Int. J. Nonlinear Anal. Appl. 13 (2022) 1, 955-961
ISSN: 2008-6822 (electronic)
http://dx.doi.org/10.22075/ijnaa.2022.5614

Optimizing RSA cryptosystem using Hermite polynomials

Raghad K. Salih[a,*]

[a]Department of Applied Sciences, University of Technology, Baghdad, Iraq

(Communicated by Madjid Eshaghi Gordji)

Abstract

The Security is requested to relocate paramount information across the networks. To protect the confidential data from hacking, this paper describes a process to increase the security of the RSA algorithm by creating additional layer protection for it using Hermite polynomials which will be represented as a square matrix, its calculation is not complicated. In the RSA process, we need to choose very large numbers that lead to complex operations which require a long computation time, while in proposed encryption due to Hermite key we don't need that because two layers of encryption give robust safeness to the ciphertext from dangers of hackers due to the hard of breakable.

Keywords: RSA system, layers and Hermite polynomials.

1. Introduction

Cryptography safes electronic secrete data while transported and only the meant recipients can see them. RSA is the public key cryptosystem. The safety of RSA process counts on factoring of very large numbers where the generating of RSA key is specified by too big prime numbers, So, it is hard to get a password every time. To grantee the safety of the ciphertext, we need the calculation cost to be very high which takes a long time to execute [2, 9].

In 2014, [10] Modified RSA algorithm via using four large prime numbers instead of two large primes which boosts the intricacy, security and hardness for breakable comparing with the traditional RSA technique.

In 2016, [5] Studied the power, feebleness, cost and effectiveness of algorithms RSA, DES, AES 3DES and blowfish to offer a comprehensive analysis of achievement, as opposed to only theoretical

*Corresponding author
Email address: Raghad.k.Salih@uotechnology.edu.iq (Raghad K. Salih)

Received: May 2021 *Accepted:* September 2021

comparisons. [6] improved a variant of a public key algorithm utilizes commutation key of Diffie-Hellman protocol and the features of magic square.

In 2019, [7] computed the number of lattices in a sphere in four dimension which play a significant role in breakable of the RSA technique.

In 2021, [8] offered hybrid encryption from playfair and RSA cipher systems to boost the security of the ciphertext from meddlers. [1] showed an attack on partial key of RSA using modulus $N=p^2q$ which procures it vulnerable.

There is a lot of research aimed at enhancing cryptographic systems to be stricter and more difficult for hackers and crypto analysts. One of this method that leads to a difficult-to-crack encryption system is to provide two layers of protection for the ciphertext. In this paper we improve the RSA system by providing an additional encryption layer using Hermite polynomials as a matrix with uncomplicated calculations that ensure adequate security of the ciphertext from breachers and brute force.

2. Hermite polynomials

Hermite polynomials are one of the classical orthogonal functions. The general form of Hermite polynomials $H_n(x)$ is [3, 4]:

$$H_{n+1}(x) = 2xH_n(x) - 2nH_{n-1}(x) \qquad n \geq 1 \tag{2.1}$$

where $H_0(x) = 1$ and $H_1(x) = 2x$.

The Hermite polynomials up the six$^{\text{th}}$ limit are demonstrated in Eq. (2.2) and Figure 1.

$$\begin{aligned} H_0(x) &= 1, \\ H_1(x) &= 2x, \\ H_2(x) &= 4x^2 - 2, \\ H_3(x) &= 8x^3 - 12x, \\ H_4(x) &= 16x^4 - 48x^2 + 12, \\ H_5(x) &= 32x^5 - 160x^3 + 120x, \\ &\vdots \end{aligned} \tag{2.2}$$

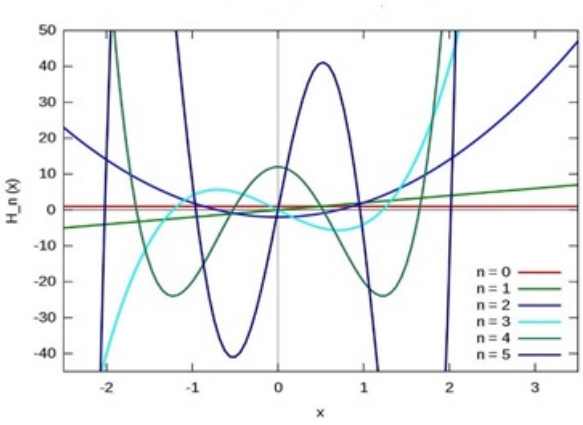

Figure 1: The six limits of Hermite polynomials

3. The proposed Model

In the proposed model, we will put the ASCII characters of plain text into a square matrix and the rest places of the matrix is filled with numbers 128,129, ... Then we encrypt it with two keys, the public key of the RSA algorithm K_1=(e,t), $t = p \times q$ where p and q are prime numbers and the other key is the Hermite polynomials which is:

