ADV MATH SCI JOURNAL
Advances in Mathematics: Scientific Journal **9** (2020), no.12, 10869–10881
ISSN: 1857-8365 (printed); 1857-8438 (electronic)
https://doi.org/10.37418/amsj.9.12.71 Special Issue SMS-2020

AN ALGORITHM FOR REDUCING $\mathrm{LUC}_{4,6}$ AND $\mathrm{LUC}_{4,6}$ELG CRYPTOSYSTEMS COMPUTATIONAL TIME

TZE JIN WONG[1], IZZATUL NABILA SARBINI, LEE FENG KOO, FATIN HANA NANING, MOHAMED OTHMAN, AND MOHAMAD MAULANA MAGIMAN

ABSTRACT. A new algorithm modified from Doubling Rule and Lucas Chains is proposed. The algorithm omits several terms of Lucas sequences in order to increase the efficiency of $\mathrm{LUC}_{4,6}$ and $\mathrm{LUC}_{4,6}$ELG cryptosystem. Comparison with existing method show that the modified algorithm successfully reduces the number of terms in calculation. Thus, reduce the computational efforts significantly.

1. INTRODUCTION

The revolution of information and communication technologies (ICT) has promoted the emergence of the cloud-based system, the Internet of Things (IoT), Big data, Industry 4.0 and BYOD (Bring Your Own Device) Security is one of the essential concerns in the revolution of ICT. Network engineering and security become a vital component in information security, for it is responsible for securing all the data and information contained in their system against the abuse, misuse, unauthorized access, and theft. The development of network engineering and security procedures require the implementation of the cryptographic function. Consequently, modern and classical cryptography which employ various mathematical technique needs primary attention in order to ensure that security and privacy concerns be comprehensively addressed.

[1]*corresponding author*

2020 *Mathematics Subject Classification.* 94A60, 11T71.

Key words and phrases. algorithm, Lucas sequence, quartic polynomial, term of sequence.

Public key cryptography is an encryption technique utilizing a public key and a private key for secret writing. The public key is used to encrypt plaintexts while the ciphertexts can be decrypted by using the private key. This cryptography concept was introduced by Diffie and Hellman in 1976. The well known RSA is the pioneer of public-key encryption and signature scheme (or called as public key cryptosystem) and is widely employed for secure date transmission. The RSA scheme is based on a hard mathematical problem, i.e. the intractability of factoring large integers. In [1, 2] the authors improved the various of RSA scheme with high security measure.

LUC based cryptosystem is a modification of RSA scheme but based on Lucas sequence. LUC type cryptosystem implemented the Lucas function in encryption and decryption process. It was developed by Smith and Lennon in 1993 after invented the weakness of RSA. LUC cryptosystem comparable security level with classic system, but with lesser key sizes. Wong *et al.* [3] advocated the fourth and sixth order of Lucas sequences to develop their $\text{LUC}_{4,6}$ cryptosystem by utilizing the characteristics of quartic polynomials. Furthermore, the fourth and sixth order Lucas sequence also employed to develop the $\text{LUC}_{4,6}$ELG cryptosystem.

Most of the current research focuses on security aspect for quartic Lucas based cryptosystem [4–6] and seldom focus on efficiency aspect for quartic Lucas based cryptosystem. Therefore, this paper focuses on the efficiency of $\text{LUC}_{4,6}$ and $\text{LUC}_{4,6}$ELG cryptosystems that related to computational time. The computational time for a Lucas based cryptosystem greatly depends on the number of terms in the Lucas sequences. Knuth [7] introduced a "Doubling rule" to omit some terms of the sequences during calculation. However, the "Doubling rule" was not customized to compute and rectify the fourth order of the Lucas sequence because the "Doubling rule" unable to fill the gaps in between the fourth order Lucas sequence, i.e. V_{2a-1}, V_{2a-3}, V_{2a+1}, V_{2a+3}, V_{a-1} and V_{a+1}. These gaps are important during the final step of computing.

