \bmod 77$

$= 4$

The Original RSA Algorithm and the Modified RSA Algorithm Based on Public Key Generation were created using the Java Programming Language. An enhanced key generation method of the RSA Algorithm will be applied to bolster its security. The technique for altering the public key equation will be applied in relation to the encryption and decryption process involved. The approach to evaluate the two algorithms is to distinguish the outcome of RSA in producing the public key 'e' between the Original RSA Algorithm and the Modified RSA Algorithm. Additionally, the two algorithms will undergo testing under identical hardware and software specifications.

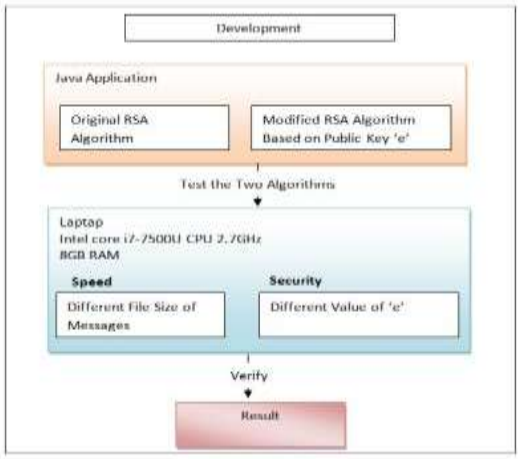

Figure4.TheMethodoftheStudy

Outcome

The RSA Algorithm in both its Original and Modified forms was implemented using the Java Programming Language. The two algorithms were evaluated using various message file sizes and measured their efficiency regarding the encryption and decryption processes. Each algorithm performed tests to generate cipher text based on formulas derived from different related research studies.

Outcome for Public Key 'e'

The initial test involved creating various values of the private key 'd' and the ciphertext based on distinct values of the public key 'e' and the message 'm'. The outcome of the ciphertext from the Original RSA and the Modified RSA Algorithm on public key 'e' follows the formula "C=Me mod n" [16] [3][19].
The next three tables display the various outcomes of cipher text with distinct values of the public key 'e' in the Original RSA Algorithm.

Table1.p=7q=11

		cipher		
e	d	m=4	m=5	m=6
13	37	53	26	62
17	53	16	3	41
23	47	9	59	62

Table2.p=5q=11

		cipher		
e	d	m=4	m=5	m=6
13	37	9	15	51
17	33	49	25	41
23	7	9	15	51
27	3	49	25	41
29	29	14	25	29

Table3.p=11q=17

		cipher		
e	d	m=4	m=5	m=6
3	107	64	125	29
7	23	115	166	184
13	37	174	37	95
19	59	47	108	46
29	149	157	146	104

Tables 4 and 5 display the varying results of cipher text with different values of the public key 'e' in both the Original RSA and the Modified RSA Algorithm.

Table4.Resultforp=11q=17

			Cipher	
e(Original RSA)	e(Modified RSA)	d	Original RSA	Modified RSA
3	7	107	64	115
7	15	23	115	166
13	27	37	174	115
19	39	59	47	0
29	59	149	157	0

Table5.Resultforp=5q=11

			cipher		
e(Original RSA)	e(Modified RSA)	d	Original RSA	Modified RSA	A
13	27	37	9	49	49
17	35	33	49	0	0
23	47	7	9	0	0
27	55	3	49	0	0
29	59	29	14	0	0

Outcome for Effectiveness

The tables and graphs below illustrate the performance outcomes of both the original and modified RSA Algorithm. The two algorithms are implemented in Java Programming Language.

Table6.EncryptionTime

Encryption		
	Time in Second/s	
Size(KB)	Original	Modified
10KB	0	1
1000KB	2	3
5000KB	8	10
10,000KB	16	18
20,000KB	28	31
50,000KB	71	75
100,000KB	146	156

Table7.DecryptionTime

Decryption		
	Time in Second/s	
Size(KB)	Original	Modified
10KB	0	0
1000KB	2	2
5000KB	7	7
10000KB	16	16
20000KB	25	26
50,000KB	62	65
100,000KB	152	159

Results and Examination

Results

The subsequent outcomes and discoveries of this research are:
In Tables 1, 2, and 3, the results indicate that varying the public key 'e' corresponds to a distinct private key value, and different values of public key 'e' will yield different cipher text values.
In Table 4 and Table 5, the results indicate that the Modified RSA Algorithm generates distinct cipher text values compared to the original RSA Algorithm in the encryption process.

The Modified RSA Algorithm experienced a minor increase in its time complexity during encryption. During the decryption phase, the Modified RSA Algorithm exhibits a time execution nearly identical to that of the Original RSA Algorithm.

