# A NOVEL HYBRID SECURITY ALGORITHM

By

MANISHA KUMARI * DEEKSHA EKKA ** NISHI YADAV ***

*-** UG Scholar, Department of Computer Science and Engineering, Guru Ghasidas University, Bilaspur, Chhattisgarh, India.
*** Assistant Professor, Department of Computer Science and Engineering, Guru Ghasidas University, Bilaspur, Chhattisgarh, India.

Date Received: 04/01/2018 Date Revised: 13/03/2018 Date Accepted: 30/04/2018

*ABSTRACT*

*A New Novel Hybrid Security Algorithm (NHSA) for Rivest-Shamir-Adleman (RSA) cryptosystem was proposed in this paper, which is based on Encryption algorithm using Dual Modulus and Enhanced method for RSA (ERSA) cryptosystem. Here, the computation of public keys and private keys depends on modulus values, each computed using three different prime integers. Thus complexity involved in factorizing the modulus value increases. It improves the security of RSA scheme against Brute Force Attack using double mod operation based encryption and decryption. Therefore, it is not possible to retrieve the original message for the cipher text even after determining a single public key. Also it is difficult to factorize the modulus value into its three prime factors. Thus it enhances the security of encrypted data two times. In this paper, the proposed algorithm is compared with "Encryption algorithm using dual modulus" in terms of key generation time and security of data.*

*Keywords: New Novel Hybrid Security Algorithm (NHSA), RSA Cryptosystem, Dual Modulus, Prime Number, Encryption, Decryption.*

## INTRODUCTION

Security can be referred as a degree of protection to harm the data and resistance to harm the data. To ensure data security, confidentiality, integrity and availability of data are important (Aboud et al., 2008; Suja and Jose, 2016). Confidentiality ensures that given information can only be accessed by an authorized person. Integrity specifies the originality of data, and ensures that data is not being modified. Availability is defined as the assurance that user has access to information anytime and anywhere in the network (Ramaporkalai, 2017).

### 1. Literature Review

Goel (2017) has mentioned that the communication over the internet is increasing day by day, and security of data on wireless network has become vital. Nowadays it is common to exchange personal data on internet. So, data security is crucial. In order to communicate over internet, the sender has to encrypt the message or plain text with receiver's public key, and then receiver has to decrypt the encrypted text or cipher text using decryption key.

Ramaporkalai (2017) has concluded that Cryptography for data security is a very powerful method for protecting data, from being stolen. Cryptography is a method to encode the information, to keep the information being hacked by the third party. Jeeva et al. (2012) concluded that the most popular method of encryption is symmetric key encryption. In this method, the same key is used for both encryption and decryption process. Symmetric key encryption takes place either in block cipher or in stream cipher. As the same key is being used for both the process of encryption and decryption, the computational power of this encryption technique is small. While in asymmetric key encryption technique, different keys are used for encryption and decryption process. It is also known as public key encryption. This encryption technique is slow and impractical in case of large amount of data.

Pancholi and Patel (2016) mentioned that the hash function involves a mathematical function to irreversibly "encrypt" the data. It consist of algorithms like Message

digest and hash function algorithm. Madaan and Agrawal (2012) and Osseily et al. (2008) concluded that Cloud computing is widely accepted around the world. But security of data on cloud server is a challenging issue. The best way to secure the information on cloud server is by using a security algorithm.

## 2. Background Details

### *2.1 Rivest, Shamir and Adleman (RSA) Algorithm*

Verma and Garg (2014), Dongjiang et al. (2012) and Bhandari et al. (2016) have described that in conventional RSA scheme, two large prime integers say "p" and "q" are used for the computation of variable n. The security of RSA algorithm relies on the practical difficulty of factorization of product of two large integers. Ambedkar et al. (2011) have described the RSA scheme as follows.

*2.1.1 Key Generation*

- Select two large random prime numbers, p and q of approximately equal size that their product $n = p*q$ is desired bit length.
- Compute $n = p * q$ and $Ø(n) = (p-1) * (q-1)$.
- Choose a positive integer e, such that $1 < e < Ø(n)$, such that GCD $(e, Ø(n)) = 1$.
- Compute the value of secret Exponent d, $1 < d < Ø(n)$, such that $e * d = 1 (\text{mod } Ø(n))$.
- The pair (e, n) is public key and (d, n) is private key. The values d, p, q and Ø(n) should be kept as a secret.

where,

n is a modulus.

e is the public exponent or encryption exponent or simply the exponent.

d is the secret exponent or decryption exponent.