Montgomery [8] was credited for being the first person to consider the Lucas chains for developing the gaps filling rules in the early eighties. Similar work had been done by Yen and Laih [9] who proposed an improved algorithm to compute the Lucas chain. The sequences of V_{a-b}, V_a and V_b were used to generate their special type of additional sequences. For fourth order Lucas sequence, V_{a-2b} and V_{a-3b} or higher sequences are necessary to generate specialty sequences. However, Lucas chains still unable to fill these gaps. Therefore, the "Doubling rule" and Lucas

chains are deemed as not suitable to compute the fourth order of Lucas sequences or higher. In this paper, a new algorithm to reduce the computational time LUC$_{4,6}$ and LUC$_{4,6}$ELG cryptosystems compare to classical method is proposed. The suggested algorithm increase the efficiency of LUC$_{4,6}$ and LUC$_{4,6}$ELG cryptosystems by omitting some terms in the Lucas sequences during the calculation.

2. PRELIMINARIES

Here, we recall some related concepts which will be useful in our study. Details can be found in [3, 4, 6].

An N-th order linear recurrence of Lucas sequence is a sequence of integers, T_k defined by

$$T_k = \sum_{i=1}^{N} (-1)^{i+1} a_i T_{k-i},$$

with initial values, $T_0, T_1, \ldots, T_{N-1}$ and a_i are coefficients in N-th order polynomial,

$$x^N + \sum_{i=1}^{N-1} (-1)^i a_i x^{N-i} + a_N = 0.$$

In the LUC$_{4,6}$ and LUC$_{4,6}$ELG cryptosystems, the fourth and sixth order of Lucas sequences are selected for encryption and decryption processes. The fourth order Lucas sequence is applied to create the first and third plaintext or ciphertext, whilst the sixth order Lucas sequence is used to create the second plaintext or ciphertext. Consequently, three plaintexts or ciphertext in each set in the system.

Let n be the product of two large secret primes, p and q. The encryption key, (e, n) is made public. (m_1, m_2, m_3) is set of plaintext. The prime number, e must be relatively prime to the Euler totient function $\phi(n) = \bar{p}\bar{q}$ because it is necessary to solve the congruence $ed \equiv 1 \mod \phi(n)$ to find the decryption key d. The Euler totient function can be defined as

$$\phi(n) = p_1^{b_1 - 1} \bar{p_1} p_2^{b_2 - 1} \bar{p_2} \ldots p_r^{b_r - 1} \bar{p_r},$$

where

$$\bar{p}_l = \begin{cases} p_i^3 + p_i^2 + p_i + 1, & \text{if } f(x) \text{ is of type of } t[4] \mod p_i \\ p_i^3 - 1, & \text{if } f(x) \text{ is of type of } t[3,1] \mod p_i \\ p_i^2 - 1, & \text{if } f(x) \text{ is of type of } t[2,1] \mod p_i \\ p_i + 1, & \text{if } f(x) \text{ is of type of } t[2] \mod p_i \\ p_i - 1, & \text{if } f(x) \text{ is of type of } t[1] \mod p_i \end{cases}$$

and $f(x) = x^4 - m_1x^3 + m_2x^2 - m_3x + 1$. As a matter of fact, the receiver receives the ciphertext, (c_1, c_2, c_3) but not the plaintext, (m_1, m_2, m_3). Therefore, it is necessary to make sure that the type of $g(x) = x^4 - c_1x^3 + c_2x^2 - c_3x + 1$ equivalent to the type of $f(x)$. In practice, since $\phi(n)$ depends on the type of auxiliary polynomial, the encryption key, e must be relatively prime to $p-1, q-1, p+1, q+1, p^2+p+1, q^2+q+1, p^3+p^2+p+1$, and q^3+q^2+q+1 to cover all possible cases.

With these preliminary evaluations, a public-key cryptosystem will be set based on the Lucas sequence V_k d erived from the quartic polynomial, $x^4 - m_1x^3 + m_2x^2 - m_3x + 1 = 0$.

3. The Cryptosystems

3.1. $\mathbf{LUC}_{4,6}$.

