AIMS Mathematics

AIMS Mathematics, 8(8): 18833–18845.
DOI: 10.3934/math.2023958
Received: 18 January 2023
Revised: 06 May 2023
Accepted: 16 May 2023
Published: 05 June 2023

http://www.aimspress.com/journal/Math

Research article

Generalization of RSA cryptosystem based on $2n$ primes

Tariq Shah[1], Muhammad Zohaib[1], Qin Xin[2], Bander Almutairi[3] and Muhammad Sajjad[1,*]

[1] Department of Mathematics, Quaid-I-Azam University, Islamabad 45320, Pakistan
[2] Faculty of Science and Technology, University of the Faroe Islands, Faroe Islands, Denmark
[3] Department of Mathematics, College of Sciences, King Saud University, P.O.Box 2455 Riyadh 11451, Saudi Arabia

* **Correspondence:** Email: m.sajjad@math.qau.edu.pk; Tel: +923067759056.

Abstract: This article introduced a new generalized RSA crypto-system based on $2n$ prime numbers called generalized RSA (GRSA). This is a modern technique to provide supreme security for the computer world by factoring the variable N, where its analysis process has become much easier nowadays with the development of tools and equipment. $2n$ primes (prime numbers) are used in the GRSA crypto-system to provide security over the network system. This includes encryption, key generation, and decryption. In this method we used $2n$ primes which are not easily broken, $2n$ primes are not comfortably demented. This method provides greater performance and fidelity over the network system.

Keywords: RSA cryptosystem; generalized RSA cryptosystem; primes; key generation; encryption; decryption; private key; public key
Mathematics Subject Classification: 68P25, 68U15

1. Introduction

Modern computers and the transmission of high technology are an important part of the powerful economy, so it is important to have appropriate security systems and technologies to meet these security demands. Modern security systems and conventions have been evolved that are established on standards, predominantly from well-known establishments such as the Internet Architecture Board (IAB) and Internet Engineering Task Force (IETF). These establishments offer a wide range of security settlement, algorithms, and implementations that give security services and converge the requirements of data privacy, uprightness, and secure transmissions. A significant mechanism for data conservation is utilized cryptography, which overrides many security tools and evolves the science of material encryption and decryption [1]. Cryptography delegates us to securely save secret data or communicate it to insecure networks that no one but the intended recipient can read (Kahn, 1967) [2,3]. Using an important mechanism such as encryption we have limited access to privacy, legitimacy, rectitude, and data. Cryptography distinguishes between private (also known as traditional intelligence) systems and public key cryptographic systems. The private key also called the non-public key or secret key, it has classical history, and is based on the use of a common non-public key encryption and decryption [4,5]. Nonlinear components of a block cipher over vector algebra for symmetric key cryptography are explained by Sajjad and Shah in [6–8].

Many algorithms have been proposed for public key cryptography, the most developed in 1978 by Rivest, Shamir, and Adelman [9,10]. In RSA, security relies on the supposition that it is hard to find the factorization of big numbers and obtain the private key used for decryption. But there are some flaws in the RSA algorithm, decryption is established on $\aleph$, and d the private key is uncomplicated to factor in [11–14]. Public key Cryptography represents a massive change in the field of cryptosystems. It uses two separate keys that are linked together such that the private key may be used to decrypt the message and the public key is used to encrypt the message. Improve security by modifying the RSA algorithm [15], the schema is based on four prime numbers rather than two, with a double encryption and decryption process. Multiple prime numbers increase the factoring time required to obtain the private key.

This research article introduced a new modification of the RSA cryptosystem, which is generalized RSA (GRSA) based on $2n$ different primes. The e_1, e_2, e_3,...e_n are public keys used for encryption, and d_1, d_2, d_3,...d_n are non-public keys used for decryption. This conception is based on $2n$ different primes instead of two primes, which allows a larger encryption exponent from the huge product $\aleph$ to intensify security. Several prime numbers and large encryption exponents increase the factoring time compared to the RSA algorithm.

2. RSA cryptographic algorithm using two primes

In [11], the implementation of RSA focused on three areas: key generation, decryption, and encryption procedure.

2.1. Key generation

The private key and the public key are the two distinct varieties of RSA keys. The major steps in a key generation are shown below;

- Pick two primes $\wp_1$, and $\wp_2$. Let $\aleph$ be the product of $\wp_1$ and $\wp_2$ as $\aleph = \wp_1 \cdot \wp_2$
- Find Euler Phi Function of $\aleph$ as $\varphi(\aleph) = \varphi(\wp_1) \cdot \varphi(\wp_2)$.
- Choose a number e coprime to $\varphi(\aleph)$.
- Find d is the inverse of e as $d.e \equiv 1 \ (mod \ \varphi(\aleph))$.
- Public key $= (\aleph, e)$.
- Private key$= (\aleph, d)$.

2.2. Encryption algorithm

Encryption is performed in RSA using a public key to create cipher text. The steps necessary to decrypt are described as

- The recipient's public key is received $(\aleph, e)$.
- Displays text as a number.
- Determine the Cipher message $C = (T)^e (mod\ \aleph)$.
- Send encoded data.

2.3. Decryption algorithm

In RSA, the private key is used for decryption in order to obtain plain text. These are the decryption steps:

- Calculate $T = (C)^d (mod\ \aleph)$ using a non-public key.
- Eliminates plain text from a number T.