$$HK_2 = [n1, n2, \ldots, nm] = [H_{n1} + H_{n2} + \ldots + H_{\mathrm{nm}}] = a_0 + a_1 h + a_2 h^2 + \cdots + a_{\mathrm{nm}} h^{\mathrm{nm}} \tag{3.1}$$

where $n1,\ n2,\ \ldots,\ nm$ are positive integers chosen by agreement with the recipient. The Hermite key K_2 is represented as a square matrix, all their elements are ones except the main diagonal which is filled by the absolute coefficients of HK_2 in Eq. (3.1) as shown in Eq. (3.2).

$$K_2 = \begin{pmatrix} |a_0| & 1 & 1 & \ldots & 1 \\ 1 & |a_1| & 1 & \ldots & 1 \\ 1 & 1 & \ddots & 1 & \vdots \\ \vdots & \vdots & 1 & |a_{nm-1}| & 1 \\ 1 & 1 & \ldots & 1 & |a_{\mathrm{nm}}| \end{pmatrix} \tag{3.2}$$

The following flowchart explains the suggested model.

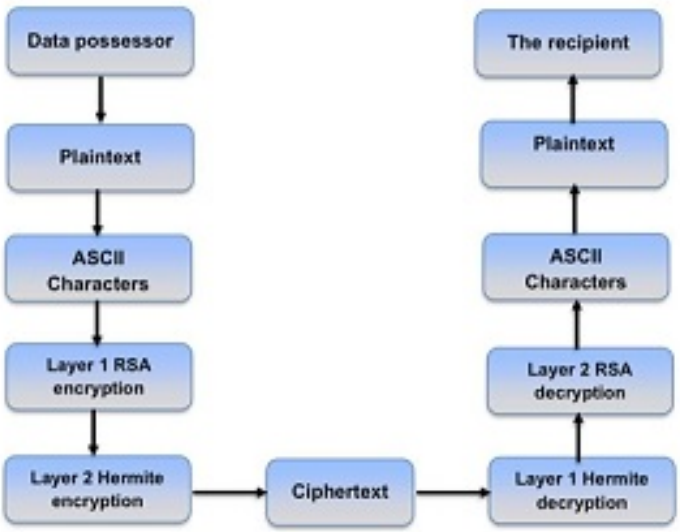

Figure 2: The flowchart of the proposed encryption using Hermite polynomials.

3.1. The Encryption Algorithm (ERSA-Hermite)

Input

- The two prime numbers p and q.
- The Hermite key $HK_2 = [n1\ n2 \ldots nm]$
- The plaintext M.

Output
Ciphertext (Ct).
Procedure

1. Numerate $t = p \times q$ *and* $\varnothing(t) = (p-1) \times (q-1)$
2. Pick out e where $1 < e < \varnothing(t)$ and $\gcd(e, \varnothing(t)) = 1$, K_1=(e,t)
3. Set the plaintext M into a square matrix row by row as illustrated above.
4. Compute the first ciphertext C_1 by using RSA process: $C_1 = M^{*e} \bmod t$, *where* M^{*e} *means each element of the matrix M is raised to the power e.*
5. Form the Hermite key matrix K_2 from HK_2 as shown in Eq. (3.1) and Eq. (3.2).
6. Calculate The ultimate ciphertext Ct as: $Ct = C_1 \times K_2$.

3.2. The Decryption Algorithm (DRSA-Hermite)

In decryption, the private key of RSA K_1^{-1} =(d,t), $d = e^{-1} mod\ \varnothing(t)$ *where* $e \times d (mod\ \varnothing(t)) = 1$ and the inverse Hermite key matrix K_2^{-1} are utilized to get the ultimate plaintext.

Input

- K_1^{-1} =(d,t),
- K_2
- The ciphertext (Ct).