The encryption function is defined as

$$\begin{aligned} E(m_1, m_2, m_3) &= (c_1, c_2, c_3) \\ &= (V_e(m_1, m_2, m_3, 1), \\ &\quad V_e(m_2, m_1m_3 - 1, m_1^2 + m_3^2 - 2m_2, m_1m_3 - 1, m_2, 1), \\ &\quad V_e(m_3, m_2, m_1, 1)) \mod n, \end{aligned}$$

where $n = pq$ as above; (m_1, m_2, m_3) constitutes the plaintexts and the coefficients of quartic polynomial; (e, n) is the encryption key. $V_e(m_1, m_2, m_3, 1)$ and $V_e(m_3, m_2, m_1, 1)$ are the e-th term of the fourth order of Lucas sequence. Whilst, $V_e(m_2, m_1m_3 - 1, m_1^2 + m_3^2 - 2m_2, m_1m_3 - 1, m_2, 1)$ is e-th term in the sixth order of Lucas sequence.

The decryption key is (d, n) where d is the inverse of e modulo $\phi(n)$. In order to decipher the message, the receiver must be adequate to compute $\phi(n)$ and then calculate

$$\begin{aligned} D(c_1, c_2, c_3) &= (m_1, m_2, m_3) \\ &= (V_d(c_1, c_2, c_3, 1), \\ &\quad V_d(c_2, c_1c_3 - 1, c_1^2 + c_3^2 - 2c_2, c_1c_3 - 1, c_2, 1), \\ &\quad V_d(c_3, c_2, c_1, 1)) \mod n \end{aligned}$$

to recover the original message of (m_1, m_2, m_3).

3.2. **LUC$_{4,6}$ELG.** The encryption function is defined as

$$\begin{aligned} E(m_1, m_2, m_3) &= (c_1, c_2, c_3, c_4) \\ &= (V_{aQ}(m_1, m_2, m_3, 1), \\ &\quad V_{aQ}(m_2, m_1m_3 - 1, m_1^2 + m_3^2 - 2m_2, m_1m_3 - 1, m_2, 1), \\ &\quad V_{aQ}(m_3, m_2, m_1, 1), aR) \mod n, \end{aligned}$$

with $n = pq$ as above and $Q = bR$ is public key. a and b are the secret numbers chosen by sender and receiver, respectively. R is a shared secret number chosen by sender and receiver. (m_1, m_2, m_3) constitutes the plaintexts and the coefficients of quartic polynomial; $V_{aQ}(m_1, m_2, m_3, 1)$ and $V_{aQ}(m_3, m_2, m_1, 1)$ are the aQ-th term of the fourth order of Lucas sequence; $V_{aQ}(m_2, m_1m_3 - 1, m_1^2 + m_3^2 - 2m_2, m_1m_3 - 1, m_2, 1)$ is aQ-th term in the sixth order of Lucas sequence.

The decryption key is (d, n) where d is the inverse of e modulo $\phi(n)$ and $e = b \cdot c_4$. In order to decipher the message, the receiver must know or be able to compute $\phi(n)$, and subsequently evaluate

$$\begin{aligned} D(c_1, c_2, c_3, c_4) &= (m_1, m_2, m_3) \\ &= (V_d(c_1, c_2, c_3, 1), \\ &\quad V_d(c_2, c_1c_3 - 1, c_1^2 + c_3^2 - 2c_2, c_1c_3 - 1, c_2, 1), \\ &\quad V_d(c_3, c_2, c_1, 1)) \mod n \end{aligned}$$

to recover the original message of (m_1, m_2, m_3).

4. The Algorithm

4.1. **Computing Fourth Order Lucas Sequence.** The method to compute fourth order Lucas sequence has been discussed by Wong et al. [10]. Let define some

abbreviations as follow:

$$\begin{aligned} V_k(M_1) &= V_k(m_1, m_2, m_3, 1), \\ V_k(M_2) &= V_k(m_2, m_1m_3 - 1, m_1^2 + m_3^2 - 2m_2, m_1m_3 - 1, m_2, 1), \\ V_k(M_3) &= V_k(m_3, m_2, m_1, 1). \end{aligned} \tag{4.1}$$

The $(a + b)$-th term of fourth order of Lucas sequence can be written as

$$\begin{aligned} V_(a + b)(M_1) = V_a(M_1)V_b(M_1) - V_(a - b)(M_1)V_b(M_2) \\ + V_(a - 2b)(M_1)V_b(M_3) - V_(a - 3b)(M_1), \end{aligned}$$

where $m_1, m_2,$ and m_3 constitute the coefficients of a quartic polynomial.

