**Advances in Networks**
2018; 6(1): 40-47
http://www.sciencepublishinggroup.com/j/net
doi: 10.11648/j.net.20180601.14
ISSN: 2326-9766 (Print); ISSN: 2326-9782 (Online)

# Rivest Shamir Adleman Encryption Scheme Based on the Chinese Remainder Theorem

**Salifu Abdul-Mumin[1, *], Kazeem Alabge Gbolagade[2]**

[1]Department of Computer Science, University for Development Studies, Navrongo, Ghana

[2]Computer Science Department, Kwara State University, Malete, Nigeria

**Email address:**
smumin@uds.edu.gh (S. Abdul-Mumin); Kazeem.gbolagade@kwasu.edu.ng (K. A. Gbolagade)

[*]Corresponding author

**To cite this article:**
Salifu Abdul-Mumin, Kazeem Alabge Gbolagade. Rivest Shamir Adleman Encryption Scheme Based on the Chinese Remainder Theorem. *Advances in Networks.* Vol. 6, No. 1, 2018, pp. 40-47. doi: 10.11648/j.net.20180601.14

**Received**: February 19, 2018; **Accepted**: March 13, 2018; **Published**: April 4, 2018

**Abstract:** Sensitive information is transmitted across the internet every day and keeping such information as sacred is an important adventure. This is because malicious activities are on the increase as hackers are doing everything possible to steal such information. In this paper, we have implemented a new Rivest Shamir Adleman (RSA) encryption scheme based on the Chinese Remainder Theorem (CRT). The scheme consists of two level of encryption and two level of decryption. The first level of encryption is the classical RSA encryption and in the second level of encryption, we used forward conversion technique in Residue Number System. In the first level of decryption, we employed the CRT and the classical RSA decryption process is used for the second level of decryption. This new scheme will ensure that smaller messages, $m$ for which $c=m^e<n$ can be encrypted which would otherwise not be able to be encrypted with the classical RSA encryption scheme. The proposed scheme is evaluated with the state of the art and the classical RSA cryptosystem. The proposed scheme performs better than the classical RSA cryptosystem for smaller messages in terms of security and performs better than the state of the art in terms of delay and cost. The private key length in the new scheme is also enhanced by 1-bit as against the state of the art.

**Keywords:** Security, Encryption, Decryption, Rivest Shamir Adleman, Residue Number System, Chinese Remainder Theorem

## 1. Introduction

Confidentiality and security requirements are becoming more rigorous as malicious activities are on the increase every day. The use of internet applications to ease and to maximize returns are gradually gaining more attention in the areas of e-business, military surveillance, medicine and education. During the last decade, fast hardware implementations of public key cryptosystems have been widely studied [1-4]. Different approaches have been proposed to accelerate the implementation of RSA. For the deciphering, a well-known solution performs the computations over $\mathbb{Z} = p\mathbb{Z}$ and $\mathbb{Z} = q\mathbb{Z}$ independently and reconstructs the final result via the Chinese Remainder Theorem (CRT) [1, 5]. More recently, other CRT-based solutions have been proposed [6-9]. They all use a quite similar version of the Montgomery multiplication based on the Residue Number System (RNS) which is well-adapted to fast parallel arithmetic [10].

Most public-key cryptosystems currently in use, including RSA, depend on the intractability of factoring and computing discrete logarithms. However, in 1994, Shor proposed efficient quantum algorithms to solve these problems [11].

This implies when the quantum computer is finally built, current public key cryptosystems will be broken. Hence the need for research on efficient post-quantum public-key cryptosystems is most valuable.

In some instances in the classical RSA cryptosystem, it is easy to compute modular roots without knowledge of the prime factors. For example, if $m$ is known to be very small, such that $c = m^e < n$, then $m$ can be recovered from $c$ by taking $e^{th}$ roots over the integers, which is easy.

In this paper, we proposed and implement a CRT based RSA encryption system which will have a two level

encryption process and a two level decryption process. This will further enhance the classical RSA encryption where smaller messages that could not be encrypted by the classical RSA encryption can be encrypted by the new proposal. The private key will also be enhanced in the new proposal.

## 2. Background of RNS

Residue Number Systems (RNS) have been widely studied and used in many applications, from digital signal processing to multiple precision arithmetic [10, 12, 13] RNS offers very useful applications in addition, subtraction, and multiplication dominated arithmetic operations, for example, digital filtering, fast Fourier Transform (FFT), image processing etc. This is due to the inherent properties of RNS such as parallelism, modularity, fault tolerance, and carry free operations [12, 14, 15]. Additionally, it has been shown that RNS based processors can even reduce power dissipation in very large scale integrated circuits system design [16]. This area of research have not seen wide range of usage due to the following RNS difficult arithmetic operations: Overflow detection, sign detection, magnitude comparison among others.