2.4. Example

Key Generation

Let $\wp_1 = 89$, and $\wp_2 = 101$ be the two primes $\aleph = \wp_1 \cdot \wp_2 = 89.101 = 8989$,

$$\varphi(\aleph) = (\wp_1 - 1).(\wp_2 - 1) = 8800$$

We choose $e = 3$ less than and co-prime to $\varphi(\aleph)$.
The inverse of 3 is $3^{-1}\ mod\ (8800) = 5867$. Hence the secret key is $d = 5867$.
Public key= $(8989, 3)$.
Private key= $(8989, 5867)$.

Encryption

Alice sends a message $T = 8765$.
The cipher text calculated by Alice is $C = T^e\ (mod\ \aleph)$.

$$C = (8765)^3\ (mod\ 8989) = 5815$$

Decryption

Bob can retrieve the plain text from cipher text using $T = C^d\ (mod\ \aleph)$.

$$T = (5815)^{5867} (mod\ 8989)$$

Hence, $T = 8765$ is a recovered plain text.

3. RSA cryptographic algorithm using four primes

In this article, we will discuss the RSA algorithm for four primes. The private and public key consists of three components [13]. Let $\aleph$ be the product of primes $\wp_1$, $\wp_2$, $\wp_3$, and $\wp_4$. $(\aleph, e_1, e_2)$ be the components of the public key, where e_1 and e_2 are chosen randomly. Since $\aleph$ is kept as private and public components, given with the information of $\aleph$, is unable to ascertain the value of the four basic prime numbers, which form the basis for calculating the value of $\aleph$, and later e_1

and e_2. $(\aleph, d_1, d_2)$ be the components of the private key, where d_1 is the inverse of e_1 and d_2 is the inverse of e_2. For security purposes, all four selected prime bits are the same length.

3.1. Key generation

The steps for generating the key are given below

- Choose four primes $\wp_1, \wp_2, \wp_3, \wp_4$. And $\aleph$ is the product of $\wp_1, \wp_2, \wp_3, \wp_4$ as $\aleph = \wp_1 \cdot \wp_2 \cdot \wp_3 \cdot \wp_4$.
- Euler phi function of $\aleph$ is $\varphi(\aleph) = \varphi(\wp_1) \cdot \varphi(\wp_2) \cdot \varphi(\wp_3) \cdot \varphi(\wp_4)$.
- Choose e_1, e_2 two coprime to $\varphi(\aleph)$.
- Find d_1 and d_2 which is inverses of e_1 and e_2 respectively $e_1 d_1 \equiv 1\ (mod\ \varphi(\aleph))$ and $e_2 d_2 \equiv 1\ (mod\ \varphi(\aleph))$.
- Public key= $(\aleph, e_1, e_2)$.
- Private key= $(\aleph, d_1, d_2)$.

3.2. Encryption algorithm

With the use of public key $(\aleph, e_1, e_2)$ encryption is broken. The encryption steps are given below:

- Receives the public key $(\aleph, e_1, e_2)$.
- Displays the plain message as a positive number.
- Determine cipher text $C = (T^{e_1}\ mod\ \aleph)^{e_2}\ mod\ \aleph$.
- Send cipher message.

3.3. Decryption algorithm

Decrypt the cipher text into plain text with the support of a non-secret key in RSA. The decryption steps are given below:

- Compute $C = (T^{d_1}\ mod\ \aleph)^{d_2} (mod\ \aleph)$.
- Extracts a plain text from a number representing T.

3.4. Example

Choose four distinct primes $\wp_1 = 2$, $\wp_2 = 11$, $\wp_3 = 5$, $\wp_4 = 17$.

$$\aleph = \wp_1 \cdot \wp_2 \cdot \wp_3 \cdot \wp_4 = 2 \cdot 11 \cdot 5 \cdot 17 = 1870$$

Euler phi value of $\aleph$ is $\varphi(\aleph) = 640$. Choose $e_1 = 17$ satisfying $1 < e_1 < \varphi(\aleph)$ and $(e_1, \varphi(\aleph)) = 1$. Compute $d_1 = 113$, such that $d_1\, e_1 \equiv 1(\ mod\ \varphi(\aleph))$. Choose $e_2 = 21$ satisfying $1 < e_2 < \varphi(\aleph)$ and $gcd(e_2, \varphi(\aleph)) = 1$ .Calculate $d_2 = 61$ such that $d_2\, e_2 \equiv 1\ (mod\ \varphi(\aleph))$. The public key is $(1870, 17, 21)$, and plain text $= T = 1399$.

$$C = [T^{e_1} (mod\ 1870)]^{e_2} mod\ 1870$$

$$C = [1569]^{e_2}\ mod\ 1870 = 1459$$

The private key is $(1870, 113, 61)$.

$$T = [C^{d_1}\ mod\ 1870]^{d_2}\ mod\ 1870$$

$$T = [1289]^{61}\ mod\ 1870 = 1399.$$

4. RSA cryptographic algorithm using six primes

Now we discuss the RSA algorithm consists of six primes. The private and public key consists four of components. $\aleph$ is a product of prime numbers $\wp_1,\ \wp_2,\ \wp_3,\ \wp_4,\ \wp_5,\ \wp_6$. $(\aleph, e_1, e_2, e_3)$ be the components of the public key, where $e_1,\ e_2$ and e_3 are chosen randomly which are coprime to $\varphi(\aleph)$. Since $\aleph$ is kept as private and public components, with the wisdom of $\aleph$, the attacker cannot obtain the value of the six primes used to determine the value of $\aleph$ and later $e_1,\ e_2,$ and e_3. $(\aleph, d_1, d_2, d_3)$ be the components of the private key, where d_1 is the inverse of e_1 and d_2 is the inverse of e_2, and d_3 is the inverse of e_3. For security purposes, all six selected prime bits are the same length.