Definition 4.1. *For $x \geq 1$, the three special values (which are used in following proposition) are given as*

$$\begin{aligned} V_{2^x}(M_1) &= V_{2^{x-1}}(M_1)^2 - 2V_{2^{x-1}}(M_2) \\ V_{2^x}(M_2) &= V_{2^{x-1}}(M_2)^2 - 2(V_{2^{x-1}}(M_1)V_{2^{x-1}}(M_3) - 1) \\ V_{2^x}(M_3) &= V_{2^{x-1}}(M_3)^2 - 2V_{2^{x-1}}(M_2), \end{aligned}$$

with special initial values of

$$\begin{aligned} V_2(M_1) &= m_1^2 - 2m_2 \\ V_2(M_2) &= m_2^2 - 2(m_1m_3 - 1) \\ V_2(M_3) &= m_3^2 - 2m_2. \end{aligned} \tag{4.2}$$

Proposition 4.1 below can be used to decrease the number of terms of sequences which is defined in equation (4.1) .

Proposition 4.1. *Let $2^{i+2} < e < 2^{i+3}, r_x \equiv e \mod 2^x$ where $0 \leq r_x < 2^x, 1 \leq b \leq i$ and special initial values of the sequence as defined in* (4.2) *at Definition 4.1. If the secondary initial values V_{r_x}, $V_{r_x+2^x}$, $V_{r_x+2(2^x)}$, and $V_{r_x+3(2^x)}$ are given, then the e-th term of the fourth order Lucas sequence will be able to be generated via the following equations*

$$\begin{aligned} V_{k_x}(M_1) =& V_{k_x-2^x}(M_1)V_{2^x}(M_1) - V_{k_x-2(2^x)}(M_1)V_{2^x}(M_2) \\ &+ V_{k_x-3(2^x)}(M_1)V_{2^x}(M_3) - V_{k_x-4(2^x)}(M_1), \end{aligned} \tag{4.3}$$

where $r_x + 4(2^x) \leq k_x \leq e$ and $k_x = r_x + 2^x s$, for s is an integer.

Proof. We can see the proof of proposition 4.1 in Wong *et al.* [10], Proposition 1. □

The secondary initial values can be generated using (4.3) with primary initial values of,

$$\begin{aligned} V_0(m_1, m_2, m_3, 1) &= 4 \\ V_1(m_1, m_2, m_3, 1) &= m_1 \\ V_2(m_1, m_2, m_3, 1) &= m_1^2 - 2m_2 \qquad \text{and} \\ V_3(m_1, m_2, m_3, 1) &= m_1^3 - 3m_2m_1 + 3m_3. \end{aligned}$$

Making use of Proposition 4.1, some terms of the sequence involving the e-th term of fourth order Lucas sequence can be omitted during the calculation.

4.2. **Computing Sixth Order Lucas Sequence.** Since the second plaintext or ciphertext in the LUC$_{4,6}$ and LUC$_{4,6}$ELG cryptosystems are compute from sixth order Lucas sequence, computational time can be reduced by omitting some terms in the sequence.

The $(a + b)$-th term in the sixth order of the Lucas sequence can be written as

$$\begin{aligned} V_{a+b}(M_2) =& V_a(M_2)V_b(M_2) - V_{a-b}(M_2)(V_b(M_1)V_b(M_3) - 1) \\ &+ V_{a-2b}(M_2)(V_b(M_1)^2 + V_b(M_3)^2 - 2V_b(M_2)) \\ &- V_{a-3b}(M_2)(V_b(M_1)V_b(M_3) - 1) + V_{a-4b}(M_2)V_b(M_2) \\ &- V_{a-4b}(M_2), \end{aligned}$$

where $V_k(M_1)$, $V_k(M_2)$, and $V_k(M_3)$ as defined in Equation (4.1), while m_1, m_2 and m_3 constitute the coefficients for the quartic polynomial.