Output*:* The plaintext M.
Procedure

1. Lay the ciphertext Ct into a square matrix row by row.
2. Compute K_2^{-1}
3. Calculate the first plaintext P_1 as: $P_1 = Ct \times K_2^{-1}$
4. Find the final plaintext M by using RSA process:
   $M=(P_1)^{*d}$ *mod t, where* P_1^{*d} *means each element of the matrix* P_1 *is raised to the power d .*

4. Analysis and Results

4.1. Results

Take K_1 = (e, t)=(7,253), Where p=11, q=23 , $t = p \times q = 253$.

$\varnothing(t) = (p-1) \times (q-1)$=10×22=220.

The plaintext M=University of Technology.

The following results were acquired by implementing the ERSA-Hermite algorithm.

M= 85 110 105 118 101 114 115 105 116 121 32 111 102 32 84 101 99 104 110 111 108 111 103 121 128

$$M = \begin{pmatrix} 85 & 110 & 105 & 118 & 101 \\ 114 & 115 & 105 & 116 & 121 \\ 32 & 111 & 102 & 32 & 84 \\ 101 & 99 & 104 & 110 & 111 \\ 108 & 111 & 103 & 121 & 128 \end{pmatrix}$$

$$C_1 = M^{*7} mod\ 253 \quad \Rightarrow \quad C_1 = \begin{pmatrix} 156 & 121 & 239 & 2 & 73 \\ 137 & 69 & 239 & 162 & 187 \\ 142 & 199 & 152 & 142 & 149 \\ 73 & 143 & 223 & 121 & 199 \\ 202 & 199 & 214 & 187 & 193 \end{pmatrix}$$

The Hermite key is

$$\begin{aligned} \mathrm{HK}_2 &= [4,7,10] = [H_4 + H_7 + H_{10}] \\ &= 16x^4 - 48x^2 + 12 + 128x^7 - 1344x^5 + 3360x^3 - 1680x + 1024x^{10} - 23040x^8 + 161280x^6 \\ &- 403200x^4 + 302400x^2 - 30240 \\ &= 1024x^{10} - 23040x^8 + 128x^7 + 161280x^6 - 1344x^5 - 403184x^4 + 3360x^3 + 302352x^2 \\ &- 1680x - 30228 \end{aligned}$$

$$\therefore K_2 = \begin{pmatrix} 30228 & 1 & 1 & 1 & 1 \\ 1 & 1680 & 1 & 1 & 1 \\ 1 & 1 & 302352 & 1 & 1 \\ 1 & 1 & 1 & 3360 & 1 \\ 1 & 1 & 1 & 1 & 403184 \end{pmatrix}$$

$$Ct = C_1 \times K_2 = \begin{pmatrix} 4716003 & 203750 & 72262480 & 7309 & 29432950 \\ 4141893 & 116645 & 72262683 & 544952 & 75396015 \\ 4293018 & 334905 & 45958136 & 477762 & 60075051 \\ 2207330 & 240856 & 67425032 & 407198 & 80234176 \\ 6106849 & 335116 & 64704109 & 629128 & 77815314 \end{pmatrix}$$

The ultimate ciphertext is

Ct= 4716003 203750 72262480 7309 29432950 4141893 116645 72262683 544952 75396015 4293018 334905 45958136 477762 60075051 2207330 240856 67425032 407198 80234176 6106849 335116 64704109 629128 77815314.

In decipherment, DRSA-Hermite algorithm are employed to display the following outcomes:

$$P_{1=}Ct \times K_2^{-1} \rightarrow P_1 = \begin{pmatrix} 156 & 121 & 239 & 2 & 73 \\ 137 & 69 & 239 & 162 & 187 \\ 142 & 199 & 152 & 142 & 149 \\ 73 & 143 & 223 & 121 & 199 \\ 202 & 199 & 214 & 187 & 193 \end{pmatrix}$$

$d = e^{-1} mod \varnothing(t) = 63$
$M=(P_1)^{*63} \ mod \ 253$

$$M = \begin{pmatrix} 85 & 110 & 105 & 118 & 101 \\ 114 & 115 & 105 & 116 & 121 \\ 32 & 111 & 102 & 32 & 84 \\ 101 & 99 & 104 & 110 & 111 \\ 108 & 111 & 103 & 121 & 128 \end{pmatrix}$$

M= 85 110 105 118 101 114 115 105 116 121 32 111 102 32 84 101 99 104 110 111 108 111 103 121 128. Hence, the original plaintext M=University of Technology

4.2. Performance Analysis

The ERSA-Hermite algorithm beats the flaws and the intricate computations of the RSA process. In return, a highly confidential encryption was acquired as shown below:

1. In RSA process if the exponent e in encipherment is selected low and the values of the plaintext M is small (i.e. $M < t^{1/e}$), the outcomes of M^e give rise to a strictly lower than modulo t, thus encrypted texts can be readily decrypted by picking the e^{th} root of the ciphertext over the integers. For this reason a very large number has to be chosen which leads to a complex arithmetic as well as a long time to complete while in the proposed algorithm we don't need that because of the Hermite layer which provides protection even if small numbers are used for p,q and t since the second key and other layer of encryption provide sufficient security against attackers.