4.1. Key generation

The steps for generating the key are given below as;

- Let six primes $\wp_1,\ \wp_2,\ \wp_3,\ \wp_4,\ \wp_5,\ \wp_6,$ and $\aleph$ be the product of $\wp_1,\ \wp_2,\ \wp_3,\ \wp_4,\ \wp_5,\ \wp_6$ as $\aleph = \wp_1.\wp_2.\wp_3.\wp_4.\wp_5.\wp_6$
- Euler phi function of $\aleph$ is $\varphi(\aleph) = \varphi(\wp_1)\cdot\varphi(\wp_2)\cdot\varphi(\wp_3)\cdot\varphi(\wp_4)\cdot\varphi(\wp_5).\varphi(\wp_6)$.
- Choose $e_1,\ e_2,$ and e_3 three numbers coprime to $\varphi(\aleph)$.
- Find $d_1,\ d_2$ and d_3 which is inverses of $e_1,\ e_2$ and e_3 respectively
$$e_1 d_1 \equiv 1\ \left(mod\ \varphi(\aleph)\right),$$
$e_2 d_2 \equiv 1\left(mod\ \varphi(\aleph)\right)$, and $e_3 d_3 \equiv 1\ (mod\ \varphi(\aleph))$
- Public key=$(\aleph, e_1, e_2, e_3)$.
- Private key=$(\aleph, d_1, d_2, d_3)$.

4.2. Encryption algorithm

The decryption process is finished with the use of the public key $(\aleph, e_1, e_2, e_3)$. The encryption steps are given below:

- Receives the public key $(\aleph, e_1, e_2, e_3)$.
- Displays the text T as a positive number.
- Calculate text ciphering $C = [(T^{e_1}\ mod\ \aleph)^{e_2}\ mod\ \aleph]^{e_3}\ mod\ \aleph$.
- Send Cipher message

4.3. Decryption algorithm

The decryption of the cipher text into plain text with the help of a non-public key in RSA. The Decryption steps are given below:

- Compute $T = [(C^{d_1}\ mod\ \aleph)^{d_2}\ mod\ \aleph]^{d_3}\ mod\ \aleph$.
- Extracts plain text from a number representing T.

4.4. Example

Choose six distinct primes $\wp_1 = 7,\ \wp_2 = 13,\ \wp_3 = 11,\ \wp_4 = 2,\ \wp_5 = 3, \wp_6 = 5$.

$$\aleph = \wp_1 \cdot \wp_2 \cdot \wp_3 \cdot \wp_4 \cdot \wp_5 \cdot \wp_6 = 7 \cdot 13 \cdot 11 \cdot 2 \cdot 3 \cdot 5 = 30030.$$

Euler's phi function of $\aleph$ is

$$\varphi(\aleph) = \varphi(\wp_1) \cdot \varphi(\wp_2) \cdot \varphi(\wp_3) \cdot \varphi(\wp_4) \cdot \varphi(\wp_5) \cdot \varphi(\wp_6) = 6 \cdot 12 \cdot 10 \cdot 1 \cdot 2 \cdot 4 = 5760$$

Choose $e_1 = 59$ satisfying $1 < e_1 < \varphi(\aleph)$ and $gcd(e_1, \varphi(\aleph)) = 1$. Let $d_1 = 4979$ as $d_1\, e_1 \equiv 1\ (mod\ \varphi(\aleph))$.
Choose $e_2 = 13$ satisfying $1 < e_2 < \varphi(\aleph)$ and $gcd(e_2, \varphi(\aleph)) = 1$. Let $d_2 = 5317$ as $d_2\, e_2 \equiv 1\ (mod\ \ \varphi(\aleph))$.
Choose $e_3 = 7$ satisfying $1 < e_3 < \varphi(\aleph)$, and $gcd(e_3, \varphi(\aleph)) = 1$. Let $d_3 = 823$ as $d_3 e_3 \equiv 1\ (mod\ \ \varphi(\aleph))$.
Public key is $(30030, 59, 13, 7)$. Plain text $= T = 1321$.

$$C = [((T^{e_1}\ mod\ \aleph)^{e_2}\ mod\ \aleph)]^{e_3}\ mod\ \aleph$$

$$C = [((132159\ mod\ 30030)^{13}\ mod\ 30030)]^7\ mod\ 30030$$

$$C = [(1941)^{13}\ mod\ 30030)]^7\ mod\ 30030$$

$$C = [19141]^7\ (mod\ 30030) = 9901$$

The private key is $(30030, 4979, 5317, 823)$.

$$T = [((C^{d_1}\ mod\ \aleph)^{d_2}\ mod\ \aleph)]^{d_3}\ mod\ \aleph$$

$$T = [(9901^{4979}\ mod\ 30030)^{5317}\ mod\ 30030]^{823}\ (mod\ 30030)$$

$$T = [10561^{5317}\ mod\ 30030]^{823}\ (mod\ 30030)$$

$$T = 10561^{823}\ (mod\ 30030) = 1321.$$

5. RSA cryptographic algorithm using eight primes

We talk about the RSA algorithm, which uses eight big prime numbers. The private and public key contains five components. $\aleph$ is the product of primes $\wp_1, \wp_2, \wp_3, \wp_4, \wp_5, \wp_6, \wp_7, \wp_8$. $(\aleph, e_1, e_2, e_3, e_4)$ be the components of the public key, where e_1, e_2, e_3, and *are chosen* randomly which are coprime to $\varphi(\aleph)$. Since $\aleph$ is kept as private and public components, the attacker is unable to evaluate the value of the eight basic primes that are used to obtain the value of $\aleph$ without knowing the value of $\aleph$, and later e_1, e_2, e_3, and e_4. $(\aleph, d_1, d_2, d_3, d_4)$ be the components of private key, where d_1 is the inverse of e_1 and d_2 is the inverse of e_2 and d_3 is the inverse of e_3 and d_4 is the inverse of e_4. For security purposes, all eight selected prime bits are the same length.

