A New Methodology to Find Private Key of RSA Based on Euler Totient Function

Kritsanapong Somsuk

Department of Computer and Communication Engineering, Faculty of Technology, Udon Thani Rajabhat University, UDRU, Udon Thani, Thailand.

E-mail: kritsanapong@udru.ac.th

ORCID ID: https://orcid.org/0000-0002-1311-8222

Received 11/11/2019, Accepted 26/10/2020, Published Online First 11/1/2021

This work is licensed under a Creative Commons Attribution 4.0 International License.

Abstract:

The aim of this paper is to present a new methodology to find the private key of RSA. A new initial value which is generated from a new equation is selected to speed up the process. In fact, after this value is found, brute force attack is chosen to discover the private key. In addition, for a proposed equation, the multiplier of Euler totient function to find both of the public key and the private key is assigned as 1. Then, it implies that an equation that estimates a new initial value is suitable for the small multiplier. The experimental results show that if all prime factors of the modulus are assigned larger than 3 and the multiplier is 1, the distance between an initial value and the private key is decreased about 66%. On the other hand, the distance is decreased less than 1% when the multiplier is larger than 66. Therefore, to avoid attacking by using the proposed method, the multiplier which is larger than 66 should be chosen. Furthermore, it is shown that if the public key equals 3, the multiplier always equals 2.

Keywords: Euler totient function, Private key, Public key, RSA

Introduction:

Nowadays, communication which is sent through opening a network such as internet and the machine is very popular because data is rapidly transmitted. However, opening a network is known as unsecure channel. With this problem, security and confidentiality of information becomes exceedingly important. Cryptography (1), which is one of security methods, is a technique to protect information by converting original message or plaintext as the unreadable message, or ciphertext. It is called the encryption process. In fact, ciphertext will be transmitted via the channel instead of plaintext. That means intruders cannot understand data which is trapped on the network. After ciphertext is arrived to receivers, they can use the decryption process to recover original plaintext.

The first generation of cryptography is called symmetric key cryptography (2). The secret key is selected for both of senders and receivers. Advanced Encryption Standard (AES) (1, 2) is the highest performance in this group. However, the problem is about how to exchange the secret key over unsecure channel. Later, this problem was solved when W. Diffie and M. E. Hellman proposed the new technique which is called asymmetric key cryptography or public key cryptography (3) in 1976. A pair of keys, the public key and the private key, is selected for the processes. Nevertheless, this algorithm can be chosen for only exchanging the secret key. In 1978, RSA (4) which is the best well known of public key cryptography was presented by R.L. Rivest, A. Shamir and L. Adleman. In fact, RSA can be chosen to solve many problems such as data encryption, digital signature and key exchange. In addition, if the modulus is a big number, then it becomes very difficult to recover both prime numbers by using mathematical techniques such as factoring. However, the difficulty to break this system is also based on the type of computer and updating method. Therefore, RSA is still very hard to be attacked.

Assuming all parameters of RSA are strong and at least 4096 bits of modulus is chosen (5), no one can break RSA in polynomial time. One the other hand, if one of parameters becomes a weakness, RSA may be broken by using some of disclosed algorithms. The examples of weak parameters which are already solved consist of...
prime numbers (6), small private key (7, 8), small public key with some disclosed parameters (9) and common modulus attack (10) etc.

In this paper, it is shown that if the multiplier, k, of Euler totient function is very small, time to break RSA can be decreased by using the proposed method. In fact, the proposed method which is suitable for small value of k is about estimating a new initial value, f, for the private key before finding the private key by using brute force attack. In addition, the distance is decreased about 66% when k equals 1. Furthermore, it is shown that when the public key equals 3 and prime factors are larger than 3, k always equals 2.

RSA Cryptosystem

RSA is the best well known public key cryptography. It can be applied with many tasks such as data encryption, digital signature and key exchange. For data encryption, it is divided into 3 processes as follows:

1) Key Generation: First, two secret prime numbers \( p \) and \( q \), are randomly generated to compute modulus, \( n = p \times q \) and Euler totient function, \( \Phi(n) = (p - 1)(q - 1) \). Next, the public key, \( e \), is randomly chosen with the following condition, \( 1 < e < \Phi(n) \) and gcd\( (e, \Phi(n)) = 1 \). The last process is to find the private key, \( d \), from \( ed \equiv 1 \mod \Phi(n) \) or \( ed = 1 + k \Phi(n) \), by using Extended Euclidean algorithm (11, 12). The published parameters are \( \{e, n\} \) and the secreted parameters are \( \{d, \Phi(n), p, q\} \).

2) Encryption process: Assuming \( m \) is represented as plaintext, the encryption equation is:

\[
c = m^e \mod n
\]

where \( c \) is ciphertext which will be sent to receiver.

3) Decryption process: After \( c \) is arrived, \( m \) is recovered by using the decryption equation:

\[
m = c^d \mod n
\]

However, for digital signature (13, 14), the process is different from data encryption. Assuming \( z \) is represented as a hash value of the signature and \( h \) is the signed text. Therefore, \( h \) is computed from the following equation:

\[
h = z^d \mod n
\]

In addition, the equation to verify the signed text is shown in the equation (4):

\[
z = h^e \mod n
\]

Exploit parameters to break RSA

In this section, the techniques to break RSA are presented. In fact, RSA is simply attacked when some parameters are weak.

Factoring

If \( n \) is factored, \( p \) and \( q \) are disclosed. After that, \( d \) can be easily recovered. In fact, it is very difficult to find \( p \) and \( q \) from at least 4096 bits of \( n \) when both of them are assigned as a strong parameter. On the other hand, RSA becomes an unsecured algorithm in the case that prime factors are a weak parameter because they are found by using some of factoring algorithms. The examples of factorization algorithms are as follows:

1) Trial division algorithm (TDA) (15, 16) is the simplest algorithm. It chooses 3 as the first divisor and it will be increased by 2 when the result has a remainder. Therefore, TDA is suitable for small value of \( q \). In addition, there is a different technique (17) to implement TDA by changing the sequence of divisor. The value of \( \left[ \sqrt{n} \right] \) is selected as the first divisor and it will be decreased by 2 when the result has a remainder. In fact, \( \left[ \sqrt{n} \right] \) is very close to \( p \) and \( q \) when \( p - q \) is small.