*2.1.2 Encryption*

Suppose User A wants to send a message "m" to User B

- Obtain the public key (e, n) of user B.
- Represent the plaintext as positive integer m.
- Compute the cipher text $c = m^e \text{ mod } n$, using user B's public key.
- Send the cipher text c to user B.

*2.1.3 Decryption*

User B will retrieve the original message from cipher text sent by the user A.

- Use private key (d, n) to compute $m = c^d \text{ mod } n$.
- Extract the plaintext m from c.

### *2.2 Encryption Algorithm using Dual Modulus*

Panda and Chattopadhyay (2017) have used the dual modulus operation. In this algorithm, encryption as well as decryption processes are employed using double modulus operation using two private keys "e1" and "e2" and public keys "d1" and "d2" respectively. More than two large prime numbers say "p1", "p2", "q1" and "q2" are used for generation of modulus values say "n1" and "n2". Dual modulus encryption technique seems to be impractical, as it takes large amount of computational time for the generation of public key and private key (Devaota et al., 2015). But it also enhances the security two times. Dual modulus algorithm is as follows:

*2.2.1 Key Generation*

- Select four large random primes p1, p2 and q1, q2 of approximately equal size.
- Compute $n1 = p1 * q1$ and $n2 = q1 * q2$.
- Compute Euler's totient $Ø(n1) = (p1-1) * (p2-1)$ and $Ø(n2) = (q1-1) * (q2-1)$.
- Choose two positive integers e1, e2, such that $1 < e1 < Ø(n1)$, GCD $(e1, Ø(n1)) = 1$ and $1 < e2 < Ø(n2)$, GCD $(e2, Ø(n2)) = 1$.
- Compute the secret Exponent d1, d2, such that $1 < d1 < Ø(n1)$, $e1 * d1 = 1 \ (\text{mod } Ø(n1))$ and $1 < d2 < Ø(n2)$, $e2 * d2 = 1 \ (\text{mod } Ø(n2))$.
- The public key is (e1, e2, n1, n2) and the private key is (d1, d2, n1, n2). Keep all the values d1, d2, p1, p2, q1, q2, Ø(n1) and Ø(n2) secret.

where,

n1 and n2 are known as modulus.

e1 and e2 are known as the public exponent or encryption exponent, or simply the exponent.

d1 and d2 are known as the secret exponent or decryption exponent.

*2.2.2 Encryption*

Suppose, user A wants to send message to user B.

- Obtain the public key (e1, e2, n1, n2).
- Represent the plaintext as positive integer m.
- Compute the cipher text $c = ((m^{e1} \bmod n1)^{e2} \bmod n2)$.
- Send the cipher text c to user B.

*2.2.3 Decryption*

Now, user B will retrieve the original message.

- Use private key (d1, d2, n1, n2) to compute $m = ((c^{d2} \bmod n2)^{d1} \bmod n1)$.
- Extract the plaintext m from c.

***2.3 Enhanced Method for RSA Cryptosystem Algorithm***

Al-Hamami and Aldariseh (2012) changed the method for generation of public and private key. In this algorithm, three large prime numbers say "p", "q" and "r" instead of two prime numbers are used to generate public and private key. Here in ERSA, it is more difficult to factorize the modulus "n" which is a product of three different prime integers. Thus it enhances the security of conventional RSA and the steps involved in this algorithm are as follows.

*2.3.1 Key Generation*

- Generate three large random primes, p, q and r of approximately equal size that their product $n = p*q*r$ is of desired bit length.
- Compute $n = p * q * r$ and $\emptyset(n) = (p - 1) * (q - 1) * (r - 1)$.
- Choose a positive integer e, such that $1 < e < \emptyset(n)$, such that GCD $(e, \emptyset(n)) = 1$.
- Compute the secret Exponent d, $1 < d < \emptyset(n)$, such that $e * d = 1 \pmod{\emptyset(n)}$.
- The public key is (e, n) and the private key is (d, n). Keep all the values d, p, q and $\emptyset$ secret.

where,

n is known as modulus.

e is known as the public exponent or encryption exponent, or simply the exponent.

d is known as the secret exponent or decryption exponent.

*2.3.2 Encryption*

User A wants to send message "m" to user B.

- Obtain the public key (e, n).
- Represent the plaintext as positive integer m.
- Compute the cipher text $c = m^e \bmod n$.
- Send the cipher text c to user B.

*2.3.3 Decryption*

User B will retrieve the message from the cipher text.

- Use private key (d, n) to compute $m = c^d \bmod n$.
- Extract the plaintext m from c.

**3. Proposed Method**

The basic idea of the proposed approach is based on Encryption algorithm using dual modulus and Enhanced method for RSA cryptosystem algorithm.