5.1. Key generation

The steps for generating the key are given below:

- Choose eight primes $\wp_1, \wp_2, \wp_3, \wp_4, \wp_5, \wp_6, \wp_7, \wp_8$ *and*

$$\aleph = \wp_1 \cdot \wp_2 \cdot \wp_3 \cdot \wp_4 \cdot \wp_5 \cdot \wp_6 \cdot \wp_7 \cdot \wp_8.$$

- Euler phi function of $\aleph$ is

$$\varphi(\aleph) = \varphi(\wp_1)\cdot\varphi(\wp_2)\cdot\varphi(\wp_3)\cdot\varphi(\wp_4)\cdot\varphi(\wp_5)\cdot\varphi(\wp_6)\cdot\varphi(\wp_7)\cdot\varphi(\wp_8)$$

- Choose e_1, e_2, e_3 and e_4 four numbers coprime to $\varphi(\aleph)$.
- Find d_1, d_2, d_3 and d_4 which are the inverses of e_1, e_2, e_3 and e_4 respectively $e_1 d_1 \equiv 1\left(mod\ \varphi(\aleph)\right)$, $e_2 d_2 \equiv 1\left(mod\ \varphi(\aleph)\right)$, $e_3 d_3 \equiv 1\left(mod\ \varphi(\aleph)\right)$, $e_4 d_4 \equiv 1\ (mod\ \varphi(\aleph))$.
- $(\aleph, e_1, e_2, e_3, e_4)$ is the public key.
- $(\aleph, d_1, d_2, d_3, d_4)$ is private key.

5.2. Encryption algorithm

Decrypt the cipher text into plain text with the help of a non-public key in RSA. The decryption steps are given below:

- Receives the public key $(\aleph, e_1, e_2, e_3, e_4)$.
- Displays the message T as a positive number.
- Compute cipher text $C = [((T^{e_1}\ mod\ \aleph)^{e_2}\ mod\ \aleph)^{e_3}\ mod\ \aleph]^{e_4}\ mod\ \aleph$.
- Send cipher message.

5.3. Decryption algorithm

The decryption of the cipher text into plain text with the help of a non-public key in RSA. The Decryption steps are given below:

- Compute $T = [((C^{d_1}\ mod\ \aleph)^{d_2}\ mod\ \aleph)^{d_3}\ mod\ \aleph]^{d_4}\ mod\ \aleph$
- Extracts plain text from a number representing T.

5.4. Example

Let eight distinct primes $\wp_1 = 2$, $\wp_2 = 5$, $\wp_3 = 3$, $\wp_4 = 7$, $\wp_5 = 11$, $\wp_6 = 13$, $\wp_7 = 17$, $\wp_8 = 19$ and $\aleph = \wp_1\cdot\wp_2\cdot\wp_3\cdot\wp_4\cdot\wp_5\cdot\wp_6\cdot\wp_7\cdot\wp_8 = 2\cdot3\cdot5\cdot7\cdot11\cdot13\cdot17\cdot19 = 9699690$. Compute the Euler phi value of $\aleph$.

$$\begin{aligned}\varphi(\aleph) &= \varphi(\wp_1)\cdot\varphi(\wp_2)\cdot\varphi(\wp_3)\cdot\varphi(\wp_4)\cdot\varphi(\wp_5)\cdot\varphi(\wp_6)\cdot\varphi(\wp_7)\cdot\varphi(\wp_8)\\ &= 1\cdot2\cdot4\cdot6\cdot10\cdot12\cdot16\cdot18 = 1658880.\end{aligned}$$

Choose $e_1 = 1001$ satisfying $1 < e_1 < \varphi(\aleph)$ and $gcd(e_1, \varphi(\aleph)) = 1$. Find $d_1 = 1324221$ such that $d_1\, e_1 \equiv 1\ (mod\ \varphi(\aleph))$.
Choose $e_2 = 2003$ satisfying $1 < e_2 < \varphi(\aleph)$, and $gcd(e_2, \varphi(\aleph)) = 1$. Find $d_2 = 649307$ such that $d_2 e_2 \equiv 1\left(mod\ \varphi(\aleph)\right)$.
Choose $e_3 = 50003$ satisfying $1 < e_3 < \varphi(\aleph)$ and $\gcd\left(e_3, \varphi(\aleph)\right) = 1$. Fin $d_3 = 555227$ such that $d_3 e_3 \equiv 1\left(mod\ \varphi(\aleph)\right)$.
Choose $e_4 = 500011$ satisfying $1 < e_4 < \varphi(\aleph)$ and $gcd(e_4, \varphi(\aleph)) = 1$. Find $d_4 = 247171$ such that $d_4 e_4 \equiv 1\left(mod\ \varphi(\aleph)\right)$.
The public key is $(9699690, 1001, 2003, 50003, 500011)$. Plain text= $T = 1321$.

