EFFICIENT RSA CRYPTOSYSTEM DECRYPTION BASED ON CHINESE REMAINDER THEOREM AND STRONG PRIME

[1] Saheed Yakub Kayode and [2] Gbolagade Kazeem Alagbe

[1] Department of Physical Sciences, Al-Hikmah University, Ilorin, Nigeria
[2] Department of Computer Science, Kwara State University, Malete, Nigeria

Corresponding Author: Gbolagade Kazeem Alagbe, kazeem.gbolagade@kwasu.edu.ng

ABSTRACT: Number theory has turned out to be one of the useful when it comes to computer security. For instance, number theory helps to protect sensitive data such as credit card numbers when you shop online, when online transactions is performed. The prime application of it's is in the RSA cryptosystem. Rivest Shamir Adleman (RSA) algorithm is known to be a slower algorithm. The modular arithmetic in RSA is computationally expensive. In view of this, it has become a major challenge to implement RSA decryption in a faster manner. In this paper, we proposed an efficient method to implement RSA decryption based on Chinese Remainder Theorem and Strong prime. Three different operations, primitive traditional method, Chinese Remainder Theorem method and Chinese Remainder Theorem and strong prime criterion were used for comparisons. Our proposal achieves about 60% computational cost reduction of traditional method using Chinese Remainder method. More interesting, if the method based on Chinese Remainder Theorem and strong prime is implemented, about 84% computational cost can be reduced. Also, comparing to the Chinese Remainder method, the method based on Chinese Remainder Theorem and strong prime of RSA criterion takes about 37% of computational cost, almost 3.2 times faster than the Chinese Remainder Theorem based method. Theoretically, it was observed that our scheme is faster and it is also cheaper.

KEYWORDS: RSA, RSA-CRT, Number theory, computational cost, CRT-strong prime criterion.

I. INTRODUCTION

Cryptography provides techniques for keeping information secret, for determining that information has not been tampered with, and for determining who authored pieces of information [SG17].

The need for a secured communication is more profound than ever, recognizing the fact that the conduct of almost all our business and personal matters are carried out today by computer networks [SG16]. RSA [RSA78] is the most famous and widely deployed public-key cryptosystem. It is used for securing web traffic, some wireless devices and email. RSA cryptosystem encryption and decryption are computationally heavy and expensive because of modular exponentiation with very large numbers are needed. Since RSA is based on arithmetic modulo large numbers, it can be slow in constrained environments. For example, on a heavily loaded web server, RSA decryption significantly reduces the number of SSL requests per second that the server can handle.

The major bottleneck with the number theoretic RSA cryptosystem is that it require a lot of computational power for providing high and reasonable level of security with a low level efficiency most likely. Hence, how to make an efficient and faster implementation of RSA cryptosystem is great concerned.

In majority of RSA application, a small value of public key e is usually selected to speedup the encryption process. By this selection, the decryption takes more computational time because of the larger secret key d.

The alternative approach to overcome this shortcoming is the introduction of Chinese Remainder Theorem to RSA protocol, referred to as RSA-CRT [Lab11, SS16] which is a step towards efficiency. In RSA-CRT, the digital signature operation process is given as $S = M^d \bmod N$ is split in two operations $Sp = M^{dp} \bmod p$ and $Sq = M^{dq} \bmod q$, where $dp = d \bmod (p - 1)$ and $dq = d \bmod (q - 1)$ [SS16]. Chinese Remainder Theorem ensures that the combination of S_p and S_q produces signature S as $S = S_q + [(S_p - S_q).(q^{-1} \bmod p) \bmod p].q$ (1) denoted as $S = CRT(S_p - S_q)$ with a speedup of operations of approximately four times [SS16].

We propose a faster and efficient method to implement RSA decryption operation in this paper. This method is based on traditional RSA decryption method, RSA-CRT based method and CRT with strong prime criterion method. The proposed CRT with additional strong prime criterion takes 13882307 clock cycles. RSA-CRT decryption method takes 36597399 clock cycles and traditional RSA method takes 89290752 clock cycles.