Using dual modulus in proposed algorithm, a double mod operation based encryption and decryption was introduced using two public keys say "e1" and "e2", and two private keys say "d1" and "d2", respectively. Using dual modulus operation in proposed algorithm improves the security of data as compared with RSA cryptosystem to very large extent. In proposed approach, even if an intruder gets succeeded in detecting a private key, still our data will be secured as it is encrypted with two different public keys. Thus the proposed algorithm is more secured than conventional RSA cryptosystem.

Enhanced method for RSA cryptosystem algorithm used in the proposed algorithm provides an idea of using three prime numbers for the calculation of each modulus value such as "p1", "p2" and "p3" are used for the generation of "n1" and "q1", "q2" and "q3" are used for the generation of "n2" as described by Panda and Chattopadhyay (2017). It is more difficult to factorize the modulus value into its three composite prime factors. It might increase the complexity in computation but it also enhances the security of data in the proposed approach. Here, in the proposed scheme, totally six prime numbers have been used to generate the public key and private key.

Features of proposed algorithm for RSA cryptosystem are as follows:

- Three prime numbers are used to generate modulus,

say n1 and n2.

- Double mod operation based encryption using two private keys.
- Double mod operation based decryption using two public keys.

### *3.1 Steps Involved in Proposed Scheme*

*3.1.1 Key Generation*

- Generate large, random primes p1, p2, p3, q1, q2, q3.
- Compute n1 = p1 * p2 * p3 .
  n2 = q1 * q2 * q3.
- Compute Euler's totient function
  Ø(n1) = (p1-1) * (p2-1) * (p3-1) .
  Ø(n2) = (q1-1) * (q2-1) * (q3-1) .
- Generate two integers e1 and e2 such that 1 < e1 < Ø(n1), GCD(e1, Ø(n1)) = 1 and 1< e2 < Ø(n2), GCD(e2, Ø(n2)) = 1.
- Compute the private key d1 and d2 such that e1*d1 mod Ø(n1) = 1 and e2*d2 mod Ø(n2) = 1.
- (e1, e2, n1, n2) is public key and (d1, d2, n1, n2) is private key.

where,

n1 and n2 are modulus.

e1 and e2 are public exponent or encryption exponent.

d1 and d2 are private exponent or decryption exponent.

*3.1.2 Encryption*

User A wants to send message "m" to user B.

- Obtain the private key of user B i.e, (e1,n1) and (e2,n2)
- Represent the message as positive integer m.
- Encrypt the plaintext m by c = $((m^{e1} \bmod n1)^{e2} \bmod n2)$.
- Send the cipher text c to user B.

*3.1.3 Decryption*

User B will retrieve the original message as,

- Decrypt the cipher text c by m = $((c^{d2} \bmod n2)^{d1} \bmod n1)$.
- Extract the plaintext m from the cipher c.

The public keys "e1" and "e2", and private keys "d1" and "d2" depend on "n1" and "n2", respectively. In this scheme, plain text "m" is converted into final cipher text "c2" by applying encryption process twice and the original message "m" can be retrieved by the receiver by applying decryption process twice.

### *3.2 Proof*

Cipher text generated by sender using message "m" is initially encrypted using "e1" to generate intermediate cipher text "c1", and then "c1" is encrypted again using "e2" to generate final cipher text c2. Encryption process is done twice as $c1 = m^{e1} \bmod n$ and $c2 = c1^{e2} \bmod n$.

Same process is followed for decryption, "c2" is decrypted into "c1" using "d2", and "c1" is decrypted using "d1" to retrieve the original message "m". Decryption process is done twice as $c1 = c2^{d2} \bmod n$ and $m = c1^{d1} \bmod n$.

### *3.3 Example*

- Select prime numbers p1=23, p2=29, p3=31, q1=37, q2=41 and q3=43.
- Calculate modulus value, n1 = p1 * p2 * p3 = 20677 and n2 = q1 * q2 *q3 = 65231.
- Calculate totient function phi, Φ(n1) = (p1-1) * (p2-1) * (p3-1)= 18480 and Φ(n2) = (q1-1) * (q2-1) * (q3-1)= 60480.
- Choose public exponent e1 and e2 such that 1<e1< Φ(n1) and gcd (a1, Φ(n1))=1 and 1<e2< Φ(n2) and gcd (e2, Φ(n2))=1. Then, e1= 18461 and e2= 60457.
- Compute private exponent d1 and d2 such that e1 x d1 mod (Φ(n1)) =1 and e2 x d2 mod (Φ(n2))=1. Then, d1= 7781and d2= 42073.
- Public key : (e1,n1) = (18461, 20677) and (e2, n2) = (60457, 65231). Private Key : (d1,n1) = (7781, 20677) and (d2,n2) = (42073, 65231).