$$C = [((T^{e_1}\ mod\ \aleph)^{e_2}\ mod\ \aleph)^{e_3}\ mod\ \aleph]^{e_4}(mod\ \aleph).$$

$$= \left[\left(\left(1321^{1001}(mod\ 9699690)\right)^{2003}(mod\ 9699690)\right)^{50003}(mod\ 9699690)\right]^{500011}(mod\ 9699690).$$

$$= [(4724611^{2003}(mod\ 9699690))^{50003}(mod\ 9699690)]^{500011}(mod\ 9699690).$$
$$= [1932281^{50003}\ (mod\ 9699690)]^{500011}\ (mod\ 9699690).$$
$$= 6856741^{500011}\ (mod\ 9699690) = 2661781$$

The private key is $(9699690, 1324121, 649307, 555227, 247171)$

$$T = [((C^{d_1}\ mod\ \aleph)^{d_2}\ mod\ \aleph)^{d_3}\ mod\ \aleph]^{d_4}\ mod\ \aleph$$
$$= [((2661781^{1324121} mod\ 9699690)^{649307} mod\ 9699690))^{555227}\ mod\ 9699690]^{247171}\ mod\ 9699690.$$
$$T = [(8779321^{649307}\ mod\ 9699690)^{555227} mod\ 9699690]^{247171}(mod\ 9699690)$$
$$T = [730621^{555227}\ (mod\ 9699690)]^{247171}\ (mod\ 9699690).$$
$$T = 9079621^{247171}\ (mod\ 9699690) = 1321.$$

Continuously this process for up to 2n primes in the following section.

6. GRSA algorithm with $2n$ primes

Now we'll look at the RSA Algorithm, which is made up of $2n$ large primes. The private and public keys are made up of $n+1$ components. $\aleph$ is a product of primes $\wp_1$, $\wp_2$, $\wp_3$, $\wp_4$, $\wp_5$, $\wp_6$, $\cdots$, $\wp_{2n-1}, \wp_{2n}$. $(\aleph, e_1, e_2, e_3, \cdots, e_n)$ are the components of public key, where $e_1, e_2, e_3, \ldots e_n$ are choose randomly which are coprime to $\varphi(\aleph)$. If an attacker has access to the $\aleph$ key, he or she will be unable to deduce the value of the $2n$ fundamental primes, which serve as the foundation for calculating $\aleph$, and later $e_1, e_2, e_3, \cdots, e_n$. $(\aleph, d_1, d_2, d_3, \cdots, d_n)$ be the components of private key, where d_1 is the inverse of e_1 and d_2 is the inverse of e_2 and d_3 is the inverse of d_3 similarly d_n is the inverse of e_n. For security purposes, all $2n$ selected prime bits are of the same length.

6.1. Key generation

There are the following steps for generating the key;

- Let $\wp_1, \wp_2, \wp_3, \wp_4, \wp_5, \wp_6, \cdots, \wp_{2n-1}, \wp_{2n}$ be the $2n$ distinct primes and $\aleph$ is a product of $\wp_1, \wp_2, \wp_3, \wp_4, \wp_5, \wp_6, \cdots, \wp_{2n-1}, \wp_{2n}$ as $\aleph = \wp_1 . \wp_2 . \wp_3 . \wp_4 . \wp_5 . \wp_6 \ldots \wp_{2n-1} . \wp_{2n}$.
- Find the Euler phi function of $\aleph$ as
$$\varphi(\aleph) = \varphi(\wp_1).\varphi(\wp_2).\varphi(\wp_3).\varphi(\wp_4)\cdot\varphi(\wp_5).\varphi(\wp_6)\ldots,\varphi(\wp_{2n-1})\cdot\varphi(\wp_{2n}).$$
- Let $e_1, e_2, e_3, \ldots, e_n$ be the n non-negative numbers co-prime to $\varphi(\aleph)$.
- Let $d_1, d_2, d_3, \ldots, d_n$ be the inverses of $e_1, e_2, e_3, \ldots, e_n$ such that $e_1 d_1 \equiv 1\ mod\ \varphi(\aleph)$, $e_2 d_2 \equiv 1\ mod\ \varphi(\aleph)$, $e_3 d_3 \equiv 1\ mod\ \varphi(\aleph), \cdots, e_n d_n \equiv 1\ mod\ \varphi(\aleph)$.
- Public key= $(\aleph, e_1, e_2, e_3, \ldots, e_n)$.
- Private key= $(\aleph, d_1, d_2, d_3, \ldots, d_n)$.

6.2. Encryption algorithm

Its use of public keys $(\aleph, e_1, e_2, e_3, \ldots, e_n)$ puts an end to the encryption process. The encryption steps are given below;

- Receives the public key $(\aleph, e_1, e_2, e_3, \ldots, e_n)$.

- Displays text T as a positive number.
- Find Cipher text $C = \left[\left(\left(T^{e_1}(mod\ \aleph)\right)^{e_2}(mod\ \aleph)\right)^{e_3}(mod\ \aleph)\cdots\right]^{e_n}(mod\ \aleph)$.
- Send a Cipher message.

6.3. Decryption algorithm

The decryption of the cipher text into plain text with the private key in GRSA. The decryption steps are given below;

- Compute $T = \left[\left(\left(C^{d_1}(mod\ \aleph)\right)^{d_2}(mod\ \aleph)\right)^{d_3}(mod\ \aleph)\cdots\right]^{d_n}(mod\ \aleph)$.