II. CHINESE REMAINDER THEOREM

The complexity of RSA Cryptosystem depends heavily on the size of decryption exponent, d and the

modulus, n regarding the decryption process. This exponent indicates the number of modular multiplication that is needed to perform the exponentiation. An approach that has become a cornerstone technique to reduce the size of both d and n is an ancient Chinese Remainder Theorem [AB97].

Theorem 1:

Let $m_1, m_2, \ldots m_n$ be a pairwise relatively prime. That is gcd $(m_i, m_j) = 1$, for all i and j less than or equal to *n* where $i \neq j$. Then, the system of congruences

$X \equiv a_1 \pmod{m_1}$
$X \equiv a_2 \pmod{m_2}$
:
$X \equiv a_n \pmod{m_n}$

has a unique solution modulo the integer $m_1, m_2, \ldots, m_n$. Further, if $M_j = \prod^{n}_{i=1}/m_j$ and Z_j is a solution of $M_j z_j \equiv a_j \pmod{m_j}$ for each j, then the solution is stated by

$X = \sum_{j=1}^{n} M_j z_j$.

Now the special case of Chinese Remainder Theorem with two factors is shown here.

This theorem is mostly used in implementation of the RSA cryptosystem decryption operation.

Theorem 2 [Dav03]:

Let *p* and *q* be two numbers (co-prime positive integer) where gcd(p,q) =1. If $a \equiv b \pmod p$ and $a \equiv b \pmod q$, then we have $a \equiv b \pmod{pq}$.

III. EFFICIENT DECRYPTION METHOD

The prime factors p and q of modulus n in RSA cryptosystem has to be strong prime, in order to make the cryptosystem secure. It is reasonable that the private key party knows the prime factors of p-1, p+1, q-1 and q+1. The proposed decryption approach is specifically on the CRT with additional strong prime criterion. The private key (decryption key) holder performs the decryption procedure; $C^d \bmod n$ by our proposed method.

Step one: Factor *p*-1, *p*+1,*q*-1 and *q*+1 to get their prime factors. Let's assume that moduli *p*-1, *p* +1,*q*-1 and *q* +1 be expressed as follows;

$$p - 1 = \Pi_{i=1}^{h} r_i^{\propto i},$$
$$p + 1 = \Pi_{i=1}^{j} s_i^{\beta i},$$
$$q - 1 = \Pi_{i=1}^{k} t_i^{xi},$$
$$q + 1 = \Pi_{i=1}^{l} u_i^{\delta i},$$

Step two: compute the modular exponentiations with prime factors of p – 1 as the modulus. Then, apply the CRT to generate $y_1 = c^d \bmod (p - 1)$ based on these results. By the same technique, the private key (decryption key) holder generates $y_2 \bmod (p + 1)$, $y_3 = c^d \bmod (q - 1)$ and $y_4 = c^d \bmod (q + 1)$ individually. The explanations are as follows:

2.1 compute $c_{(p-1),i} = c \pmod{r_i^{\alpha i}}$, i = 1,…,h

2.2 compute $d_{(p-1),i} = d \pmod{\Phi(r_i^{\alpha i})}$, i = 1,…,h

2.3 compute the modular exponentiation

$M_{(p-1),i} = c_{(p-1),I}^{\,d(p-1)} \bmod r_i^{\alpha i}$, i = 1,…,h

2.4 Apply the CRT to generate $y_1 = c^d \bmod (p - 1)$ based on $M_{(p-1),i}$, i = 1,..,h

Step three: compute $X_1 = 2^{-1} (y_1 + y_2 - z) \bmod p$ where z = 0 if $y_1 \geq y_2$; otherwise z = 1 and $X_2 = 2^{-1} (y_3 + y_4 - z) \bmod q$ where z = 0 if $y_3 \geq y_4$; otherwise z = 1.

Step four: Apply the CRT to generate the plaintext $M = c^d \bmod n$ based on X_1 and X_2.