The RNS is defined in terms of a set of relatively prime moduli, $P = \{m_1, m_2, .....m_k\}$, where $GCD(m_i, m_j) = 1$, for i ≠ j. $M = \prod_{i=1}^{k} m_i$ denotes the moduli set. Any integer X in the range [0, M) where $M = \prod_{i=1}^{k} m_i$ can be uniquely and unambiguously represented by the residue sequence $X \leftrightarrow (x_1, x_2 ....x_k)$ where $x_i = X \bmod m_i, i = 1, 2, .....k$ is the residue modulus $m_i$ of X. The range [0, M) is called dynamic range or the legitimate range of X. [10, 17]

## 3. RSA Encryption System and Related Works

Public-key cryptography is instrumental to modem Security functions such as authentication and key exchange [18]. Public-key cryptography, or asymmetric cryptography, is an encryption scheme that uses pairs of keys: one key known as the public key that may be disseminated widely and is meant for encryption and the other known as the private key which is known to only the owner is for decryption. RSA is one of the public-key encryption, and is widely used for securing sensitive information, especially when being sent over an insecure \ channel like the Internet. RSA public-key cryptosystem was invented at MIT in 1977 by Ronald Rivest, Adi Shamir and Leonard Adleman. The public key in this cryptosystem consists of the value n, which is called the modulus, and the value, which is called the public exponent.The private key consists of the modulus n and the valued,which is called the private exponent [19]. RSA public-key encryption scheme is define as follows:

(i) GenModulus($1^n$) runs GenRSA($1^n$) to obtain n; e, and d. The public key is (n, e) and the private key is (n, d)

(ii) Encpk(m), on input a public key pk = (N, e) and a message m ϵ $Z_N$, computes the ciphertext c:= [$m^e$ mod N].

(iii) Decsk(c), on input a private key sk = (N, d) and a cipher text, c ϵ $Z_N$, computes the message m:= [$c^d$ mod N]

Different RNS approaches have been proposed to accelerate the implementation of RSA. K. C. Posch et al, in 1995 reconstructed the final result (decryption process via the Chinese Remainder Theorem. [6] S. Kawamura et al. in 2000, H. Nozaki, et al., in 2001 all use a quite similar version of the Montgomery multiplication based on the Residue Number System (RNS). [7] In 2016, Salifu and Gbolagade proposed a similar scheme where mixed radix conversion was used for the second level of decryption. [20]

## 4. Proposed Cryptosystem

Our proposed cryptosystem has two stages of encryption; the first stage is the traditional RSA encryption and the second stage is the encryption using the proposed moduli set from RNS. Forward conversion in RNS is the second stage encryption and reverse conversion is the first stage decryption and finally, the traditional RSA decryption will serve as the second stage deciphering process.

## 5. Implementation of Stage-2 Encryption Scheme (Forward Converter) Using the Moduli Set $\left(2^n - 1, 2^n + 1, 2^{2n}\right)$

$m_i = 2^n - 1, m_2 = 2^n + 1, m_3 = 2^{2n}$. X is within the dynamic range [0, ($2^{4n}$–$2^{2n}$) -1]. Where the upper end of the range is ($m_1m_2m_3$), is uniquely defined by a residue set ($r_1$, $r_2$, $r_3$), where $r_i = |X|_{mi}$ and X is an 4n – bit number.

$$X = x_{4n-1}x_{4n-2}.....x_n x_{n-1}.....x_1x_0 \tag{1}$$

The residue $r_3$ is the easiest to compute. The 2n last significant bits constitute the remainder when X is divided by $2^{2n}$. Hence $r_3$ is the number represented by the least significant 2n bits of X. These bits are obtained by nominally shifting to the right by 2n bits.

In order to determine the residues $r_1$ and $r_2$, we first partition X into three blocks $B_1$, $B_2$, and $B_3$,. $B_1$ and $B_2$ are n-bit wide and $B_3$ is 2n-bit wide.

$$\begin{aligned} B_1 &= \sum_{j=3n}^{4n-1} x_j 2^{j-n} \\ B_2 &= \sum_{j=n+2}^{3n-1} x_j 2^{j+(n-2)} \\ B_3 &= \sum_{j=0}^{n+1} x_j 2^j \end{aligned} \tag{2}$$

$$\text{Then } X = 2^{2n}B_1 + 2^{2n}B_2 + B_3 \tag{3}$$

Let the residue representation of X be ($r_1$, $r_2$, $r_3$). The residue $r_1$ is obtained as

$$
\begin{aligned}
r_1 &= |X|_{2^n-1} \\
&= |2^{2n}B_1 + 2^{2n}B_2 + B_3|_{2^n-1} \\
&= \| 2^{2n}B_1 |_{2^n-1} + |2^{2n}B_2|_{2^n-1} + |B_3|_{2^n-1} |_{2^n-1} \\
&= \| (2^n * 2^n)B_1 |_{2^n-1} + |(2^n * 2^n)B_2|_{2^n-1} + |B_3|_{2^n-1} |_{2^n-1} \quad (4) \\
&= \| (2^n - 1 + 1) * (2^n - 1 + 1)B_1 |_{2^n-1} \\
&+ |(2^n - 1 + 1) * (2^n - 1 + 1)B_2|_{2^n-1} + |B_3|_{2^n-1} |_{2^n-1} \\
r_1 &= |B_1 + B_2 + B_3|
\end{aligned}
$$