2) Fermat’s Factorization algorithm (FFA) (18, 19, 20, 21) was discovered by P. der Fermat in 1600. He found that \( n \) can be rewritten as the difference between two perfect square numbers which are mathematically relative with \( p \) and \( q \). In fact, FFA can find \( p \) and \( q \) very fast when the result of \( p - q \) is small.

3) Pollard’s \( p - 1 \) (22, 23) was proposed by J. Pollard in 1974. Fermat’s little theorem (24) is the main theorem of this algorithm. In fact, Pollard’s \( p - 1 \) has a very high performance when all prime factors of \( p - 1 \) or \( q - 1 \) are small.

4) Generalized Trial Division (25) was presented by M. Sahin. The technique behind this algorithm is to find the result of gcd\( (x, n) \), where \( x \in \mathbb{Z}^+ \), which does not equal 1, because it is one of two prime factors of \( n \). In fact, \( ip \) and \( jq \), where \( i, j \in \mathbb{Z}^+ \), \( 1 < i < q \) and \( 1 < j < p \), are all integers that gcd\( (ip, n) = p \) and gcd\( (jq, n) = q \).

Wiener’s attack

In 1990, M. Wiener (7, 8) showed that \( d \) will be recovered very simple by using continued fraction to find \( \frac{k}{d} \) which is a convergence of \( \frac{e}{n} \).
when it is a small value, $d < \frac{1}{3} n^2$. Moreover, in 1999, D. Boneh and G. Durfee (26) showed that if the following condition occurred, $\frac{1}{3} n^3 < d < n^{0.929}$, $d$ is simply recovered.

**Hastad Broadcasting Attack**

Hastad Broadcasting Attack (9) is the technique to find $d$ when $e = 3$. The condition for this method to finish the process is that the same message must be selected to be encrypted with $e = 3$ and the different values of modulus. For example, $c_1 = m^e \mod n_1$, $c_2 = m^e \mod n_2$ and $c_3 = m^e \mod n_3$. Then, $m$ can be recovered by using Chinese Remainder Theorem (CRT) (27, 28).

**Common Modulus Attack**

Common Modulus Attack (10) is the idea to find $m$ when it is encrypted two times with different public keys and common modulus, from $c_1 = m^e \mod n$ and $c_2 = m^e \mod n$, where $\gcd(e_1, e_2) = 1$.

**Partial Key Exposure attack**

Assuming $x$ is represented as bit length of $n$ and the $\frac{x}{4}$ least significant bits of $d$ is disclosed. The technique which is called Partial Key Exposure attack (10) can be chosen to recover $d$.

**Brute force Attack**

In fact, $d$ should be assigned in the following condition, $1 < d < \Phi(n)$, and it is always an odd number. Therefore, the concept of brute force attack is to find $d$ by choosing $d = 3$ as the first value to compute $t = ed \mod \Phi(n)$. If $t = 1$, then $d$ is the private key. On the other hand, $d$ must be increased by 2 when the result is not equal to 1 until the correct answer is found.

**The proposed method to attack RSA**

In this paper, a new method to recover $d$ in order to attack RSA is proposed. The key is to find an integer ($f$), where $3 < f < d$, to be a new initial value instead of 3 for brute force attack. In general, if brute force attack is selected to find $d$, then the initial value is begun as 3. On the other hand, if $f$ is chosen as an initial value, then the distance between the $f$ to $d$ decreased when it is compared with the distance between 3 and $d$. In fact, $f$ can be estimated by finding the smallest integer which is possible to be $\Phi(n)$.

**Lemma 1** The highest value of $p + q$ is $3 + \frac{n}{3}$.

**Proof:** Because $p > q$, then assigning $p = \sqrt{n} + a$ and $q = \sqrt{n} - b$, where $a, b \in \mathbb{N}$.

So, $p^aq = (\sqrt{n} + a)^a(\sqrt{n} - b) = n + a\sqrt{n} - b\sqrt{n} - ab$

Because $n = p^aq$, then $a\sqrt{n} - b\sqrt{n} - ab = 0$. That means it implies that $a$ is always larger than $b$.

Therefore, assuming $n$ is very close to $p^aq'$, where $p' = p + x, q' = q - y$ and $x, y \in \mathbb{N}^+$, the result of $p' + q'$ must be larger than $p + q$. The reason is that $x$ is always larger than $y$.

Because 3 is the smallest prime number which is an odd integer and the result of $3^e \frac{n}{3}$ equals $n$, the highest value of $p + q$ is $3 + \frac{n}{3}$.

**Theorem 1** Always $d$ equals or larger than $\left\lceil \frac{2n - 3}{3e} \right\rceil$.

**Proof:**

From $\Phi(n) = (p - 1)(q - 1) = pq - (p + q) + 1 = n - (p + q) + 1$

From Lemma 1, $\Phi(n) > n - (3 + \frac{n}{3}) + 1$

From $ed = 1 + k \Phi(n) = 1 + k(n - (p + q) + 1)$

Then, $ed > 1 + k(n - (3 + \frac{n}{3}) + 1)$

In fact, $k \in \mathbb{N}^+$, that means the minimum of $k$ is 1.

Assuming $k = 1$, then $ed > 1 + n - (3 + \frac{n}{3}) + 1$

Or $d > \frac{2n - 3}{3e}$

$d$ is always an integer, then $d \geq \left\lceil \frac{2n - 3}{3e} \right\rceil$

From Theorem 1, $f$ can be assigned as $\left\lceil \frac{2n - 3}{3e} \right\rceil$. Moreover, the convergence of $d$ equals this value whenever $k$ is small.

In addition, total loops which should be left out of the computation can be calculated by using the following equation:

$$s = \frac{f - 3}{2} \quad (5)$$
where, $s$ is the reduced loops and $3$ is the smallest prime number which is an odd number.

**Example 1:** Finding $f$ when $n = 174279334060020221413$ and $e = 212441$, $(p = 41964266303, \quad q = 4153041371, \quad d = 5332377947306261 \ (d \approx 5.33 \times 10^{16})$ and $k = 65$.