Extracts plain text from a number representing T.

7. Comparison

The proposed GRSA algorithm is executed using MATLAB manifesto using Laptop (Del, Core i7, 7th generation). Comparison of the key generation time, encryption time, and decryption time for two primes, four primes, six primes, and eight primes. During the counterfeit, 5 different combinations of arbitrary primes are chosen. There are the following results. The key generation (K.G), encryption (E), and decryption (D) time for producing public and private keys by two primes, four primes', six primes, and eight primes with magnitudes are given in Tables 1–4.

Table 1. Key generation, encryption, and decryption time for two primes.

$\wp_1$	$\wp_2$	E	K.G. (sec)	E.(sec)	D.(sec)
19	23	7	0.005716	0.376943	0.005716
17	29	3	0.004420	0.375985	0.007462
13	31	7	0.006389	0.379625	0.008168
31	11	3	0.007449	0.378423	0.007617
11	29	3	0.007284	0.376564	0.008133

Table 2. Key generation, encryption, and decryption time for four primes.

$\wp_1$	$\wp_2$	$\wp_3$	$\wp_4$	e_1	e_2	K.G.(sec)	E.(sec)	D.(sec)
19	17	29	3	5	11	0.008708	0.386488	0.012452
13	29	23	5	5	13	0.008452	0.398772	0.012664
7	23	31	11	7	13	0.008019	0.382741	0.014080
11	29	23	7	13	17	0.008071	0.380112	0.013811
13	23	19	11	13	17	0.008975	0.379947	0.015377

Table 3. Key generation, encryption, and decryption time for six primes.

$\wp_1$	$\wp_2$	$\wp_3$	$\wp_4$	$\wp_5$	$\wp_6$	e_1	e_2	e_3	K.G.(sec)	E.(sec)	D.(sec)
3	13	37	7	23	11	7	13	19	0.009489	0.393181	0.653823
37	13	3	5	29	7	5	11	13	0.012368	0.385374	0.347764
3	17	13	31	23	7	7	13	17	0.010318	0.384243	0.788194
5	31	19	13	2	7	7	11	13	0.013297	0.380655	0.93392
5	13	41	7	23	11	7	13	19	0.012737	0.383298	0.843879

Table 4. Key generation, encryption, and decryption time for eight primes.

$\wp_1$	$\wp_2$	$\wp_3$	$\wp_4$	$\wp_5$	$\wp_6$	$\wp_7$	$\wp_8$	e_1	e_2	e_3	e_4	K.G.(sec)	E.(sec)	D.(sec)
13	3	17	7	31	19	2	23	7	13	17	23	0.017519	0.392799	42.830815
31	2	7	17	23	29	3	11	13	17	19	23	0.012287	0.390375	37.090092
3	7	13	11	19	23	29	5	13	17	19	23	0.010551	0.390682	40.103547
37	2	13	7	3	29	23	11	13	17	19	23	0.013907	0.390470	32.524731
17	7	13	11	19	31	3	5	11	13	19	23	0.015120	0.392092	62.955846

Comparison of the key generation, encryption, and decryption time of two primes, four primes, six primes, and eight primes with magnitudes are given in Figures 1–6.

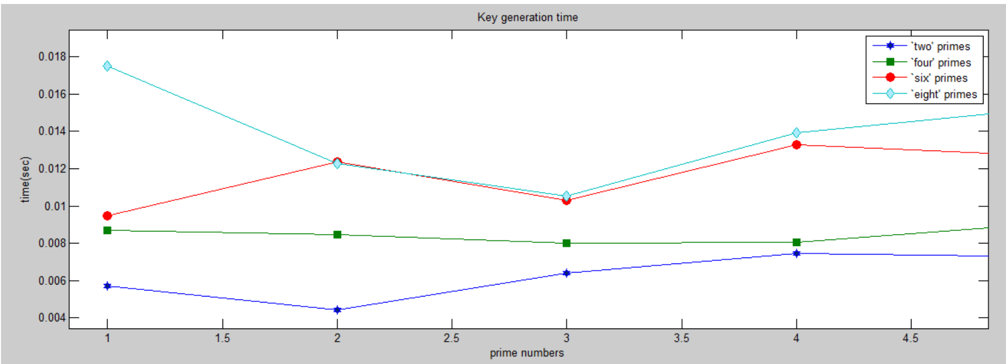

Figure 1. Key generation time vs different primes.

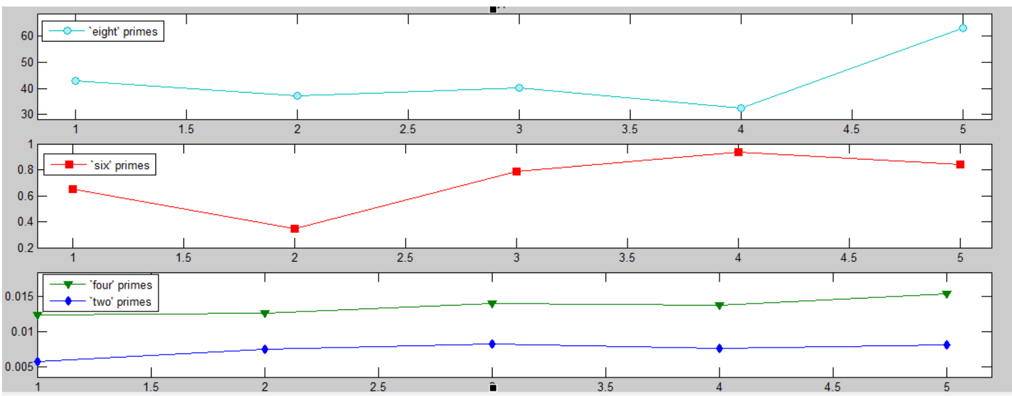

Figure 2. Decryption time vs different primes.