The residue $r_2$ is obtained as;

$$
\begin{aligned}
r_2 &= |X|_{2^n+1} \\
&= |2^{2n}B_1 + 2^{2n}B_2 + B_3|_{2^n+1} \\
&= \| 2^{2n}B_1 |_{2^n+1} + |2^{2n}B_2|_{2^n+1} + |B_3|_{2^n+1} |_{2^n+1} \\
&= \| (2^n * 2^n)B_1 |_{2^n+1} + |(2^n * 2^n)B_2|_{2^n+1} + |B_3|_{2^n+1} |_{2^n+1} \quad (5) \\
&= \| (2^n + 1 - 1) * (2^n + 1 - 1)B_1 |_{2^n+1} \\
&+ |(2^n + 1 - 1) * (2^n + 1 - 1)B_2|_{2^n+1} + |B_3|_{2^n+1} |_{2^n+1} \\
r_2 &= |B_1 + B_2 + B_3|
\end{aligned}
$$

The hardware realisation of the second encryption process (forward conversion) is based on equation (4), (5) and the value of $r_3 = B_3$ and it is represented below;

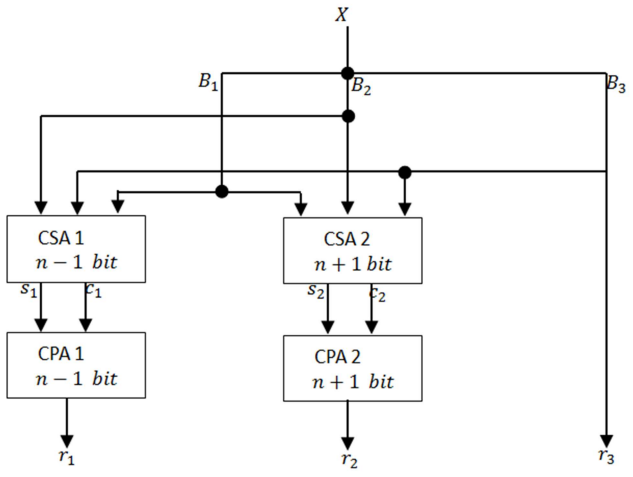


***Figure 1.*** *Second Stage Encryption (forward converter) for the Moduli Set* $\left(2^n - 1, 2^n + 1, 2^{2n}\right)$.

In figure 1, B1, B2 and B3 are added using (n-1)-bit wide CSA1 with EAC to obtain s1 and c1 which are added using a normal (n-1)-bit wide CPA1 to obtain r1. Similarly B1, B2, and B3 are added using (n+1)-bit wide CSA2 with EAC to obtain s2 and c2 which are then added using a normal (n+1)-bit wide CPA2 to obtain r2. B3 is simply r3. The propagation delay of the circuit is $(2n + 3)\Delta_{FA}$ and the area (cost) of the scheme is $4n\Delta_{FA}$.

### *5.1. Implementation of Stage-1 Decryption Scheme (Reverse Converter) Using the Moduli Set* $\left(2^n - 1, 2^n + 1, 2^{2n}\right)$

The second level decryption of the proposed algorithm is done using the moduli set $\left(2^n - 1, 2^n + 1, 2^{2n}\right)$

The residue number $(r_1, r_2, r_3)$ can be converted into the decimal number X according to the Chinese Remainder theorem (CRT) as follows;

$$X = \left| \sum_{i=1}^{k} M_i \left| M_i r_i \right| \right|_{m_i} \Bigg|_M \quad (6)$$

For k = 3

$$X = \left| \sum_{i=1}^{3} M_i \left| M_i r_i \right| \right|_{m_i} \Bigg|_M \quad m_i = 2^n - 1, m_2 = 2^n + 1, m_3 = 2^{2n}$$

Theorem 1: Given the moduli set $(m_1, m_2, m_3)$ with $m_i = 2^n - 1, m_2 = 2^n + 1, m_3 = 2^{2n}$, the following hold true

$$\left| \left(m_2 m_3\right)^{-1} \right|_{m_1} = 2^{n-1} \quad (7)$$

$$\left| \left(m_1 m_3\right)^{-1} \right|_{m_2} = 2^{n-1} \quad (8)$$

$$\left| \left(m_1 m_2\right)^{-1} \right|_{m_3} = -1 \quad (9)$$

Proof

If $\left| \left(m_2 m_3\right)^{-1} \right|_{m_1} = 2^{n-1}$ then

$\left| 2^{n-1}(2^n + 1)(2^{2n}) \right|_{2^n-1} = 1$. Considering the left hand side,

$$
\begin{aligned}
&\left| (2^{2n-1} + 2^{n-1})(2^{2n}) \right|_{2^n-1} \\
&\left| 2^{4n-1} + 2^{3n-1} \right|_{2^n-1} \\
&\left| \left| 2^{4n-1} \right|_{2^n-1} + \left| 2^{3n-1} \right|_{2^n-1} \right|_{2^n-1} \\
&\left| \left| 2^{4n} * 2^{-1} \right|_{2^n-1} + \left| 2^{3n} * 2^{-1} \right|_{2^n-1} \right|_{2^n-1} \\
&\frac{1}{2} + \frac{1}{2} = 1
\end{aligned}
$$

As required.