**Sol.** From 
\[
\begin{align*}
\frac{2n - 3}{3e} & = \frac{2 \times 174279334060020221413 - 3}{3 \times 212441} \\
& = 546910543194017 \approx 5.47 \times 10^{14}
\end{align*}
\]

From, 
\[
\begin{align*}
s = \frac{f - 3}{2} & = \frac{546910543194017 - 3}{2} \\
& = 273455271597007
\end{align*}
\]

$s \approx 2.73 \times 10^{14}$

The information in Fig. 1 is shown that the distance between $d$ and $f$ is smaller than the distance between $d$ and $3$.

![Figure 1. Some of parameters in example 1 on Numbers Line](image)

**Example 2:** Finding $f$ when $n = 174279334060020221413$ and $e = 212439$, $(d = 135050712179508995819 \ (d \approx 1.35 \times 10^{20})$ and $k = 164621$)(Fig.2)

**Sol.** From 
\[
\begin{align*}
\frac{2n - 3}{3e} & = \frac{2 \times 174279334060020221413 - 3}{3 \times 212439} \\
& = 546915692065394
\end{align*}
\]

$d$ should be an odd number, $f = f + 1 = 546915692065395$

Then, 
\[
f \approx 5.47 \times 10^{14}
\]

Then, it implies that,
\[
s = \frac{546915692065395 - 3}{2} = 273457846032696
\]
\[
s \approx 2.73 \times 10^{14}
\]

![Figure 2. Some of parameters in example 2 on Numbers Line](image)

From both Examples 1 and 2, it implies that the difference between $d$ and $f$ in example 2 is very high when it is compared with the result in example 1. The reason is that $k$ in example 2 is larger than the same parameter in example 1. In fact, the proposed technique is suitable for small value of $k$, because $k = 1$ is assigned in Theorem 1. Moreover, it is highly efficient with small value of $q$, because $q = 3$ is assigned in the same theorem.

In general, $k$ should be assigned as 1, because it is unknown value. However, in the next section, it is shown that if $e = 3$ and $q > 3$, then $k$ always equals 2. Therefore, $f$ can be expanded when the parameters are fallen in this case.

**Lemma 2** Assigning $k \in [1, 2]$, if

1. $e = 3$ and $\gcd(e, \Phi(n)) = 1$, then only $k = 1$ or 2 can be chosen to generate different values of $d$.
2. $e = 3, q = 3$ and $\gcd(e, \Phi(n)) = 1$, then $d$ can be generated from only $k = 1$.
3. $e = 3, q > 3$ and $\gcd(e, \Phi(n)) = 1$, then $d$ can be generated from only $k = 2$.

**Proof 1:** From \[ed = 1 + k \Phi(n)\]

Assuming $k = 3, \quad 3d = 1 + 3 \Phi(n)$

\[
d = \frac{1 + 3\Phi(n)}{3} = \frac{1}{3} + \Phi(n)
\]

That means, $d$ is always larger than $\Phi(n)$ when $k$ is higher than 2. However, it is impossible to occur. Therefore, assigning $e = 3$, the result which is fallen in range during 3 to $\Phi(n)$ must be generated from $k = 1$ or 2. In addition, it is shown in Table 1 that $k = 2, k = 5$ and $k = 8$ has the same value of $d$ when $e = 3$ and $n = 174279334060020221413$.

Assuming $k = 2, \quad 3d = 1 + 2 \Phi(n)$

\[
d = \frac{1 + 2\Phi(n)}{3}
\]

Therefore, it is possible to be occurred.

Assuming $k = 1, \quad 3d = 1 + \Phi(n)$

\[
d = \frac{1 + \Phi(n)}{3}
\]

Therefore, it is possible to occur.
Therefore, if \( e = 3 \) and \( \gcd(e, \Phi(n)) = 1 \), then only \( k = 1 \) or 2 can be chosen to generate different value of \( d \). \( \square \)

**Proof 2:**

From, \( ed = 1 + k \Phi(n) \)

Assuming \( k = 2 \), \( 3d = 1 + 2(3 - 1)^*(\frac{n}{3} - 1) \)

Because \( q = 3 \), then \( n = 3p = 1 + 4(p - 1) \)

\[ d = \frac{4p - 3}{3}, \quad 4p - 3 > 3p \]

It implies that \( d \) is larger than \( p \). Because \( n = 3p, \ d^*e = 3d > 9p > n \). Therefore, this case is impossible to occur. Therefore, \( k \) cannot be assigned as 2.

Next, assuming \( k = 1 \), \( 3d = 1 + (3 - 1)^*(\frac{n}{3} - 1) \)

Because \( q = 3 \), then \( n = 3p, \ 1 + 2(p - 1) \)

\[ d = \frac{2p - 1}{3}, \quad d < p < \Phi(n) \]

Then, it is possible to occur. Therefore, if \( e = 3, \ q = 3 \) and \( \gcd(e, \Phi(n)) = 1 \), then \( d \) is always generated from \( k = 1 \). \( \square \)

**Proof 3:**

From, \( ed = 1 + k \Phi(n) \)

Assuming \( k = 1 \), \( 3d = 1 + \Phi(n) \)

If there is a solution, then \((1 + \Phi(n)) \mod 3 = 0 \)

Because all odd prime numbers except 3 have the remainder as 1 or 2 when all of them are divided by 3, then there are 4 cases to consider \( d \).

**Case 1:** \( p \mod 3 = 1 \) and \( q \mod 3 = 1 \)

Then, \( (p - 1) \mod 3 = 0 \) and \((q - 1) \mod 3 = 0 \)

Therefore, \( \Phi(n) \mod 3 = 0 \)

Therefore, \((1 + \Phi(n)) \mod 3 = 1 \), the contradiction occurred, then there is no solution.

**Case 2:** \( p \mod 3 = 1 \) and \( q \mod 3 = 2 \)

Then, \((p - 1) \mod 3 = 0 \) and \((q - 1) \mod 3 = 1 \)

Therefore, \( \Phi(n) \mod 3 = 0 \)

Therefore, \((1 + \Phi(n)) \mod 3 = 1 \), the contradiction occurred, then there is no solution.