Definition 4.2. *Let* $6^{j+1} < e < 6^{j+2}, r_b \equiv e \mod 6^b$ *where* $0 \leq r_b < 6$ *and* $1 \leq b \leq j$*, then the special values for* $V_{a+b}(m_2, m_1m_3 - 1, m_1^2 + m_3^2 - 2m_2, m_1m_3 - 1, m_2, 1)$ *can be defined as follow:*

$$\begin{aligned} &V_{6^b}(m_2, m_1m_3 - 1, m_1^2 + m_3^2 - 2m_2, m_1m_3 - 1, m_2, 1) \\ \equiv& Q_{b-1}^2 - 2(P_{b-1}R_{b-1} - 1) \mod n, \end{aligned}$$

$$V_{6^b}(m_1, m_2, m_3, 1) \equiv P_{b-1}^2 - 2Q_{b-1} \mod n \quad \textit{and}$$

$$V_{6^b}(m_3, m_2, m_1, 1) \equiv R_{b-1}^2 - 2Q_{b-1} \mod n,$$

where

$$\begin{aligned} P_{b-1} =& V_3(V_{6^{b-1}}(M_1), V_{6^{b-1}}(M_2), V_{6^{b-1}}(M_3), 1), \\ Q_{b-1} =& V_3(V_{6^{b-1}}(M_2), V_{6^{b-1}}(M_1)V_{6^{b-1}}(M_3) - 1, \\ & V_{6^{b-1}}(M_1)^2 + V_{6^{b-1}}(M_3)^2 - 2V_{6^{b-1}}(M_2), \\ & V_{6^{b-1}}(M_1)V_{6^{b-1}}(M_3) - 1, V_{6^{b-1}}(M_2), 1) \quad \textit{and} \\ R_{b-1} =& V_3(V_{6^{b-1}}(M_3), V_{6^{b-1}}(M_2), V_{6^{b-1}}(M_1), 1), \end{aligned}$$

with $V_{6^{b-1}}(M_1), V_{6^{b-1}}(M_2)$, *and* $V_{6^{b-1}}(M_3)$, *as defined in* (4.1).

Proposition 4.2. *Let* $6^{j+1} < e < 6^{j+2}, r_b \equiv e \mod 6^b$ *where* $0 \leq r_b < 6$ *and* $1 \leq b \leq j$, *and the special values of the sequence were defined at Definition 4.2. If the secondary initial values* $V_{r_b}, V_{r_b+6^b}, V_{r_b+2(6^b)}, V_{r_b+3(6^b)}, V_{r_b+4(6^b)}$ *and* $V_{r_b+5(6^b)}$ *are given, then, the* e*-th term of the sixth order Lucas sequence,* $V_e(m_2, m_1m_3 - 1, m_1^2 + m_3^2 - 2m_2, m_1m_3 - 1, m_2, 1)$ *can be generated by using the following equation,*

$$\begin{aligned} & V_{k_b}(m_2, m_1m_3 - 1, m_1^2 + m_3^2 - 2m_2, m_1m_3 - 1, m_2, 1) \\ \equiv & V_{r_b}(M_2)V_{6^b}(M_2) - V_{r_b+6^b}(M_2)[V_{6^b}(M_1)V_{6^b}(M_3) - 1] \\ & + V_{r_b+2(6^b)}(M_2)[V_{6^b}(M_1)^2 + V_{6^b}(M_3) - 2V_{6^b}(M_2)] \\ & - V_{r_b+3(6^b)}(M_2)[V_{6^b}(M_1)V_{6^b}(M_3) - 1] \\ & + V_{r_b+4(6^b)}(M_2)V_{r_b+6^b}(M_2) \\ & - V_{r_b+5(6^b)}(M_2) \mod n, \end{aligned} \tag{4.4}$$

where $r_b + 6^{b+1} \leq k_b \leq e$, $k_b = r_b + 6^b s$ *and* s *is an integer.*

Proof. This proposition can be proved by mathematical induction. Let $6^{j+1} < e < 6^{j+2}, r_b \equiv e \mod 6^b$ where $0 \leq r_b < 6$ and $1 \leq b \leq j$, then $e = r_b + 6^b t$, where t is an integer greater or equal to 6. Therefore, $r_b < 6^{j+1}$, $r_b + 6^b < 6^{j+1}$, $r_b + 2(6^b) < 6^{j+1}$, $r_b + 3(6^b) < 6^{j+1}$, $r_b + 4(6^b) < 6^{j+1}$, and $r_b + 5(6^b) < 6^{j+1}$. Thus, if the secondary initial values, $V_{r_b}, V_{r_b+6^b}, V_{r_b+2(6^b)}, V_{r_b+3(6^b)}, V_{r_b+4(6^b)}$ and $V_{r_b+5(6^b)}$ are given, the i-th term in the sixth order of Lucas sequence can be generated from $r_b + 6^{b+1}$ until $e = r_b + 6^{b+1}t$. □