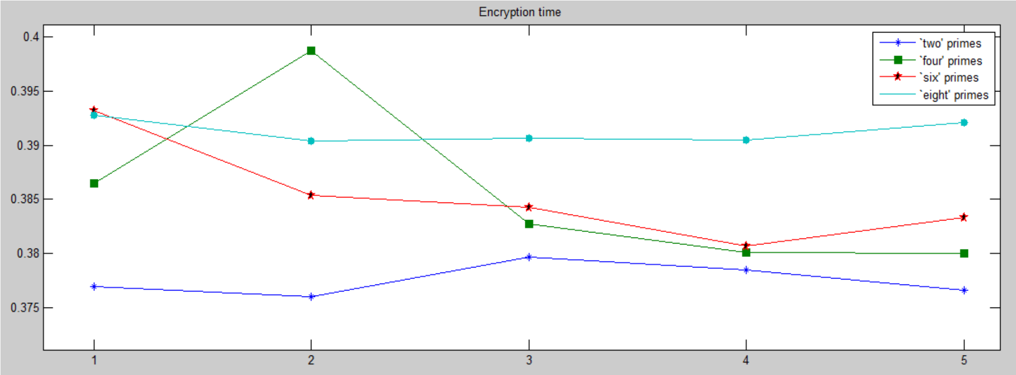

Figure 3. Encryption time vs different primes.

Comparison of the key generation, encryption and decryption time of two primes, four primes, six primes, and eight primes with the help of bar the given graphs.

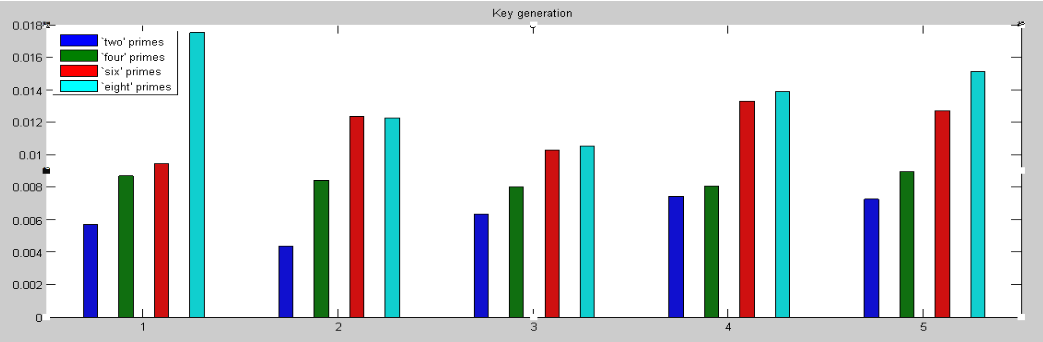

Figure 4. Key generation for different primes.

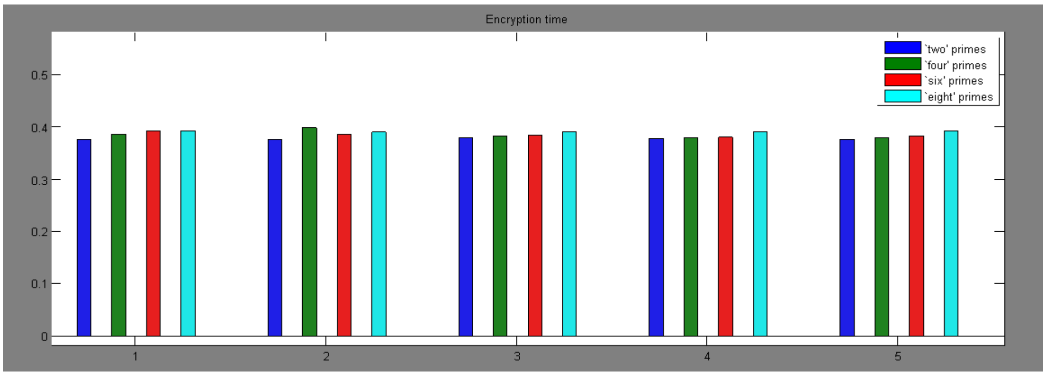

Figure 5. Encryption time for different primes.

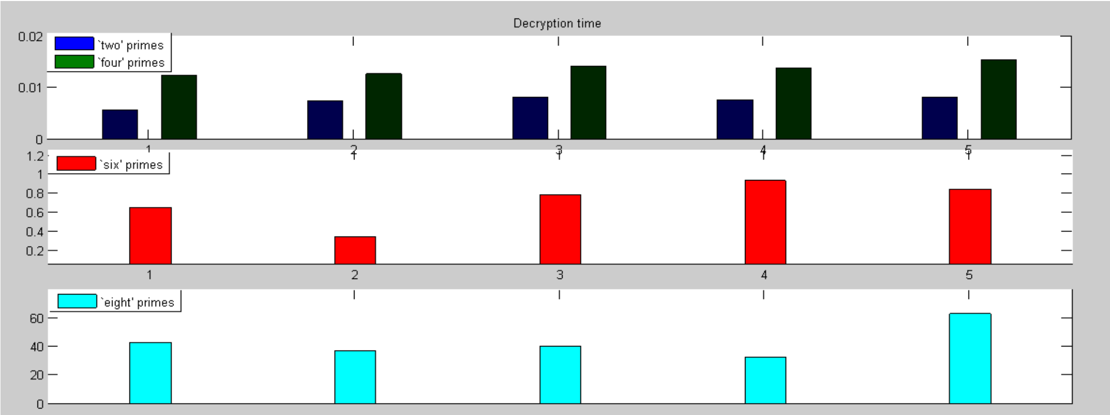

Figure 6. Decryption time for different primes.