**Case 3:** \( p \mod 3 = 2 \) and \( q \mod 3 = 1 \)

Then, \((p - 1) \mod 3 = 1 \) and \((q - 1) \mod 3 = 1 \)

Therefore, \( \Phi(n) \mod 3 = 0 \)

Therefore, \((1 + \Phi(n)) \mod 3 = 1 \), the contradiction occurred, then there is no solution.

**Case 4:** \( p \mod 3 = 2 \) and \( q \mod 3 = 2 \)

Then, \((p - 1) \mod 3 = 1 \) and \((q - 1) \mod 3 = 1 \)

Therefore, \( \Phi(n) \mod 3 = 0 \)

Therefore, \((1 + \Phi(n)) \mod 3 = 1 \), the contradiction occurred, then there is no solution.

From all cases above, there is no result of \((1 + \Phi(n)) \mod 3 = 0 \). Therefore, it is impossible to find \( d \) from \( e = 3 \) and \( k = 1 \).

In fact, there is a solution in case 4 when \( e = 3 \) and \( k = 2 \) as follows:

From, \( \Phi(n) \mod 3 = 1 \)

Then, \( k \Phi(n) \mod 3 = 2 \Phi(n) \mod 3 = 2 \)

Therefore, \((1 + 2 \Phi(n)) \mod 3 = (1 + 2) \mod 3 = 0 \)

Therefore, if \( e = 3, \ q > 3 \) and \( \gcd(e, \Phi(n)) = 1 \), then \( d \) is always generated from \( k = 2 \). \( \square \)

**Theorem 2** Assigning \( e = 3 \) and \( q > 3 \), then \( d \geq \left[ \frac{4n - 9}{9} \right] \)

**Proof:**

From Lemma 1, \( \Phi(n) > n - (3 + \frac{n}{3}) + 1 \)

and \( ed > 1 + k(n - (3 + \frac{n}{3}) + 1) \)

From Lemma 2, \( 3d = 1 + 2 \Phi(n) \)

\[ 3d > 1 + 2(n - (3 + \frac{n}{3}) + 1) \]

\[ 9d > 3 + 6n - 18 + 2n + 6 \]

\[ 9d > 4n - 9 \]

\[ d > \frac{4n - 9}{9} \]

Therefore, if \( e = 3 \) and \( q > 3 \), then \( d \geq \left[ \frac{4n - 9}{9} \right] \) \( \square \)

In fact, assuming \( e = 3, \ q > 3 \) and \( f = \left[ \frac{4n - 9}{9} \right] \), then it implies that \( f' \approx 2f \).

**Example 3:** Finding \( f' \) when \( n = 174279334060020221413 \) and \( e = 3, \ (d = 116186222675935275827, \ d \approx 1.16 \times 10^{36}) \) and \( k = 2 \)

**Sol.Start,**

\[ f = \left[ \frac{4n - 9}{9} \right] = \]

\[ \frac{3 \times 174279334060020221413 - 9}{9} = 77457481804453431738 \]

\[ d \text{ should be an odd number, } f' = f + 1 = \]

\[ 77457481804453431739 \]

\[ f' \approx 7.75 \times 10^{39} \]
Then, it implies that,
\[ s = \frac{77457481804453431739 - 3}{2} \]
\[ = 38,728,740,902,226,715,868 \]
\[ s \approx 3.87 \times 10^{19} \]
In addition,
\[ f = \frac{2n - 3}{3e} = \left[ \begin{array}{c} 2 \times 1742793340600221413 - 3 \\ 9 \end{array} \right] \]
\[ = 38,728,740,902,226,715,870 \]
d should be an odd number, \( f = f + 1 = 38,728,740,902,226,715,871 \)
\[ f \approx 3.87 \times 10^{19} \]
Therefore, it implies that \( f \approx 2f \).

![Figure 3. Some of parameters in example 3 on Number Line](Image)

From Fig. 3, \( f' \) which is about \( 2f \) is chosen to be a new initial value of \( d \). In fact, \( f' \) can be estimated as \( 2f \) when \( e = 3 \) and \( q > 3 \) are selected as parameters of RSA algorithm.

In addition, if RSA is chosen for digital signature, both \( z \) and \( h \) must be disclosed. Moreover, \( h \) can be also recovered by using the following equation:
\[ t = h \cdot z^{(d-1)} \mod n \]  
(6)
In fact, \( z \) is always recovered by using equation (6), because
\[ t = h \cdot z^{(d-1)} \mod n = z^{(d-1)} \mod n = z \]
However, it is not necessary to compute \( d - 1 \) times of the multiplication. The process to find \( d \) is improved as follows:
First, the process to find \( u \),
\[ u = c \cdot z^{f} \mod n \]  
(7)
And find \( t \) from,
\[ t = u \cdot z^{(1)^{d-1}} \mod n \]  
(8)
Therefore, it requires only \( d \cdot f - 1 \) times of modular multiplication and \( 1 \) times of modular exponentiation. In addition, loops of modular multiplication decreased as \( \frac{d - f - 1}{2} \).

Assuming RSA is applied with digital signature, then both \( z \) and \( h \) can be found. On the other hand, only \( c \) will be known when RSA is chosen for data encryption. Because both of original plaintext and ciphertext must be chosen for the proposed algorithm, the proposed method can be chosen to find \( d \) when RSA is applied with digital signature.