5. Result and Discussion

This section describes the algorithm for computations based on Proposition 4.1 and 4.2. The algorithm for the process of encryption and can be summarized as follow:

Step 1. Determine the values of i and j, where $2^{i+2} < e < 2^{i+3}$ and $6^{j+1} < e < 6^{j+2}$ where e is the encryption key.

Step 2. Define the primary initial values. The four primary initial values of the fourth order of Lucas sequence are

$$\begin{aligned} V_0(x_1, x_2, x_3, 1) &= 4, \\ V_1(x_1, x_2, x_3, 1) &= x_1, \\ V_2(x_1, x_2, x_3, 1) &= x_1^2 - 2x_2, \\ V_3(x_1, x_2, x_3, 1) &= x_1^3 - 3x_1x_2 + 3x_3, \end{aligned}$$

where $(x_1, x_2, x_3) = (m_1, m_2, m_3)$ for the first ciphertext and $(x_1, x_2, x_3) = (m_3, m_2, m_1)$ for the third ciphertext. The six primary initial values of the sixth order of Lucas sequence are

$$\begin{aligned} V_0(y_1, y_2, y_3, y_4, y_5, 1) &= 4, \\ V_1(y_1, y_2, y_3, y_4, y_5, 1) &= y_1, \\ V_2(y_1, y_2, y_3, y_4, y_5, 1) &= y_1^2 - 2y_2, \\ V_3(y_1, y_2, y_3, y_4, y_5, 1) &= y_1^3 - 3y_1y_2 + 3y_3, \\ V_4(y_1, y_2, y_3, y_4, y_5, 1) &= y_1^4 - 4y_1^2y_2 + 2y_2^2 + 4y_1y_3 - 4y_4, \\ V_5(y_1, y_2, y_3, y_4, y_5, 1) &= y_1^5 - 5y_1^3y_2 + 5y_2^2y_3 - 5y_2y_3 - 5y_1y_4 + 5y_5, \end{aligned}$$

where $(y_1, y_2, y_3, y_4, y_5) = (m_2, m_1m_3 - 1, m_1^2 + m_3^2 - 2m_2, m_1m_3 - 1, m_2)$ for second ciphertext.

Step 3. Generate the special values for fourth and sixth order Lucas sequence. Loop a from $a = 2$ until i, for the sequences as defined in Definition 4.1 and loop b from $b = 1$ until j, for sequences as defined in Definition 4.2.

Step 4. Generate the secondary initial values. Loop $a = 0$ to i for $r_a \equiv e \mod 2^a$. Then, loop again for k_a from $k_a = r_a + 4(2^a)$ until $r_{a+1} + 3(2^{a+1})$ for sequence defined in (4.3). Loop $b = 0$ to j for $r_b \equiv e \mod 6^b$. Then, loop again for k_b from $k_b = r_b + 6(2^b)$ until $r_{b+1} + 5(6^{b+1})$ for sequence defined in (4.4).

Step 5. Generate the ciphertext. The first ciphertext can be generated by

$$\begin{aligned} V_e(m_1, m_2, m_3, 1) \equiv & V_{k_i-2^i}(M_1)V_{2^i}(M_1) - V_{k_i-2(2^i)}(M_1)V_{2^i}(M_2) \\ & + V_{k_i-3(2^i)}(M_1)V_{2^i}(M_3) - V_{k_i-4(2^i)}(M_1) \mod n. \end{aligned}$$