8. Conclusions

RSA and GRSA have different important parameters that affect their level of security and speed. Increasing the length of the modulus gives rise to the complexity of decomposing it into factors. Thus, the length of the private key increased and the key is harder to trace. RSA and GRSA parameter changes are time-dependent and constant to study other relevant stresses. We concluded that the key generation time of GRSA is higher than that of RSA due to an increase in the number of primes. The higher key generation time of GRSA can be seen as an advantage of the fact that the time to crack the system is longer due to the added complexity. Encryption time shows the same amount of time used by RSA and GRSA for the lower bit lengths of primes (two and four primes). As the number of primes and magnitude of primes increase, then the time will be increased. Decryption time shows the same amount of time used by RSA and GRSA for the lower bit lengths of primes (two and four primes). As the number of primes increased, the difference between the curves becomes steeper. From the above discussion, we concluded that the encryption and decryption time of GRSA is higher than the RSA. The increase in time would be acceptable if it substantially would increase the security of the proposed GRSA method.

Acknowledgments

The authors present their appreciation to King Saud University for funding this research through Researchers Supporting Project number: RSPD2023R650, King Saud University, Riyadh, Saudi Arabia.

Conflict of interest

The authors declared that they had no conflict of interest.

Use of AI tools declaration

The authors declare they have not used Artificial Intelligence (AI) tools in the creation of this article.

References

1. H. Ukwuoma, M. Hammawa, Optimised Key Generation for RSA Encryption, *IISTE*, **6** (2015), 2222–2871.
2. M. Preetha, M. A. Nithya, Study and performance analysis of RSA algorithm, *IJCSMC*, **2** (2013), 126–139.
3. C. Paar, J. Pelzl, Introduction to cryptography and data security, In*: Understanding Cryptography Springer*, 2010, 1–27. https://doi.org/10.1007/978-3-642-04101-3_1
4. N. Koblitz, A. J. Menezes, Y. H. Wu, R. J. Zuccherato, *Algebraic aspects of cryptography, Springer*, **198** (1998), 1–17. https://doi.org/10.1007/978-3-662-03642-6_1
5. Y. Li, Q. Liu, T. Li, Design and implementation of an improved RSA algorithm, In:*2010 International Conference on E-Health Networking Digital Ecosystems and Technologies (EDT), IEEE*, **1** (2010), 390–393.
6. M. Sajjad, T. Shah, J. R. Serna, Nonlinear components of a block cipher over Gaussian integers, *CMC-Comput. Mater. Con.*, **75** (2022), 5287–5305. https://doi.org/10.32604/cmc.2023.035347
7. M. Sajjad, T. Shah, M. M. Hazzazi, A. R. Alharbi, I. Hussain, Quaternion integers based higher length cyclic codes and their decoding algorithm, *CMC-Comput. Mater. Con.*, **73** (2022), 1177–1194. https://doi.org/10.32604/cmc.2022.025245
8. M. Sajjad, T. Shah, J. R. Serna, A. Z. E. Suarez, O. S. Delgado, Fundamental results of cyclic codes over octonion integers and their decoding algorithm, *Computation*, **10** (2022), 1–12. https://doi.org/10.3390/computation10120219
9. T. Takagi, Fast RSA-type cryptosystem modulo p k q, In: *Annual International Cryptology Conference, Springer, Berlin, Heidelberg*, 1998, 318–326. https://doi.org/10.1007/BFb0055738
10. D. Boneh, Twenty years of attacks on the RSA cryptosystem, *Notices AMS*, **46** (1999), 203–213.
11. B. P. U. Ivy, P. Mandiwa, M. Kumar, A modified RSA cryptosystem based on ‘n’prime numbers, *Int. J. Eng. Compu. Sci.*, **1** (2012), 63–66.
12. J. S. Yadav, A. S. Sheregar, V. M. Panjri, S. L. Gharat, Secure approach for encrypting data, In: *2018 International Conference on Smart City and Emerging Technology (ICSCET)*. *IEEE*, 2018; 1–3. https://doi.org/10.1109/ICSCET.2018.8537290
13. P. K. Panda, S. Chattopadhyay, A hybrid security algorithm for RSA cryptosystem, In: *2017 4th International Conference on Advanced Computing and Communication Systems (ICACCS), IEEE,* 2017, 1–6. https://doi.org/10.1109/ICACCS.2017.8014644
14. K. Bhatele, A. Sinhal, M. Pathak, A novel approach to the design of a new hybrid security protocol architecture, In: *2012 IEEE International Conference on Advanced Communication Control and Computing Technologies (ICACCCT), IEEE,* 2012, 429–433. https://doi.org/10.1109/ICACCCT.2012.6320816
15. M. A. Islam, N. Islam, B. Shabnam, A modified and secured RSA public key cryptosystem based on “n” prime numbers, *J. Comput. Comm.*, **6** (2018), 1–13. https://doi.org/10.4236/jcc.2018.63006