The numbers *p*-1, *p* +1,*q* -1 and *q* +1 can be individually decomposed into three prime factors at least at step 1. The bit lengths of $r_i^{\alpha i}$ (i = 1,…,h) are smaller than *p*-1. The bit lengths of $d_{(p-1),i}$ (i = 1,…,h) are smaller than d.

The complexity of modular exponentiation in RSA cryptosystem depends on the bit length of exponent and modulus. The totally computation time of step 2 to get $y_1 = c^d \bmod (p -1)$ is smaller than to compute y_1 by $c^d \bmod (p -1)$ directly. The efficient results are similar to $y_2 = c^d \bmod (p +1)$, $y_3 = c^d \bmod (q -1)$, and $y_4 = c^d \bmod (q +1)$.

The proposed decryption method using CRT with strong prime criterion is more efficient than the traditional method. The following theorem demonstrates that X_1 and X_2 generated in step 3 above are correct.

Theorem 3:

Given $y_1 = X \bmod (p-1)$, $y_2 = X \bmod (p +1)$ such that $0 \leq X \leq (p^2 - 1)/2$ and p is a prime. Then $X = X_1 = 2^{-1} (y_1 + y_2 - z) \bmod p$, where z = 0 if $y_1 \geq y_2$; otherwise z = 1 [Dav03].

IV. ANALYSIS OF COMPUTATIONAL COST

In this section, we will show that the decryption method that has been proposed in section III is more efficient than the traditional decryption method and the decryption method that is based on the Chinese Remainder theorem only. Nowadays, the bit length of modulus should be at least 2048 bits in order to make the operations secure. We will use this value in our calculation below. Before that, let's define some notations as follows:

- $MOD_E(y,z)$ denotes an operation of modular exponentiation, $x^y \bmod z$.
- M(w), A(w) and Mod(w) denote operations of multiplication, addition and modulus with the bit length of operand is w.
- l(w) denotes the bit length of w.
- S denotes the shift operator.

As was shown earlier, we implement the decryption operation by computing $C^d \pmod n$, where *d* is the decryption exponent. The encrypted message can be recovered by repeated modular exponentiation. The modulo operation $c^d \pmod n$ can be replace with other equation. According to [Dav03] the modulo operation $c^{d\,(}\text{mod } n)$ can be expressed as follows;

$MOD_E(d, n) = 1.5 \times l(d)\,[M(l(n) + 2Mod(l(n)) + 1]$

Before we compute the total number of operations for the three methods that we want to compare, three equations concerning the multiplication operation, the addition operation and also the modulus operation will be introduce. These equations are very important in order to do the calculation. The multiplication operations can be expressed as follows

$MOD_E(d, n) = 1.5 \times l(d)\,[M(l(n) + 2Mod(l(n)) + 1]$ (1)

$M(w) = 3M(w/2) + 5A(w) + 2s$ (2)

$A(w) = w/32$ (3)

$Mod(w) = Mod(w/2) + 4M(w/2) + 1.5A(w) + 3s$ (4)

In the first place, assume that all of Mod (32), M(32), A(32) and S take one clock cycle. For convenient, we find all the values of M(w) and Mod(w) first. Using equations (2) and (3), we get:

M(2048) = 3M (1024) + 5A (2048) + 2s
= 3M (1024) + 322
= 3[3M (512) + 5A (1024) + 2s] + 322
= 9M (512) + 808
= 9 [3M (256) + 5A (512) + 2s] + 808
= 27M (256) + 1546
= 27 [3M(128) + 5A (256) + 2s] + 1546
= 81M (128) + 2680
= 81M[3M(64)+5A(128)+2s] +2680
=243M (64)+ 4462
= 243[3M (32) + 5A(64) + 2s] +4462
= 729M(32) + 7378
= 8107 clock cycles.