The second ciphertext can be generated by

$$\begin{aligned} & V_e(m_2, m_1m_3 - 1, m_1^2 + m_3^2 - 2m_2, m_1m_3 - 1, m_2, 1) \\ \equiv & V_{k_j}(M_2)V_{6^j}(M_2) - V_{k_j+6^j}(M_2)[V_{6^j}(M_1)V_{6^j}(M_3) - 1] \\ & + V_{k_j+2(6^j)}(M_2)[V_{6^j}(M_1)^2 + V_{6^j}(M_3) - 2V_{6^j}(M_2)] \\ & - V_{k_j+3(6^j)}(M_2)[V_{6^j}(M_1)V_{6^j}(M_3) - 1] \\ & + V_{k_j+4(6^j)}(M_2)V_{k_j+6^j}(M_2) \\ & - V_{k_j+5(6^j)}(M_2) \mod n. \end{aligned}$$

The third ciphertext can be generated by

$$\begin{aligned} V_e(m_3, m_2, m_1, 1) \equiv & V_{k_i-2^i}(M_3)V_{2^i}(M_3) - V_{k_i-2(2^i)}(M_3)V_{2^i}(M_2) \\ & + V_{k_i-3(2^i)}(M_3)V_{2^i}(M_1) - V_{k_i-4(2^i)}(M_3) \mod n. \end{aligned}$$

The algorithm for the process of decryption is similar to the algorithm of the encryption process. The plaintexts (m_1, m_2, m_3) are replaced by the ciphertexts (c_1, c_2, c_3). Whilst, the encryption key e is replaced by the decryption key d. In this manner, the algorithm can be transformed into the algorithm of the process decryption.

The computation time depends on the number of terms in the sequence. The computational time can be shortened if the number of terms is reduced. In the $\text{LUC}_{4,6}$ and $\text{LUC}_{4,6}$ELG, each set of plaintext or ciphertext contains three messages. If the encryption key e is the term of the sequence, then the total number of terms to compute is $3e$ for the common computational method.

In this paper, we propose a method requires to compute primary initial values for fourth order of Lucas sequence and six primary initial values for sixth order of Lucas sequence. The special values for fourth order Lucas sequence are V_{2^a} where $2 \leq a \leq i$ and $2^{i+2} < e < 2^{i+3}$. Therefore, the number of special values for fourth order Lucas sequence is $i-1$. The special values for sixth order Lucas sequence are V_{6^b} where $2 \leq b \leq j$ and $6^{j+1} < e < 6^{j+2}$. However, it is necessary to determine V_{3^b} before finding the special values. Therefore, the number of special values for sixth order Lucas sequence is $2j-2$. The number of terms for secondary initial values of the fourth order Lucas sequence is $\sum_{a=0}^{i-1}(k_{a+1} - k_a)/2^a$, while the number of terms

for secondary initial values of the sixth order Lucas sequence is $\sum_{b=0}^{j-1}(k_{b+1}-k_b)/6^b$, where k_{a+1}, k_a, k_{b+1} and k_b are secondary initial values for level $a+1$, a, $b+1$ and b, respectively.

Finally, the number of terms to compute at the final step in both of fourth and sixth order Lucas sequence are $(e-k_i)/2^i$ and $(e-k_j)/6^j$ respectively. There are some repeating terms like V_4 in fourth order Lucas sequence and V_6,V_{18} for sixth order Lucas sequence.

The comparison between the number of terms compute using common method and proposed method is tabulated in Table 1. Result shows that the number of terms decreased and consequently reducing the computational efforts in proposed method.

TABLE 1. A comparison between the number of terms compute for the common method and the proposed method

e	Common method	Proposed method	Number of terms to omit
517	1551	145	1406
1031	3093	175	2918
2053	6159	198	5961
4099	12297	215	12082
8209	24627	233	24394

6. CONCLUSION

In this study, a method to decrease the computational time in LUC$_{4,6}$ and LUC$_{4,6}$ELG cryptosystems had been proposed. The equation $(a+b)$-th term of fourth and sixth order Lucas sequence have been rewritten based on the characteristic of high order Lucas sequence. An algorithm has been constructed based on the equation. This algorithm can be extracted/explain in five steps. Firstly, based on the encryption key or decryption key, the values i and j for the following steps used are determined. Secondly, the primary initial values based on the plaintexts or ciphertexts are computed. Thirdly, special values, which depend on values i and j is calculated. Fourthly, special values are used to calculate the secondary initial values. Finally, after computing the secondary initial values, the ciphertexts can be generated or the plaintexts can be recovered based on special values and

secondary initial values. Compared with existing algorithm, the proposed algorithm reduce the number of terms in calculation. The computational efforts is reduced significantly. These effort making $\text{LUC}_{4,6}$ and $\text{LUC}_{4,6}$ELG cryptosystems more efficient for security implementation.