Say, A want to encrypt the message m = 20

*Encryption*

$c = ((m)^{e1} \bmod n1)^{e2} \bmod n2)$

$= ((21)^{18461} \bmod 20677)^{60457} \bmod 65231) = ?ô$

Say, B wants to decrypt the cipher text c, then

*Decryption*

$m = ((c)^{d2} \bmod n2)^{d1} \bmod n1)$

$= ((?\hat{o})^{42073} \bmod 65231)^{7781} \bmod 20677) = 20$

## 4. Results and Discussion

The proposed algorithm is being compared with RSA using dual modulus on the basis of the following parameters such as

- Key generation time
- Security of data

Performance of NHSA by taking various combinations of prime numbers for desired key length is given in Table 1. Ten different combinations of prime numbers of bit length 512 bits are used to generate the key of 2048 bits.

### *4.1 Key Generation Time*

The proposed algorithm takes less time than RSA using dual modulus in key generation. Using both dual modulus operation and eight prime numbers might increase complexity, but it also enhances the security. Comparison of key generation time in proposed algorithm and RSA with dual modulus for various combinations of prime numbers is shown in Figure 1.

The speed up (%) of NHSA and Dual Modulus RSA for Key generation is given in Table 2.

### *4.2 Security of Data*

Proposed scheme is more secured as compared to conventional RSA cryptosystem, ERSA and RSA with dual modulus. Proposed scheme is using double encryption and decryption with two different public keys and two different private keys along with eight prime integers, which has been used instead of four or six prime integers, for the computation of modulus value n1 and n2. Hence it enhances the security of information or data to a very large extent.

| Combinations | Key Generation Time (milliseconds) | Encryption Time (milliseconds) | Decryption Time (milliseconds) |
|---|---|---|---|
| 1 | 1326 | 116 | 165 |
| 2 | 1327 | 102 | 168 |
| 3 | 1479 | 126 | 163 |
| 4 | 1520 | 140 | 172 |
| 5 | 1853 | 117 | 165 |
| 6 | 2064 | 123 | 166 |
| 7 | 2065 | 122 | 139 |
| 8 | 2076 | 143 | 168 |
| 9 | 2227 | 120 | 158 |
| 10 | 2236 | 129 | 191 |

Table 1. Performance of NHSA

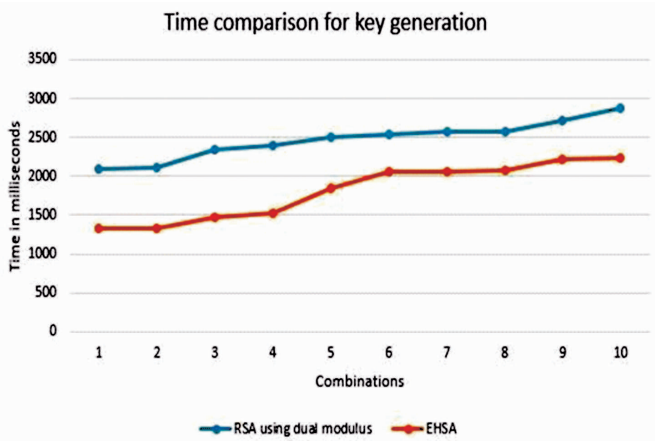


Figure 1. Analysis of Key Generation Time

| Combinations | Key Generation Time (milliseconds) | | NHSA vs Dual mod RSA (Speed up (%)) |
|---|---|---|---|
| | Dual mod RSA | NHSA | |
| 1 | 2101 | 1326 | 36.89 |
| 2 | 2115 | 1327 | 37.26 |
| 3 | 2343 | 1479 | 36.87 |
| 4 | 2402 | 1520 | 36.72 |
| 5 | 2508 | 1853 | 26.12 |
| 6 | 2539 | 2064 | 18.71 |
| 7 | 2574 | 2065 | 19.77 |
| 8 | 2579 | 2076 | 19.50 |
| 9 | 2717 | 2227 | 18.03 |
| 10 | 2884 | 2236 | 22.47 |

Table 2. Comparison of NHSA with Dual mod RSA in Terms of Key Generation Time

## Conclusion

In the proposed scheme, six prime numbers are used, and encryption and decryption has been done twice which might increase the computational complexity and the encryption and decryption time. It also enhances the security two times as compared to conventional RSA cryptosystem, and also it reduces the key generation time than RSA using dual modulus. Also, it is more secured against the Brutal Force Attack than conventional RSA.