M(1024) = 3M(512) + 5A (1024) +2s
= 3M(512) +162
= 3[3M(256) +5A (512)+2s]+162
= 9M (256) + 408
= 9[3M (128) +5A (256) +2s] +408
= 27M(128) + 786
= 27[3M (64) +5A (128) +2s] +786
=81M (64) + 1380
= 81M [3M(32)+5A(64) + 2s] + 1380
=243M(32) +2352
= 2595 clock cycles.

Using equation (2), (3) and (4)

Mod (64) = Mod (32) + 4M (32) + 1.5A (64) + 3s
= 11 clock cycles
Mod (128) = Mod (64) + 4M (64) + 1.5A (128) + 3s
= 80 clock cycles
Mod (256) = Mod (128) + 4M (128) + 1.5A (256) + 3s
= 363 clock cycles
Mod (512) = Mod (256) + 4M (256) + 1.5A (512) + 3s
= 1362 clock cycles
Mod (1024) = Mod (512) + 4M (512) + 1.5A (1024) +3s
= 4657 clock cycles
Mod (2048) = Mod (1024) + 4M (1024) +1.5A (2048) +3s
= 10479 clock cycles

The values for multiplication operation, addition operation and modulus operation are M(W), Mod (W) and A(W).

Table 1.

W(bits)	2048	1024	512	256	128	64	32
M(W)	8107	2595	811	243	67	15	1
Mod(W)	10479	4657	1362	363	80	11	1
A(W)	64	32	16	8	4	2	1

i. Primitive traditional method

$MOD_E(d,n) = 1.5\ X\ l(d)\ [\ M(l(n)) + 2 \bmod (l(n)) + 1]$
$= 1.5 \times 2048\ [\ M(2048) + 2 \bmod (2048) + 1]$
$= 1.5 \times 2048\ [\ 8107 + 2\ (10479) + 1]$
$= 1.5 \times 2048\ [\ 8107 + 20958 + 1]$
$= 3072\ [29066]$
$= 89290752$ clock cycles

So, the traditional RSA cryptosystem decryption method should take 89290752 clock cycles.

ii. Method on CRT-RSA

The bit length of two distinct primes is equal in this method, so, the total number of decryption operation for this method can be shown as

Total number = $2MOD_E(d/2, n/2) + A(1024) + 2M(1024) + \bmod(1024)$
$MOD_E = (d/n, n/2) = 1.5\ X\ l(d/2)\ [M(1(n/2)) + 2\bmod(l(n/2)) + 1]$
$= 1.5 \times 1024\ [M(1024) + 2\bmod(1024) + 1\ ]$
$= 1.5 \times 1024\ [2595 + 2\ (4657) + 1]$
$= 1536\ [2595 + 9315\ ]$
$= 18293760$ clock cycles
Total number = $2MOD_E(d/2, n/2) + A(1024) + 2M(1024) + \bmod(1024)$
$= 2(18293760) + 32 + 2(2595) + 4657$
$= 36597399$ clock cycles

The Method based on the Chinese Remainder Theorem should take 36597399 clock cycles.

iii. Method based on CRT with additional strong prime criterion

In this approach, p-1,p+1, q-1 and q+1 can be factored into atleast three numbers. Without loss of

generality, assuming that the bit length of the largest prime factor is about l(n)/4 and others are about l(n)/8. The total decryption operation for this method is;

Total number = 4 MOD_E (d/4,n/4) + 8 MOD_E (d/8, n/8) + 4[A(512) + 2M(512) + mod(512)] +4 [a(256)+ 2M (256) + mod (256)] + 2[A(1024) + M (1024) + mod (1024)] + A (1024) + 2 M(1024) + mod(1024).
MOD_E [d/4,n/4] = 1.5 x l (d/4) [m (l(n/4)] + 2 mod [l(n/4) +1].
= 1.5 x 512[m (512)[m(512) + 2 mod (512) + 1]
= 768(3536)
= 2715648 clock cycles

Therefore,

Total number = 4(2715648) + 8(372480) +4 (3000) + 4(857) +2 (7284) + 9879
= 13882307 clock cycles.