Also if $\left| \left(m_1 m_3\right)^{-1} \right|_{m2} = 2^{n-1}$ then,

$\left| 2^{n-1}(2^n - 1)(2^{2n}) \right|_{2^n+1} = 1$. Considering LHS, we have

$$\left|(2^{2n-1}-2^{n-1})(2^{2n})\right|_{2^n+1}$$
$$\left|2^{4n-1}-2^{3n-1}\right|_{2^n+1}$$
$$\left|\left|2^{4n-1}\right|_{2^n+1}-\left|2^{3n-1}\right|_{2^n+1}\right|_{2^n+1}$$
$$\left|\left|2^{4n}*2^{-1}\right|_{2^n+1}-\left|2^{3n}*2^{-1}\right|_{2^n+1}\right|_{2^n+1}$$
$$\frac{1}{2}-(-\frac{1}{2})=1$$

As required.

Similarly, if $\left|(m_1m_2)^{-1}\right|_{m_3}=-1$ then $\left|-1(2^n-1)(2^n+1)\right|_{2^{2n}}=1$. Considering LHS, we have

$$\left|(-2^n+1)(2^n+1)\right|_{2^{2n}}$$
$$\left|-2^{2n}-2^n+2^n+1\right|_{2^{2n}}$$
$$\left|-2^{2n}+1\right|_{2^{2n}}$$
$$\left|-\left|2^{2n}\right|_{2^{2n}}+\left|1\right|_{2^{2n}}\right|_{2^{2n}}$$
$$0+1=1$$

As required. This implies (7), (8), and (9) are true.

The following relations introduce necessary preliminary information which enables a further simplification of the traditional CRT in the subsequent theorem. Given the moduli set $(m_1,m_2,m_3)$ with $m_i=2^n-1, m_2=2^n+1, m_3=2^{2n}$, the following hold true:

$$m_3=2^nm_1+2^n \tag{10}$$

$$m_3=2^nm_2-2^n \tag{11}$$

$$m_1=m_2-2 \tag{12}$$

The following theorem introduces a simplified way to compute the decimal equivalent of the RNS number.

Theorem2: The decimal equivalent of the residue $(x_i,i=1,3)$ for the moduli set $(m_i,i=1,3)$ in the form $\left(2^n-1,2^n+1,2^{2n}\right)$ assuming $X\in[0,M-1]$ can be computed as follows;

$$X=m_2\left\lfloor\frac{X}{m_2}\right\rfloor+x_2 \tag{13}$$

Where

$$\left\lfloor\frac{X}{m_2}\right\rfloor=\left|2^{2n-1}x_1-2^{2n-1}x_2+m_1\left|-x_3+2^{2n-1}x_2+2^{2n-1}x_1\right|_{m_3}\right|_{m_3m_1} \tag{14}$$

Proof:

Since equation (13) follows the basic integer division definition in RNS, which is always true, we only need to show the correctness of equation (14). For k = 3, equation (6) becomes

$$X=\left|\sum_{i=1}^{3}M_i\left|M_ir_i\right|_{m_i}\right|_M \tag{15}$$

By substituting equation (7), (8), and (9) into (15), we have

$$X=\left|m_2m_3(2^{n-1})x_1+m_1m_3(2^{n-1})x_2+m_1m_2(-1)x_3\right|_M$$

And by substituting equation (10) and (11) in the above equation, we obtain;

$$\begin{aligned}X&=\left|2^{n-1}m_2(2^nm_1+2^n)x_1+2^{n-1}m_1(2^nm_2-2^n)x_2+m_1m_2(-x_3)\right|_M\\X&=\left|2^{2n-1}m_2m_1x_1+2^{2n-1}m_2x_1+2^{2n-1}m_1m_2x_2-2^{2n-1}x_2-m_1m_2x_3\right|_M\end{aligned} \tag{16}$$

Equitation (16) can be further simplified by using the following lemma presented in [21][22].

$$\left|am_1\right|_{m_1m_2}=m_1\left|a\right|_{m_2} \tag{17}$$

Applying equation (17), equation (16) becomes;

$$X=\left|2^{2n-1}m_2x_1-2^{2n-1}m_1x_2+m_1m_2\left|-x_3+2^{2n-1}x_2+2^{2n-1}x_1\right|_{m_3}\right|_M \tag{18}$$

If equation (12) is utilised in equation (18), we have;

$$X=\left|2^{2n-1}m_2x_1-2^{2n-1}(m_2-2)x_2+m_1m_2\left|-x_3+2^{2n-1}x_2+2^{2n-1}x_1\right|_{m_3}\right|_M$$
$$X=\left|2^{2n-1}m_2x_1-2^{2n-1}m_2x_2+2^{2n}x_2+m_1m_2\left|-x_3+2^{2n-1}x_2+2^{2n-1}x_1\right|_{m_3}\right|_M$$

Dividing both sides of the equation by $m_2$ and taking the floor, we shall have;

$$\left\lfloor \frac{X}{m_2} \right\rfloor = \left| 2^{2n-1}x_1 - 2^{2n-1}x_2 + m_1 \mid -x_3 + 2^{2n-1}x_2 + 2^{2n-1}x_1 \mid_{m_3} \right|_{m_3 m_1}$$

As required.

The hardware complexity of equation (14) can be further reduced by using the following properties from [17][21].

Property 1: Modulo $(2^s - 1)$ multiplication of a residue number by $2^t$, where s and t are positive integers is equivalent to t-bit circular left shifting.