**Algorithm:** Finding \( d 

**Input:** \( z, h, e, n \)

1. \( f \leftarrow \left[ \frac{2n - 3}{3e} \right] \)
2. IF \( f \) is an even number then
3. \( f \leftarrow f + 1 \)
4. End IF
5. \( z \leftarrow h^{s} \mod n \)
6. \( u \leftarrow z^{t} \mod n \)
7. \( t \leftarrow h \cdot u \mod n \)
8. \( i \leftarrow 0 \)
9. While \( t \neq z \) do
10. \( t \leftarrow t \cdot u \mod n \)
11. \( i \leftarrow i + 1 \)
12. End While
13. \( d \leftarrow f + i - 1 \)

**Output:** \( d \)

**Experimental Results and Analysis**

The aim of this section is to consider the performance of \( f \) which is generated from different values of \( k \). The experiment is divided into 4 tables: Table 1 is the considering \( f \) from \( n = 1742793340600221413 \), which is the modulus from examples 1 to 3; Table 2 is to consider \( n = 1081304974177129789 \) (\( p = 8514213993520707 \), \( q = 127 \) and \( \Phi (n) = 10727907623183608956 \)). The difference between Table 1 and Table 2 is the aspect of \( p - q \). The result of \( p - q \) is small in Table 1 but it is high in Table 2; Table 3 is to consider 1024 bits of \( n \):

Table 3 is to consider 1024 bits of \( n \):

\[
\begin{align*}
77457481804453431739 - 3 & = 38,728,740,902,226,715,868 \\
3.87 \times 10^{19} & = 38,728,740,902,226,715,870 \\
2n - 3 & = 38,728,740,902,226,715,871 \\
3e & \approx 3.87 \times 10^{19} \\
2n - 3 & \approx 3.87 \times 10^{19} \\
\end{align*}
\]

chosen to find \( d \) when RSA is applied with digital signature.
## Table 1. The pair of \((e, d)\) generated from \(n = 174279334060220221413\) which small result of \(p - q\)

<table>
<thead>
<tr>
<th>Row</th>
<th>(k)</th>
<th>Public Key ((e))</th>
<th>Private Key ((d))</th>
<th>The new initial value ((f))</th>
<th>(d - f)</th>
<th>Decreased Distance ((%))</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>11</td>
<td>1584357819445719431</td>
<td>1056238838824567964</td>
<td>528119193702054176</td>
<td>66%</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>253</td>
<td>688851122584596497</td>
<td>459234081844558553</td>
<td>229671047004100934</td>
<td>66%</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>16821112276539527827</td>
<td>3872874092226715869</td>
<td>757447877377059958</td>
<td>33%</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>148691</td>
<td>2344181342706297</td>
<td>78139378107676</td>
<td>1562787561599534</td>
<td>33%</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>74916809</td>
<td>46797758201809</td>
<td>1565914901017</td>
<td>3131388807933</td>
<td>33%</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>253</td>
<td>253</td>
<td>1608512056495977977</td>
<td>11128504591597797</td>
<td>3931865072693546</td>
<td>22%</td>
</tr>
<tr>
<td>7</td>
<td>20107</td>
<td>2600278210258708</td>
<td>5778396712919985</td>
<td>202238848838818</td>
<td>16%</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>201</td>
<td>16092151842134769861</td>
<td>151058364606009481</td>
<td>571973920147563890</td>
<td>16%</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>9049</td>
<td>63669498220495499</td>
<td>10611583050171501</td>
<td>5307591583973484</td>
<td>16%</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>3</td>
<td>1161622276539527827</td>
<td>3872874092226715869</td>
<td>757447877377059958</td>
<td>33%</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>19</td>
<td>4586296823532760897</td>
<td>116105435293166666</td>
<td>3979471822854443213</td>
<td>14%</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>22</td>
<td>9456394970479507297</td>
<td>671596634043939197</td>
<td>6308138555729681</td>
<td>16%</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>173</td>
<td>70515356400100117987</td>
<td>405125319557097610</td>
<td>288529521955350</td>
<td>16%</td>
<td></td>
</tr>
</tbody>
</table>

## Table 2. The pair of \((e, d)\) generated from \(n = 10813049747171729789\) which high result of \(p - q\)

<table>
<thead>
<tr>
<th>Row</th>
<th>(k)</th>
<th>Public Key ((e))</th>
<th>Private Key ((d))</th>
<th>The new initial value ((f))</th>
<th>(d - f)</th>
<th>Decreased Distance ((%))</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>11</td>
<td>1584357819445719431</td>
<td>1056238838824567964</td>
<td>528119193702054176</td>
<td>66%</td>
</tr>
<tr>
<td>2</td>
<td>253</td>
<td>68851122584596497</td>
<td>459234081844558553</td>
<td>229671047004100934</td>
<td>66%</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>33</td>
<td>16821112276539527827</td>
<td>3872874092226715869</td>
<td>757447877377059958</td>
<td>33%</td>
</tr>
<tr>
<td>4</td>
<td>148691</td>
<td>2344181342706297</td>
<td>78139378107676</td>
<td>1562787561599534</td>
<td>33%</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>74916809</td>
<td>46797758201809</td>
<td>1565914901017</td>
<td>3131388807933</td>
<td>33%</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>9049</td>
<td>63669498220495499</td>
<td>10611583050171501</td>
<td>5307591583973484</td>
<td>16%</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>3</td>
<td>1161622276539527827</td>
<td>3872874092226715869</td>
<td>757447877377059958</td>
<td>33%</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>19</td>
<td>4586296823532760897</td>
<td>116105435293166666</td>
<td>3979471822854443213</td>
<td>14%</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>22</td>
<td>9456394970479507297</td>
<td>671596634043939197</td>
<td>6308138555729681</td>
<td>16%</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>173</td>
<td>70515356400100117987</td>
<td>405125319557097610</td>
<td>288529521955350</td>
<td>16%</td>
<td></td>
</tr>
</tbody>
</table>

ISSN: 2078-8665
Published Online First: January 2021
Baghdad Science Journal
P-ISSN: 2078-8665
E-ISSN: 2411-7986