From the above calculation, the CRT method should take about 40% computational cost of the primitive traditional method. It means that 60% computational cost can be reduced if the decryption method based on CRT is implemented. In addition, if the method based on CRT with additional strong prime criterion is implemented, about 84% computational cost can be reduced from the traditional method. Therefore, the method based on CRT with additional strong primes takes only 15% computational cost. More interesting, comparing to the CRT based method, the method on CRT and strong prime take about 37% computational cost, almost 3.2 times faster than the Chinese Remainder Theorem based method.

Table 2: Evaluation of computational cost

Method	Cost reduction (%)	Clock cycles
TRSA	37	18293760
RSA-CRT	60	36597399
RSA and Strong Prime	84	13882307

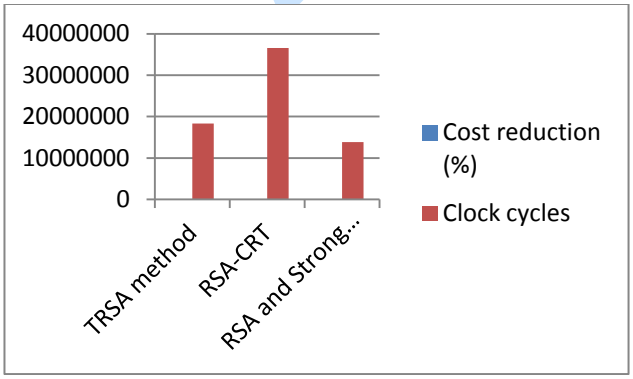

Figure 1. Comparison of Proposed Approach

V. CONCLUSIONS

In this paper, we introduced traditional RSA decryption, RSA-CRT, Chinese Remainder Theorem and strong prime for RSA decryption operation. The CRT and strong prime methods is faster and efficient than RSA-CRT and traditional RSA decryption method. The private key decryption d is very significant in RSA algorithm. The cipher-text can only be change to plaintext if the key d is known. Time spent to decrypt the message depends on the efficiency of algorithms to be used. We have improved the efficiency in order to reduce the time spent for decryption. The performance of our proposed method is evaluated on cost reduction and number of operation cycles.
We based our proposed approach on three methods, Primitive traditional method, RSA-CRT method and method based on CRT and Strong prime and comparison between them was evaluated. The method based on CRT and strong prime criterion is the best whereas RSA-CRT performs better than the traditional method.

REFERENCES

[AB97] **Anderson J. A., Bell J. M.** - *Number Theory with Applications*. New Jersey: Prentice-Hall. 1997.

[Dav03] **Davis T.** - *RSA Encryption* [Online]. [Accessed 10th, January 2017]. Available from URL: http://www.geometer.org /mathcircles. 2003.

[H+05] **Hwang R. J., Su F. F., Yeh Y. S., Chen C. Y.** - *An Efficient Decryption Method for RSA Cryptosystem*. AINA'05. 1, 585-590. 2005.

[Lab11] **Lab R.** - *High-Speed RSA Implementation* (2011). URL ftp://ftp.rsasecurity.com/ pub/pdfs/tr201.pdf

[RSA78] **Rivest R., Shamir A., Adleman L.** - *A Method for Obtaining Digital Signatures and Public Key Cryptosystems*. Commun. ACM, 21(2):120–126. Feb. 1978.

[SG16] **Saheed Y. K., Gbolagade K. A.** - *Efficient Image Encryption based on the moduli set {2^n - 1, 2^n, 2^n +1}*. Al-Hikmah Journal of Pure & Applied Sciences Vol.3. 2016: 15-21.

[SG17] **Saheed Y. K., Gbolagade K. A.** - *An Improved RSA algorithm based on Residue Number System*. In: Proceedings of 13th Nigeria Computer Society International Conference, 2017.Vol. 28. Pp 86-91.

[SS16] **Schinianakis D., Stouraitis T.** - *Residue Number systems in Cryptography: design, challenges, robustness*. In: Chang CH., Potkonjak M. (eds) Secure System Design and Trustable Computing. Springer, Cham, 2016.