Proper 2: Modulo $(2^s - 1)$ of a negative number is equivalent to the one's complement of the number, which is obtained by subtracting the number from $(2^s - 1)$.

Suppose that equation (14) is represented by

$$\begin{aligned} \left\lfloor \frac{X}{m_2} \right\rfloor &= \left| 2^{2n-1}x_1 - 2^{2n-1}x_2 + (2^n - 1)A \right|_{m_3 m_1} \\ &= \left| 2^{2n-1}x_1 - 2^{2n-1}x_2 + 2^n A - A \right|_{2^{3n} - 2^{2n}} \end{aligned} \quad (19)$$

Where

$$A = \left| u_1 + u_2 + u_3 \right|_{2^{2n}} \quad (20)$$

The addition $u_1 + u_2 + u_3$ modulo $2^{2n}$ is easier to compute. The 2n last significant bits constitute the remainder when $u_1 + u_2 + u_3$ is divided by $2^{2n}$ which is the number represented by the least significant 2n bits of X. These bits are obtained by nominally shifting to the left by 2n bits.

For simplicity sake, let's represent equation (19) by the following:

$$\left\lfloor \frac{X}{m_2} \right\rfloor = \left| B_1 + B_2 + B_3 \right|_{2^{3n} - 2^{2n}} \quad (21)$$

Where

$$\begin{aligned} B_1 &= -2^{2n-1}x_2 \\ B_2 &= 2^n A + 2^{2n-1}x_1 \\ B_3 &= -A \end{aligned} \quad (22)$$

Let the binary representation of the residues be the following;

$$\begin{aligned} x_1 &= (x_{1,n-1}x_{1,n-2}.....x_{1,1}x_{1,0}) \\ x_2 &= (x_{2,n}x_{2,n-1}.....x_{2,1}x_{2,0}) \\ x_3 &= (x_{3,2n-1}....x_{3,n-1}.....x_{3,1}x_{3,0}) \end{aligned}$$

In equation (20), $u_1$, $u_2$, and $u_3$ are represented as follows;

$$\begin{aligned} u_1 &= \left| -x_3 \right|_{2^{2n}} \\ &= \left| -(x_{3,2n-1}....x_{3,n-1}.....x_{3,1}x_{3,0}) \right|_{2^{2n}} \\ &= -\underbrace{(x_{3,2n-1}....x_{3,n-1}.....x_{3,1}x_{3,0})}_{2n} \\ &= \underbrace{(\overline{x}_{3,2n-1}....\overline{x}_{3,n-1}.....\overline{x}_{3,1}\overline{x}_{3,0})}_{2n} \end{aligned}$$

$$\begin{aligned} u_2 &= \left| 2^{2n-1}x_2 \right|_{2^{2n}} \\ &= \left| 2^{2n-1}(x_{2,n}x_{2,n-1}.....x_{2,1}x_{2,0}) \right|_{2^{2n}} \\ &= \underbrace{(x_{2,n}x_{2,n-1}.....x_{2,1}x_{2,0}}_{2n}\underbrace{00.....00)}_{2n-1} \\ &= \underbrace{(x_{2,n}x_{2,n-1}.....x_{2,1}x_{2,0}00....00)}_{2n} \end{aligned}$$

$$\begin{aligned} u_3 &= \left| 2^{2n-1}x_1 \right|_{2^{2n}} \\ &= \left| 2^{2n-1}(x_{1,n-1}x_{1,n-2}.....x_{1,1}x_{1,0}) \right|_{2^{2n}} \\ &= \underbrace{(x_{1,n-1}x_{1,n-2}.....x_{1,1}x_{1,0}}_{2n}\underbrace{00.....00)}_{2n-1} \\ &= \underbrace{(x_{1,n-1}x_{1,n-2}.....x_{1,1}x_{1,0}00.....00)}_{2n} \end{aligned}$$

Given that A has the following binary representation

$$A = \underbrace{(a_n a_{n-1}.....a_1 a_0)}_{2n}$$

Then $B_2$ will be given by

$$\begin{aligned} &(2^n(a_n a_{n-1}.....a_1 a_0)) + (2^{2n-1}(x_{1,n-1}x_{1,n-2}.....x_{1,1}x_{1,0})) \\ &= \underbrace{(a_n a_{n-1}.....a_1 a_0}_{2n}\underbrace{00.....00)}_{n} + \underbrace{(x_{1,n-1}x_{1,n-2}.....x_{1,1}x_{1,0}}_{n}\underbrace{00.....00)}_{2n-1} \\ &= \underbrace{(a_n a_{n-1}.....a_1 a_0 00....00)}_{3n} + \underbrace{(x_{1,n-1}x_{1,n-2}.....x_{1,1}x_{1,0}00.....00)}_{3n-1} \\ &= \underbrace{(a_n a_{n-1}.....a_1 a_0 x_{1,n-1}x_{1,n-2}.....x_{1,1}x_{1,0})}_{3n}.....(18) \end{aligned}$$