344
Table 3. The pair of \((e, d)\) generated from 1024 bits of \(n\)

<table>
<thead>
<tr>
<th>Row</th>
<th>(k)</th>
<th>Public Key ((e))</th>
<th>Private Key ((d))</th>
<th>The new initial value ((f))</th>
<th>(d \cdot f)</th>
<th>Decreased Distance ((%))</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>7</td>
<td>6.82 \times 10^{36}</td>
<td>4.55 \times 10^{36}</td>
<td>2.27 \times 10^{36}</td>
<td>66%</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>2275679</td>
<td>2.09 \times 10^{31}</td>
<td>1.39 \times 10^{31}</td>
<td>6.99 \times 10^{30}</td>
<td>66%</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>2.09 \times 10^{30}</td>
<td>2275679</td>
<td>1517119</td>
<td>75860</td>
<td>66%</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>2623</td>
<td>3.64 \times 10^{34}</td>
<td>1.21 \times 10^{34}</td>
<td>2.42 \times 10^{34}</td>
<td>33%</td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>1757281</td>
<td>5.43 \times 10^{31}</td>
<td>1.81 \times 10^{31}</td>
<td>3.62 \times 10^{31}</td>
<td>33%</td>
</tr>
<tr>
<td>6</td>
<td>2</td>
<td>2492887</td>
<td>3.83 \times 10^{31}</td>
<td>1.27 \times 10^{31}</td>
<td>2.55 \times 10^{31}</td>
<td>33%</td>
</tr>
<tr>
<td>7</td>
<td>3</td>
<td>457</td>
<td>3.14 \times 10^{30}</td>
<td>6.96 \times 10^{29}</td>
<td>2.43 \times 10^{29}</td>
<td>23%</td>
</tr>
<tr>
<td>8</td>
<td>4</td>
<td>709</td>
<td>2.69 \times 10^{30}</td>
<td>4.49 \times 10^{30}</td>
<td>2.69 \times 10^{30}</td>
<td>16%</td>
</tr>
<tr>
<td>9</td>
<td>4</td>
<td>3545</td>
<td>5.39 \times 10^{34}</td>
<td>8.98 \times 10^{34}</td>
<td>4.49 \times 10^{34}</td>
<td>16%</td>
</tr>
<tr>
<td>10</td>
<td>5</td>
<td>23</td>
<td>1.04 \times 10^{31}</td>
<td>1.38 \times 10^{31}</td>
<td>9.00 \times 10^{30}</td>
<td>14%</td>
</tr>
<tr>
<td>11</td>
<td>8</td>
<td>203</td>
<td>1.88 \times 10^{39}</td>
<td>1.56 \times 10^{39}</td>
<td>1.72 \times 10^{39}</td>
<td>9%</td>
</tr>
<tr>
<td>12</td>
<td>8</td>
<td>2291</td>
<td>1.67 \times 10^{35}</td>
<td>1.39 \times 10^{35}</td>
<td>1.52 \times 10^{35}</td>
<td>9%</td>
</tr>
<tr>
<td>13</td>
<td>8</td>
<td>43529</td>
<td>8.78 \times 10^{30}</td>
<td>7.31 \times 10^{31}</td>
<td>8.04 \times 10^{30}</td>
<td>9%</td>
</tr>
<tr>
<td>14</td>
<td>9</td>
<td>25</td>
<td>1.25 \times 10^{33}</td>
<td>1.04 \times 10^{32}</td>
<td>1.14 \times 10^{32}</td>
<td>9%</td>
</tr>
<tr>
<td>15</td>
<td>9</td>
<td>6895519</td>
<td>6.24 \times 10^{31}</td>
<td>4.61 \times 10^{30}</td>
<td>5.77 \times 10^{30}</td>
<td>8%</td>
</tr>
<tr>
<td>16</td>
<td>9</td>
<td>34477595</td>
<td>1.24 \times 10^{31}</td>
<td>2.92 \times 10^{29}</td>
<td>1.15 \times 10^{30}</td>
<td>8%</td>
</tr>
<tr>
<td>17</td>
<td>13</td>
<td>103</td>
<td>6.03 \times 10^{36}</td>
<td>3.09 \times 10^{39}</td>
<td>5.72 \times 10^{39}</td>
<td>6%</td>
</tr>
<tr>
<td>18</td>
<td>13</td>
<td>18121</td>
<td>3.34 \times 10^{34}</td>
<td>1.75 \times 10^{35}</td>
<td>3.25 \times 10^{34}</td>
<td>6%</td>
</tr>
<tr>
<td>19</td>
<td>16</td>
<td>37</td>
<td>2.07 \times 10^{37}</td>
<td>8.61 \times 10^{30}</td>
<td>1.97 \times 10^{30}</td>
<td>5%</td>
</tr>
<tr>
<td>20</td>
<td>16</td>
<td>1155547</td>
<td>6.62 \times 10^{32}</td>
<td>2.73 \times 10^{35}</td>
<td>6.34 \times 10^{35}</td>
<td>5%</td>
</tr>
<tr>
<td>21</td>
<td>22</td>
<td>353</td>
<td>2.98 \times 10^{32}</td>
<td>9.02 \times 10^{30}</td>
<td>2.89 \times 10^{30}</td>
<td>4%</td>
</tr>
<tr>
<td>22</td>
<td>23</td>
<td>2471</td>
<td>4.25 \times 10^{30}</td>
<td>1.29 \times 10^{29}</td>
<td>4.12 \times 10^{29}</td>
<td>4%</td>
</tr>
<tr>
<td>23</td>
<td>25</td>
<td>22399</td>
<td>5.33 \times 10^{34}</td>
<td>1.42 \times 10^{31}</td>
<td>5.19 \times 10^{31}</td>
<td>3%</td>
</tr>
<tr>
<td>24</td>
<td>25</td>
<td>393947</td>
<td>5.69 \times 10^{35}</td>
<td>8.08 \times 10^{30}</td>
<td>5.61 \times 10^{30}</td>
<td>2%</td>
</tr>
<tr>
<td>25</td>
<td>27</td>
<td>10743361</td>
<td>2.09 \times 10^{32}</td>
<td>2.96 \times 10^{36}</td>
<td>2.06 \times 10^{36}</td>
<td>2%</td>
</tr>
<tr>
<td>26</td>
<td>47</td>
<td>32694499</td>
<td>9.64 \times 10^{32}</td>
<td>9.74 \times 10^{36}</td>
<td>9.54 \times 10^{36}</td>
<td>1%</td>
</tr>
<tr>
<td>27</td>
<td>66</td>
<td>4971403</td>
<td>6.44 \times 10^{32}</td>
<td>6.41 \times 10^{31}</td>
<td>6.37 \times 10^{31}</td>
<td>&lt;1%</td>
</tr>
<tr>
<td>28</td>
<td>67</td>
<td>761</td>
<td>7.72 \times 10^{36}</td>
<td>4.18 \times 10^{34}</td>
<td>7.72 \times 10^{34}</td>
<td>&lt;1%</td>
</tr>
<tr>
<td>29</td>
<td>100</td>
<td>44059</td>
<td>2.24 \times 10^{37}</td>
<td>7.22 \times 10^{32}</td>
<td>2.24 \times 10^{37}</td>
<td>&lt;1%</td>
</tr>
<tr>
<td>30</td>
<td>769682</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The results from Table 1, Table 2 and Table 3 imply that:

1) The ratio between the decreased distance, \(d - f\), and \(d\) is quite stable for the same value of \(k\) when \(d = 1 + k\Phi(n)\) < \(\Phi(n)\).