2. In the Hermite matrix key shown in section 3. Hermite layer is very difficult to know by attacker since the degrees of the polynomials are impossible determined. Moreover, Hermite key does not need much time in the calculations because it depends on the matrices which are easy to compute but hard to break, so the ciphertext is robust enough to withstand offensive utilizing current techniques.

4.3. Security Analysis

- The proposed encryption is impossible to crack by statistical analysis because the ERSA-Hermite algorithm depends on two keys and also two layers, the RSA layer and the Hermite layer which produce a robust security ciphertext. In the proposed scheme, we note that there are no iteration numbers for the ciphertext even though the characters of the plaintext are repeated unlike RSA. Refer to the example in section 4.1, notice the iteration of the ASCII characters 110,105,101, 121 and 111 of the plaintext, iteration more than once, there is no iterate of the numbers in the final ciphertext. This gives more confusion and rigidity to the cryptanalyst. See Table 1.

- In the proposed model, even if the numbers p, q, and d are detected, it is very hard to know the plaintext due to a Hermite layer where it is too hard to penetrate it.

Table 1: Iteration the letters of the plaintext and the corresponding in ciphertext

The iteration letters in plaintext M	**n**	**i**	**e**	**y**	**o**
The number of repetitions	**2**	**2**	**2**	**2**	**3**
The ciphertext using RSA algorithm	**121**	**239**	**73**	**187**	**199**
The ciphertext using ERSA-Hermite algorithm	**203750 407198**	**72262480 72262683**	**29432950 2207330**	**75396015 629128**	**334905 80234176 335116**

5. Conclusion

The present work offers a proposed enciphering algorithm using Hermite polynomials. It has been verified that the suggested algorithm promoted the level of confidentiality of the ciphertext as well as avoiding the drawbacks of RSA scheme which are intricacy, too large numbers and time-consuming computations. Hermite layer equips adequate protection and strong security to the ciphertext.

References

[1] N.N.H. Adenan, M.R. Kamel Ariffin, S.H. Sapar, A.H. Abd Ghafar and M.A. Asbullah, *New Jochemsz–May cryptanalytic bound for RSA system utilizing common modulus N= p2q*, Mathematics, 9 (4) 2021 340.

[2] N.A. Hassan and A.K. Farhan, *Security improves in ZigBee protocol based on RSA public algorithm in WSN*, Engin. Tech. J. 37(3B) (2019) 67–73.

[3] Y. He and F. Yang, *Some recurrence formulas for the Hermite polynomials and their squares*, Open Math. 16(1) (2018) 553–560.

[4] AJ. Kadhim, *Expansion methods for solving linear integral equations with multiple time lags using B-spline and orthogonal functions*, Engin. Tech. J. Signif. 29(9) (2011) 1651–1661.

[5] P. Patil, P. Narayankar, D.G. Narayan and S.M.A. Meena, *comprehensive evaluation of cryptographic algorithms: DES, 3DES, AES, RSA and Blowfish*, Procedia Computer Sci. 78 (2016) 617–624.

[6] A. M. S. Rahma, A. M. J. A. Hossen and O. Dawood, *Public key cipher with signature based on Diffie-Hellman and the magic square problem*, Engin. Tech. J. Part (B) 34(1) (2016).

[7] Sh. A. Salman, *Lattice Point and Its Application in RSA Cryptosystem*, Energy Procedia 157 (2019) 39–42.

[8] R.K. Salih and M. S. Yousif, *Hybrid encryption using playfair and RSA cryptosystems*, Int. J. Nonlinear Anal. Appl. 12(2) (2021) 2345–2350.

[9] R.K. Salih and M.S. Yousif, *Playfair with multi strata encryption*, Iraqi J. Sci. 62 (9) 2021 3237–3242.

[10] M. Thangavel, P. Varalakshmi, M. Murrali and K. Nithya, *An enhanced and secured RSA key generation scheme (ESRKGS)*, J. Inf. Secur. Appl. 20 (2014) 1–10.