Even if an intruder detects the private key, it is not possible to retrieve the information from the cipher text as the sender has encrypted the information twice.

Therefore, this proposed algorithm improves the performance and security of RSA using dual modulus.

References

[1]. Aboud, S. J., AL-Fayoumi, M. A., Al-Fayoumi, M., & Jabbar, H. S. (2008). An efficient RSA public key encryption scheme. In *Information Technology: New Generations, 2008. ITNG 2008. Fifth International Conference on* (pp. 127-130). IEEE.

[2]. Al-Hamami, A. H., & Aldariseh, I. A. (2012). Enhanced method for RSA cryptosystem algorithm. In *Advanced Computer Science Applications and Technologies (ACSAT), 2012 International Conference on* (pp. 402-408). IEEE.

[3]. Ambedkar, B. R., Gupta, A., Gautam, P., & Bedi, S. S. (2011). An efficient method to factorize the RSA public key encryption. In *Communication Systems and Network Technologies (CSNT), 2011 International Conference on* (pp. 108-111). IEEE.

[4]. Bhandari, A., Gupta, A., & Das, D. (2016). Secure algorithm for cloud computing and its applications. In *Cloud System and Big Data Engineering (Confluence), 2016 6th International Conference* (pp. 188-192). IEEE.

[5]. Devkota, D., Ghimire, P., Burris, J., & Alkadi, I. (2015). Comparison of security algorithms in cloud computing. In *Aerospace Conference, 2015 IEEE* (pp. 1-7). IEEE.

[6]. Dongjiang, L., Yandan, W., & Hong, C. (2012). The research on key generation in RSA public-key cryptosystem. In *Computational and Information Sciences (ICCIS), 2012 Fourth International Conference on* (pp. 578-580). IEEE.

[7]. Goel, A. (2017). Encryption algorithm using dual modulus. In *Computational Intelligence & Communication Technology (CICT), 2017 3rd International Conference on* (pp. 1-4). IEEE.

[8]. Jeeva, A. L., Palanisamy, D. V., & Kanagaram, K. (2012). Comparative analysis of performance efficiency and security measures of some encryption algorithms. *International Journal of Engineering Research and Applications (IJERA),* 2(3), 3033-3037.

[9]. Madaan, S., & Agrawal, R. K. (2012). Implementation of identity based distributed cloud storage encryption scheme using PHP and C for Hadoop File System. In *Tier 2 Federation Grid, Cloud & High Performance Computing Science (RO-LCG), 2012 5th Romania* (pp. 74-77). IEEE.

[10]. Osseily, H. A., Haidar, A. M., & Kassem, A. (2008). Implementation of RSA Encryption using Identical Modulus Algorithm. In *Information and Communication Technologies: From Theory to Applications, 2008. ICTTA 2008. 3rd International Conference on* (pp. 1-6). IEEE.

[11]. Pancholi, V. R., & Patel, B. P. (2016). Enhancement of cloud computing security with secure data storage using AES. *International Journal for Innovative Research in Science and Technology,* 2(9), 18-21.

[12]. Panda, P. K., & Chattopadhyay, S. (2017). A hybrid security algorithm for RSA cryptosystem. In *Advanced Computing and Communication Systems (ICACCS), 2017 4th International Conference on* (pp. 1-6). IEEE.

[13]. Ramaporkalai, T. (2017). Security Algorithms in Cloud Computing. *International Journal of Computer Science Trends and Technology (IJCST),* 5(2), 500-503.

[14]. Suja, G. J., & Jose, S. (2016). New approach for highly secured I/O transfer with data on timer streaming. In *Computing for Sustainable Global Development (INDIACom), 2016 3rd International Conference on* (pp. 885-889). IEEE.

[15]. Verma, S., & Garg, D. (2014). An Improved RSA Variant. *International Journal of Advancements in Technology,* 5(2), 161-169.

## ABOUT THE AUTHORS

*Manisha Kumari is pursuing her Bachelor of Engineering Degree in the Department of Computer Science and Engineering, School of Studies in Engineering and Technology at Guru Ghasidas University, Bilaspur, Chhattisgarh, India.*

*Deeksha Ekka is pursuing her Bachelor of Engineering Degree in the Department of Computer Science and Engineering, School of Studies in Engineering and Technology at Guru Ghasidas University, Bilaspur, Chhattisgarh, India.*

*Nishi Yadav is currently working as an Assistant Professor in the Department of Computer Science and Engineering, Schools of Studies in Engineering and Technology at Guru Ghasidas University, Bilaspur, Chhattisgarh, India.*