2) In fact, an exception is occurred when \(d > \Phi(n)\), because \(d\) must be decreased in the following condition, \(1 < d < \Phi(n)\), and it also decreases value of \(k\). Therefore, the ratio of this case is similar to the same pair of \((e, d)\) which is generated from a smaller value of \(k\). The example that the same pair of \((e, d)\) can be generated from the different values of \(k, k = 2, k = 5\) and \(k = 8\), and the ratio of them are 33% is shown in 4th row, 11th row and 16th row of Table 1.

3) The size of \(e\) does not affect the ratio; it is shown in the 2nd Row and the 3rd Row that although \(e\) is alternated with \(d\), the ratio is not changed. In fact, only size of \(k\) affects the ratio.

4) The maximum decreased distance is 66% for all the values of \(q > 3\) and \(k = 1\).

5) The ratio that is less than 1% is begun at \(k > 66\). Therefore, to avoid attacking by using the proposed method \(k > 66\).

6) From Theorem 2 and the results in Table 1, Table 2 and Table 3, it implies that \(d\) is rapidly recovered when \(e = 3\), because \(k\) is stable (\(k = 2\)) and the ratio is always 33%. Therefore, we can estimate 99% of the ratio by using the equation \(f = \left[ f \cdot 99 \right] \approx \left[ f \cdot 2.91 \right] \), 34 is from [33.xx] % to prevent \(f > d\). The example is as follows:

From 4th Row of Table 1, \(f = 38728740902226715869\), then \(f = \left[ 38728740902226715869 \cdot 2.91 \right] = 112700636025479743179 = 1.12 \times 10^{20}\)

In Table 4, \(q = 3\) \((n = 24473764731015097203, p = 8157921577005032401\) and \(\Phi(n) = 16315843154010064800\) is chosen for the experiment to consider the performance of \(f\). The reason is that \(q = 3\) is assigned with the main equation in Theorem 1.
<table>
<thead>
<tr>
<th>Row</th>
<th>k</th>
<th>Public Key (e)</th>
<th>Private Key (d)</th>
<th>The new initial value (f)</th>
<th>d - f</th>
<th>Decreased Distance (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>3779</td>
<td>4317502819267019</td>
<td>4317502819267019</td>
<td>0</td>
<td>100%</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>43019</td>
<td>3792768218777</td>
<td>439019</td>
<td>0</td>
<td>100%</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>3792768218777</td>
<td>43019</td>
<td>0</td>
<td>100%</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>761671</td>
<td>42842322811831</td>
<td>21421116405915</td>
<td>0</td>
<td>50%</td>
</tr>
<tr>
<td>5</td>
<td>3</td>
<td>64849</td>
<td>754792355053249</td>
<td>25197451834417</td>
<td>50319490368833</td>
<td>33%</td>
</tr>
<tr>
<td>6</td>
<td>3</td>
<td>453943</td>
<td>10782479357607</td>
<td>3594249319203</td>
<td>71884986238404</td>
<td>33%</td>
</tr>
<tr>
<td>7</td>
<td>3</td>
<td>104957</td>
<td>447026958529679</td>
<td>14900898532097</td>
<td>29901797064196</td>
<td>33%</td>
</tr>
<tr>
<td>8</td>
<td>4</td>
<td>4049</td>
<td>1611839281649924</td>
<td>4029598210424813</td>
<td>1208879463127436</td>
<td>25%</td>
</tr>
<tr>
<td>9</td>
<td>4</td>
<td>28225579</td>
<td>2312206690819</td>
<td>578051672705</td>
<td>173415018114</td>
<td>25%</td>
</tr>
<tr>
<td>10</td>
<td>5</td>
<td>221</td>
<td>3691367229414041</td>
<td>73827344588280837</td>
<td>29530937835132344</td>
<td>20%</td>
</tr>
<tr>
<td>11</td>
<td>5</td>
<td>182579</td>
<td>4468159852450949</td>
<td>89363197049003</td>
<td>357452788196016</td>
<td>20%</td>
</tr>
<tr>
<td>12</td>
<td>7</td>
<td>47</td>
<td>243001919310435183</td>
<td>34715599021490741</td>
<td>20828735941289444</td>
<td>15%</td>
</tr>
<tr>
<td>13</td>
<td>7</td>
<td>23226801</td>
<td>4921441006801</td>
<td>70306300971</td>
<td>4218378005830</td>
<td>15%</td>
</tr>
<tr>
<td>14</td>
<td>8</td>
<td>907</td>
<td>143910413706814243</td>
<td>17988801713351781</td>
<td>125921611993462462</td>
<td>13%</td>
</tr>
<tr>
<td>15</td>
<td>8</td>
<td>7841</td>
<td>166466696241816161</td>
<td>20808370302270201</td>
<td>14565859211589140</td>
<td>13%</td>
</tr>
<tr>
<td>16</td>
<td>9</td>
<td>28211</td>
<td>5208153606259691</td>
<td>587350400695121</td>
<td>462680320369070</td>
<td>12%</td>
</tr>
<tr>
<td>17</td>
<td>11</td>
<td>12572633</td>
<td>14274955117897</td>
<td>1297726828899</td>
<td>1297726828899</td>
<td>10%</td>
</tr>
<tr>
<td>18</td>
<td>15</td>
<td>101</td>
<td>242314520227781901</td>
<td>16154300152458217</td>
<td>226160202134792797</td>
<td>7%</td>
</tr>
<tr>
<td>19</td>
<td>15</td>
<td>437</td>
<td>560040382860757337</td>
<td>37336025524050491</td>
<td>52270435736706882</td>
<td>7%</td>
</tr>
<tr>
<td>20</td>
<td>15</td>
<td>44137</td>
<td>5544954285750073</td>
<td>369663619050005</td>
<td>517529066670068</td>
<td>7%</td>
</tr>
<tr>
<td>21</td>
<td>16</td>
<td>1523</td>
<td>171404713305424187</td>
<td>1071296331389011</td>
<td>16069449973835176</td>
<td>7%</td>
</tr>
<tr>
<td>22</td>
<td>16</td>
<td>4124977</td>
<td>22857369423313</td>
<td>1428585589857</td>
<td>2142878384356</td>
<td>7%</td>
</tr>
<tr>
<td>23</td>
<td>11</td>
<td>108613</td>
<td>3483989654448877</td>
<td>152010994292949</td>
<td>315461967015928</td>
<td>5%</td>
</tr>
<tr>
<td>24</td>
<td>22</td>
<td>6300011</td>
<td>569758607386981</td>
<td>2589811851759</td>
<td>5438604886932</td>
<td>5%</td>
</tr>
<tr>
<td>25</td>
<td>45</td>
<td>696359</td>
<td>1054359808562039</td>
<td>23430217968045</td>
<td>103092950953994</td>
<td>3%</td>
</tr>
<tr>
<td>26</td>
<td>45</td>
<td>4874513</td>
<td>1504622829794577</td>
<td>3347173995435</td>
<td>147275655799143</td>
<td>3%</td>
</tr>
<tr>
<td>27</td>
<td>66</td>
<td>263</td>
<td>4094470145112791927</td>
<td>6203742644102909</td>
<td>403243271867168902</td>
<td>2%</td>
</tr>
<tr>
<td>28</td>
<td>100</td>
<td>329</td>
<td>5008815071505794969</td>
<td>495922284316413515</td>
<td>495922284316413434</td>
<td>1%</td>
</tr>
<tr>
<td>29</td>
<td>102</td>
<td>5471</td>
<td>304188631725640031</td>
<td>2982241483094511</td>
<td>30120638972545520</td>
<td>&lt;1%</td>
</tr>
<tr>
<td>30</td>
<td>769682</td>
<td>769682</td>
<td>12894267308382733921</td>
<td>249932494202143</td>
<td>12894017357888531177</td>
<td>&lt;1%</td>
</tr>
</tbody>
</table>