Results

Examination

According to the study's findings, various valid values of the public key ‘e’ correspond to distinct values of the private key ‘d’ and yield different cipher values. Consequently, the security level in the encryption and decryption processes enhances. Regarding the transformation of the chosen value of public key ‘e,’ it generates a different cipher value, enhancing the security in the encryption and decryption process.

Regarding performance, changing the public key ‘e’ by gathering the public key list and converting it to a new value will marginally elevate the time complexity during the key generation and encryption phases of the RSA Algorithm, but it will not affect the decryption phase.

Conclusion

This research showcased various public key 'e' values that were assessed during the encryption and decryption procedures. The conclusion is derived from the two primary categories focused on security and performance, comparing the Modified RSA Algorithm with the Original RSA Algorithm.

As a result of performance evaluation, the modified RSA algorithm shows a slight rise in time complexity during the encryption process. However, in the decryption process, the Modified RSA Algorithm takes nearly the same amount of time as the original RSA Algorithm.

In the outcome related to security, the Modified RSA Algorithm using Public Key 'e' generated a more intricate result regarding the encryption process compared to the Modified RSA Algorithm. Varying the public key value ‘e’ resulted in different private key and cipher values, and transforming these into new values generated more complex cipher results, enhancing the security of the Modified RSA Algorithm based on Public Key ‘e’ and improving the decryption process's security.

Ultimately, this study indicated that in altering the RSA Algorithm, the Public Key 'e' is a factor that bolsters the security during the encryption and decryption process.

Future Improvement

This study's implementation employs the limited public key list ‘e’. The subsequent phase of this project involves applying this to the larger list and executing the RSA algorithm modification using the public key ‘e’ alongside the modified modulus ‘n’, both of which are made public to enhance the RSA Algorithm's security against factorization attacks, including the processes of encryption and decryption. Additionally, the Modified RSA Algorithm utilizing public key ‘e’ and modulus ‘n’ will be evaluated against other Modified RSA Algorithms in terms of security and efficiency.

References

[1] A. S. Alkalbani, T. Mantoro, and A. O. M. Tap, "Comparison between RSA hardware and software implementation for WSNs security schemes," *Proceeding 3rd Int. Conf. Inf. Commun. Technol.Moslem World ICT Connect. Cult. ICT4M 2010*, 2010.

[2] A. Biryukov and E. Kushilevitz, "Improved cryptanalysis of rc5," in *Lecture Notes in ComputerScience (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 1998.

[3] R. Rivest, M. J. B. Robshaw, R. Sidney, and Y. L. Yin, "The RC6 Block Cipher," *First Adv.Encryption ...*, 1998.

[4] B.Schneier,"The{Blowfish}encryptionalgorithm,"*Dr.Dobb'sJ.Softw.Tools*,1994.

[5] A.Mansour,A.Davis,M.Wagner,andR.Bassous,"MultisymmetricCryptographicRSAScheme," pp. 1–8, 2017.

[6] P. Patel, R. Patel, and N. Patel, "Integrated ECC and Blowfish for Smartphone Security," in*PhysicsProcedia*, 2016.

[7] W.Diffie,W.Diffie,andM.E.Hellman,"NewDirectionsinCryptography,"*IEEETrans.Inf. Theory*, 1976.

[8] F.LombardiandR.DiPietro,"Transparentsecurityforcloud,"in*Proceedingsofthe2010ACM Symposium on Applied Computing - SAC '10*, 2010.

[9] R.TripathiandS.Agrawal,"ComparativeStudyofSymmetricandAsymmetricCryptography,"*Int.J.Adv.Found.Res.Comput.*,vol.1,no.6,pp.2348–4853,2014.

[10] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-keycryptosystems," *Commun. ACM*, 1978.

[11] P.K.Arya,M.S.Aswal,andV.Kumar,"ComparativeStudyofAsymmetricKeyCryptographic Algorithms," *Int. J. Comput. Sci. Commun. Networks,* vol. 5, no. 1, pp. 17–21, 2012.

[12] D.AggarwalandU.Maurer,"BreakingRSAGenericallyIsEquivalenttoFactoring,"vol.62,no.11, pp.6251–6259,2016.

[13] R.S.Dhakar,"Modified RSAEncryptionAlgorithm(MREA),"pp.2–5,2012.

[14] J. Sahu, V. Singh, V. Sahu, and A. Chopra, "An Enhanced Version of RSA to Increase the Security," vol. 7, no. 4, pp. 1–4, 2017.