ACKNOWLEDGMENT

The authors wish to acknowledge financial support from Putra Grant (Vote No. 9664500).

REFERENCES

[1] M.A.M. ISA, N.N.A. RAHMAN, M.A. ASBULLAH, M.H.A. SATHAR, A.F.N. RASEDEE: *On the Insecurity of Generalized (Rivest-Shamir-Adleman)-Advance and Adaptable Cryptosystem*, Journal of Physics: Conference Series, **1366**(1)(2019), 012021

[2] Z. MAHAD, M.A. ASBULLAH, M.R.K. ARIFFIN: *Efficient Methods to Overcome Rabin Cryptosystem Decryption Failure*, Malaysian Journal of Mathematical Sciences, **11**(S) (2017), 9–20.

[3] T.J. WONG, M.R.M. SAID, K.A.M. ATAN, B. URAL: *The Quartic Analog to the RSA Cryptosystem*, Malaysian Journal of Mathematical Sciences, **1**(1) (2007), 63–81.

[4] W.T. JIN, H. KAMARULHALI, M.R.M. SAID: *On the Hastad's Attack to LUC4,6 Cryptosystem and compared with Other RSA-Type Cryptosystem*, Malaysian Journal of Mathematical Science, **7**(S)(2013), 1–17.

[5] I.N. SARBINI, T.J. WONG, M. OTHMAN, L.F. KOO, A.F.N. RASEDEE, P.H. YIU, F.N. NANING: *Cryptographic attack on luc-type cryptosystems using gmitm (Type 1)*, Journal of Advanced Research in Dynamical and Control Systems, **11**(12 Special Issue) (2019), 806–813.

[6] T.J. WONG, M.R.M. SAID, M. OTHMAN, L.F. KOO : *On the common modulus attack into LUC4,6 Cryptosystem*, AIP Conference Proceeding, **1660** (2015), 090052.

[7] D.E. KNUTH: *The Art of Computer Programming, Volume 2: Seminumerical Algorithms*, 3rd ed., Massachusetts, United State, 1998.

[8] P.L. MONTGOMERY: *Evaluating Recurrences of Form* $X_{m+n} = f(X_m, Xn, X_{m-n})$ *via Lucas Chains*, unpublished manuscript.

[9] S.M. YEN, C.S. LAIH: *Fast Algorithms for LUC Digital Signature Computation*, IEEE Proceedings Computers and Digital Techniques, **142** (1995), 165–169.

[10] T.J. WONG, M.R.M. SAID, M. OTHMAN, L.F. KOO : *A Method to Decrease Computation Time for Fourth Order Lucas Sequence*, AIP Conference Proceeding, **1557** (2013), 55–58.

Department of Science and Technology, Universiti Putra Malaysia,
Bintulu Campus, Nyabau Road, 97008 Bintulu, Sarawak, Malaysia.
Email address: w.tzejin@upm.edu.my

Faculty Computer Science and Technology, Universiti Malaysia Sarawak
94300 Kota Samarahan, Sarawak, Malaysia.
Email address: sinabila@unimas.my

Department of Science and Technology, Universiti Putra Malaysia,
Bintulu Campus, Nyabau Road, 97008 Bintulu, Sarawak, Malaysia.
Email address: leefeng@upm.edu.my

Department of Science and Technology, Universiti Putra Malaysia,
Bintulu Campus, Nyabau Road, 97008 Bintulu, Sarawak, Malaysia.
Email address: fatinhanaz@upm.edu.my

Faculty of Computer Science and Information Technology, Universiti Putra Malaysia
43400 UPM Serdang, Selangor, Malaysia.
Email address: mothman@upm.edu.my

Department of Social Science and Management, Universiti Putra Malaysia,
Bintulu Campus, Nyabau Road, 97008 Bintulu, Sarawak, Malaysia.
Email address: mdmaulana@upm.edu.my