The results in Table 4 imply that,  
1) \( f = d \) when \( k = 1 \), because \( q = 3 \) is chosen in the experiment.  
2) For the same value of \( k \), the ratio considered from \( q = 3 \) is always larger than the others which are generated from \( q > 3 \).

In addition, the proposed method will be compared with some other algorithms. In general, the prominent point of each method is different from each other. However, \( k = 1 \) and a small private key are assigned for all values in Fig. 1 to ensure that the proposed method is efficient when \( k \) is small. There are 4 compared methods as follows:  
1) Brute force attack that the initial value is 3, this method is efficient when \( d \) is small.  
2) The improvement of FFA in (21), this method is efficient when the result of \( p - q \) is close to 0.  
3) The improvement of TDA in (17), this method is efficient when \( q \) is close to \( \sqrt{n} \).  
4) Pollard’s \( p - 1 \), this method is efficient when all prime factors of \( p - 1 \) or \( q - 1 \) are small.

---

**Figure 4. Logarithm of total loops from each algorithm**

---

**Table 4. The pair of (e, d) generated from \( n = 24473764731015097203 \) with \( q = 3 \) (\( p = 815792157700532401 \))**

The new initial value (\( f \)) is close to \( d \) when \( k = 1 \), because for prime factors of \( p = 3 \), this method is efficient when all prime factors of \( p - 1 \) or \( q - 1 \) are small.
Assuming \( l \) is represented as total loops to finish the process of each algorithm, Fig. 4 shows the result of \( \log_{10}(l) \) from the proposed method and all compared methods. The experimental results show that the proposed method requires the smallest loops when \( k \) equals 1 and \( d \) is small. On the other hand, it cannot guarantee that the proposed method is the most efficient algorithm whenever \( k \) and \( d \) are not assigned in the conditions suitable for this method.

Therefore, the conclusion is that the proposed method is a special proposed method that is suitable for the small values of \( k \) and \( d \).

**Conclusion:**

In this paper, the new technique to recover the private key \((d)\) is proposed by estimating the new initial value, \( f \), before using brute force attack. In fact, \( f \) can be computed by choosing the smallest value which may be Euler totient function, \( \Phi(n) \), instead of the real value of \( \Phi(n) \) and selecting \( 1 \) instead of \( k \). The method is suitable for the small value of \( k \), especially \( k \) equals 1. In fact, assuming a prime factor \((q)\) is higher than \( 3 \) and \( k \) equals \( 1 \), the distance between \( f \) and \( d \) decreased about \( 66\% \). On the other hand, the decreased distance is less than \( 1\% \) when \( k \) is larger than 66. Therefore, to avoid attacking RSA by using the proposed technique, \( k \) should be assigned very large. Furthermore, \( d \) which is computed from \( \frac{1+k\Phi(n)}{e} \) must be less than \( \Phi(n) \). In addition, it is shown that \( k \) always equals 2 when \( e \) equals 3. However, if \( e \) and \( q \) equal 3, then \( k \) always equals 1.

**Author's declaration:**

- Conflicts of Interest: None.
- I hereby confirm that all the Figures and Tables in the manuscript are mine. Besides, the Figures and images, which are not mine, have been given the permission for re-publication attached with the manuscript.
- Ethical Clearance: The project was approved by the local ethical committee in Udon Thani Rajabhat University.

**References:**


347


Somsuk K, Chiawchanwattana T, Sanemueang C. Speed up RSA’s Decryption Process with Large sub Exponents using Improved CRT, Proceedings of International Conference on Information Technology. Thailand, 2018, pp. 1-5. IEEE.