In equation (16), in order to carry out the summation, $B_1$ and $B_3$ must have equal number of bits (ie 3n-bits) as $B_2$. They are represented as follows;

$$\begin{aligned} B_1 &= -2^{2n-1}x_2 \\ &= -(2^{2n-1}(x_{2,n}x_{2,n-1}.....x_{2,1}x_{2,0})) \\ &= -\underbrace{(x_{2,n}x_{2,n-1}.....x_{2,1}x_{2,0}}_{n+1}\underbrace{00....00)}_{2n-1} \\ &= \underbrace{(\overline{x}_{2,n}\overline{x}_{2,n-1}.....\overline{x}_{2,1}\overline{x}_{2,0}11....11)}_{3n} \\ B_3 &= -A \\ &-\underbrace{(00.....00}_{n}\underbrace{a_n a_{n-1}.....a_1 a_0)}_{2n} \\ &= \underbrace{(11.....11\overline{a}_n\overline{a}_{n-1}.....\overline{a}_1\overline{a}_0)}_{3n} \end{aligned}$$

### *5.2. Hardware Implementation*

The hardware implementation of the second level of decryption stage is based on equation (20) and (21). In figure 2, u1, u2, and u3 are added by CSA1 with end around carry (EAC) producing s1 and c1. Next, these must be added modulo $2^{2n}$ in order to obtain A, which is easier to compute. The 2n last significant bits constitute the remainder when $u_1 + u_2 + u_3$ is divided by $2^{2n}$ which is the number represented by the least significant 2n bits of X. These bits are obtained by nominally shifting to the left by 2n bits, which does not require hardware

resources. The results of these CPAs are passed on to a multiplexer (MUX 1) which would then pass either of them down. MUX will pass on the result of CPA 1 if the carry out of CSA 1 is a '1', otherwise the result of CPA 2 is passed on.

$B_2$ is easily obtained by concatenating the result of n-bit left shift of A and the result on (2n-1) –bit left shift of x1. This concatenation does not require any hardware resources. The three operands $B_1$, $B_2$ and $B_3$ are added using CSA2 with EAC to obtain s4 and c4. These are now added using CPA3. The final result, which is computed based on Equation (13) is obtained just by a shift and a concatenation operation with no computational hardware.

CSA1, CPA1 and CPA2 each require an area of $2n\Delta_{FA}$ while CSA2 and CPA3 each will require an area of (3n-1) $\Delta_{FA}$. Therefore, in order to obtain X will require a total area of $(12n-2)\Delta_{FA}$

Regarding the delay, each CSA (i.e. CSAs 1 and 2) impose a delay of $D_{FA}$ while the CPA pair 1 and 2 impose a delay of $4nD_{FA}$ since they are in parallel, thus the delay imposed on computing X is $10nD_{FA}$ . Below is the hardware architecture of the scheme.

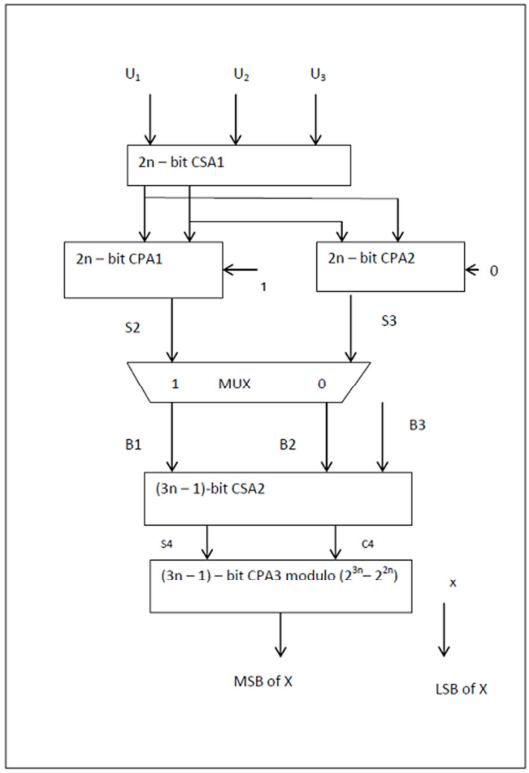


***Figure 2.*** *Architecture for First Stage Decryption (Reverse converter) for the moduli set* $\left(2^n-1, 2^n+1, 2^{2n}\right)$.

Example To Illustrate The Proposed Scheme

The scheme is illustrated with the example below;

Let p = 101, q = 113, e = 3 (e is an odd public exponent between 3 and n – 1 that is relatively prime to p – 1 and q – 1)

Then

$N = pq = 101*113 = 11413, \Phi(N) = (p-1)(q-1) = 11200$ which implies $e.d = 1 \bmod \Phi(N)$. Therefore d = 7467. To encrypt the binary message m = 10111 with respect to the key $p_k = (N = 11413, e = 3)$ , taking $m$ as 23 (hence an element of Zn = 11200) in the natural way. Computing $c_1 = \left|23^3\right|_{11413} = 754$ . For $c_1$to fall within the dynamic range, we choose n = 3. $PK_2$ = (7, 9, 64). $C_2$ is calculated as follows: $\left|754\right|_7 = 5, \left|754\right|_9 = 7, \left|754\right|_{64} = 50$

$$C_2 = (5, 7, 50)$$

For first-level decryption, we have the following:

Moduli set (private key-2, pk2) = (7, 9, 64), dynamic range, M = 4032, $M_1$ = 576, $M_2$ = 448, $M_3$ = 63

$$\left|M_1^{-1}\right|_{m_1} = \left|576^{-1}\right|_7 = 4$$
$$\left|M_2^{-1}\right|_{m_2} = \left|448^{-1}\right|_9 = 4$$
$$\left|M_3^{-1}\right|_{m_3} = \left|63^{-1}\right|_{64} = 63$$

$$c_1 = \left|\left|576*4*5\right|_{4032} + \left|448*4*7\right|_{4032} + \left|63*63*50\right|_{4032}\right|_{4032}$$
$$= \left|3456+448+882\right|_{4032} = 754$$

For the second-level decryption with respect to the private key, (N = 11413, d= 7467), we have the following:

$\left|754^{7467}\right|_{11413} = 23$ As required

# 6. Performance Evaluation

***Table 1.*** *RSA Results Evaluation.*

| Plaintext (m) | $m^e$ | Cipher text (c) |
|---|---|---|
| 30 | 27000 | 4174 |
| 25 | 15625 | 4212 |
| 20 | 8000 | 8000 |
| 15 | 3375 | 3375 |
| 10 | 1000 | 1000 |
| 5 | 125 | 125 |

In the table above, transmitting 30 and 25 are secured because $m^e$ is greater than n = 11413. Transmitting 20, 15, 10 and 5 however are not secured because $m^e$ is less than n = 11413.

Taking eth root of their corresponding cipher text will recover the original messages without the knowledge of the prime numbers. The private key length is a 13-bit number.

***Table 2.*** *State of the art [20] Results Evaluation.*

| Plaintext (m) | $m^e$ | Cipher text$_1$ ($c_1$) | Cipher text$_2$ ($c_2$) |
|---|---|---|---|
| 30 | 27000 | 4174 | (2, 0, 6) |
| 25 | 15625 | 4212 | (1, 2, 2) |
| 20 | 8000 | 8000 | (0, 0, 5) |
| 15 | 3375 | 3375 | (0, 3, 1) |
| 10 | 1000 | 1000 | (1, 0, 6) |
| 5 | 125 | 125 | (2, 1, 6) |

In the table above, transmitting all the plaintexts are secured because the second level of encryption will transform $C_1$ into $C_2$. This will not only cause more confusing to the hacker, but also enable the plaintexts, 20, 15, 10 and 5 to be encrypted.

The private key-length is a 20- bit number.

Total delay for stage-2 encryption and stage-1 decryption is $12n + 10\ D$ and the total cost for both forward and reverse conversion is $15n + 5\ A$

***Table 3.*** *Proposed Scheme Results Evaluation with the module set* $\left(2^n-1, 2^n+1, 2^{2n}\right)$.

| Plaintext (m) | $m^e$ | Cipher text$_1$ ($c_1$) | Cipher text$_2$ ($c_2$) |
|---|---|---|---|
| 30 | 27000 | 4174 | (2, 0, 0) |
| 25 | 15625 | 4212 | (1, 4, 14) |
| 20 | 8000 | 8000 | (0, 2, 4) |
| 15 | 3375 | 3375 | (0, 0, 15) |
| 10 | 1000 | 1000 | (1, 0, 8) |
| 5 | 125 | 125 | (2, 0, 3) |

In the table above, transmitting all the plaintexts are secured because the second level of encryption will transform $C_1$ into $C_2$. This will also ensure that the plaintexts 20, 15, 10 and 5 will be encrypted. The private key-length in the proposed scheme is a 21- bit number.

In the scheme proposed in [20], the total delay for stage-2 encryption and stage-1 decryption is $(12n + 10)_{FA}$ and the total cost or area for both forward and reverse conversion is $(15n + 5)D_{FA}$

In the proposed scheme, the total delay for stage-2 encryption and stage-1 decryption is $(12n +3)D_{FA}$ whiles the total cost or area is $(16n – 2)D_{FA.}$

***Table 4.*** *Percentage Gain Of Delay and Area of the Proposed Scheme.*

| n | State of the art | Proposed Scheme | State of the art | Proposed Scheme | Percentage Gain | |
|---|---|---|---|---|---|---|
| | Delay | Delay | Cost | Cost | Delay | Cost |
| 2 | 34 | 27 | 35 | 30 | 20.588 | 14.286 |
| 3 | 46 | 39 | 50 | 46 | 15.217 | 8 |
| 4 | 58 | 51 | 65 | 62 | 12.069 | 4.615 |
| 5 | 70 | 63 | 80 | 78 | 10 | 2.5 |

The prosed scheme outperformed the classical RSA encryption scheme in terms of security with respect to messages such that $m^e < N$. The key-length is also enhanced in the proposed scheme. The proposed scheme outperformed the scheme proposed in [20] in terms of propagation delay and cost. The dynamic range of the proposed scheme is also more than the dynamic range in the scheme proposed in [20]. This means that more information can be represented in the proposed scheme than that in [20].

# 7. Conclusion

A new RSA cryptosystem have been implemented using Chinese Remainder Theorem. The system has two level of encryption and two level of decryption. The first level made use of the classical RSA encryption scheme and in the second level, a moduli set which are relatively smaller are used. CRT is used in the first stage of decryption and the classical RSA decryption process is employed for the final stage. Evaluation performance is illustrated and smaller message that could not be encrypted using the classical RSA encryption scheme has the ability to be encrypted with the new scheme. The moduli set used is part of the private key which enhances the security of the proposed system.

# References

[1] Laurent Lambert and Jean-Claude Bajard: A Full RNS Implementation of RSA, IEEE TRANSACTIONS ON COMPUTERS, VOL. 53, NO. 5, MAY 2004.

[2] E. F. Brickell, “A Survey of Hardware Implementation of RSA,” Advances in Cryptology, Proc. CRYPTO ’vol: 89, pp. 368-370, 1990.

[3] S. E. Eldridge and C. D. Walter, “Hardware Implementation of Montgomery’s Modular Multiplication Algorithm,” IEEE Trans. Computers, vol. 42, no. 6, pp. 693-699, June 1993.

[4] M. Shand and J. Vuillemin, “Fast Implementation of RSA Cryptography,” Proc. 11th IEEE Symp. Computer Arithmetic, pp. 252-259, June 1993.

[5] J.-J. Quisquater and C. Couvreur, “Fast Decipherment Algorithm for RSA Public-Key Cryptosystem,” IEE Electronics Letters, vol. 18, no. 21, pp. 905-907, Oct. 1982.

[6] K. C. Posch and R. Posch, “Modulo Reduction in Residue Number Systems,” IEEE Trans. Parallel and Distributed Systems, vol. 6, no. 5, pp. 449-454, May 1995.

[7] S. Kawamura, M. Koike, F. Sano, and A. Shimbo, “Cox-Rower Architecture for Fast Parallel Montgomery Multiplication,” Advances in Cryptology, Proc. EUROCRYPT 2000, pp. 523-538, May 2000.

[8] H. Nozaki, M. Motoyama, A. Shimbo, and S. Kawamura, “Implementation of RSA Algorithm Based on RNS Montgomery Multiplication,” Proc. Cryptographic Hardware and Embedded Systems (CHES 2001), pp. 364-376, Sept. 2001.

[9] J.-C. Bajard, L.-S. Didier, and P. Kornerup, “Modular Multiplication and Base Extension in Residue Number Systems,” Proc. 15th IEEE Symp. Computer Arithmetic, N. Burgess, ed., pp. 59-65, June 2001.

[10] F. J. Taylor, “Residue Arithmetic: A Tutorial with Examples,” Computer, vol. 17, no. 5, pp. 50-62, May 1984.

[11] P. W. Shor. Algorithms for Quantum Computation: Discrete Logarithms and Factoring. In IEEE Symp. on Found. of Comp. Sci., p. 124-134, 1994.

[12] Szabo N. S. and Tanaka R. I. Residue Arithmetic and its Applications to Computer Technology, McGraw Hill. New Tork, 1967.

[13] D. E. Knuth, The Art of Computer Programming, Vol. 2: Semi numerical Algorithms, third ed. Addison-Wesley, 1997.

[14] Edem K. Bankas and Kazeem A. Gbolagade: A New Efficient RNS Reverse Converter for the 4-Moduli Set $\{2^n, 2^n+1, 2^n-1, 2^{2n+1}-1\}$ International Journal of Computer, Electrical, Automation, Control and Information Engineering Vol:8, No:2, 2014 pp: 328 332.

[15] K. A. Gbolagade, R. Chaves, L. Sousa, and S. D. Cotofana. *Residue-to-Binary Converters for the* $\{2^{2n+1}-1, 2^{2n}+1, 2^n-1\}$ *Moduli set*. 2nd IEEE International Conference on Adaptive Science and Technology. pp. 26 - 33, Accra, Ghana. December, 2009.

[16] Shew M., Lin S., Chen C., and Yang S. *An Efficient VLSI Design for a Residue to Binar Converter for General Balance moduli* $\{2^n-3, 2^n+1, 2^n-1, 2^{2n+1}+3\}$. IEEE Transactions on Circuits and Systems -II Express Briefs, Vol. 51, No. 3, March, 2004, pp. 152-155.

[17] Amir Sabbagh Molahosseini and Keivan Nav: New Arithmetic Residue to Binary Converters, International Journal of Computer Sciences and Engineering Systems, Vol. 1, No. 4, pp. 295-299, October, 2007.

[18] William L. Freking and Keshab K. Parhi. Modular Multiplication in the Residue Number System with Application to Massively-Parallel Public-Key Cryptography Systems, Conference of Circuit, Systems and Computers, Vol. 2. pp. 1339-1343.

[19] Burt Kaliski: "The Mathematics of the RSA Public-Key Cryptosystem", RSA Laboratories. April 9, 2006.

[20] Salifu Abdul-Mumin and Kazeem A. Gbolagade: Mixed Radix Conversion Based RSA Encryption System, Internation Journal of Computer Applications vol 150 no. 1 Sept 2016.

[21] Y. Wang. Residue-to-binary converters based on new chinese remainder theorems. IEEE Trans. on Circuits and Systems-II: Analog and Digital Signal Processing, Vol. 47, No. 3, pp. 197-205, March, 2000.

[22] Kazeem A. Gbolagade: Effective Reverse Conversion in Residue Number System Processors, PhD Theses presented in TuDelf, Netherland, 2010.