A Comparative *S*-Index in Factoring RSA Modulus via Lucas Sequences

[1]**Nur Azman Abu,** [1]**Shekh Faisal Abdul-Latip and** [2]**Muhammad Rezal Kamel Ariffin**

[1]INSFORNET, Faculty of ICT,
Universiti Teknikal Malaysia Melaka,
Hang Tuah Jaya, Durian Tunggal, 76100 Melaka
email: {nura, shekhfaisal}@utem.edu.my

[2]AI-Kindi Cryptography Research Laboratory,
Institute for Mathematical Research,
Universiti Putra Malaysia,
43400 Serdang, Selangor, Malaysia
e-mail: rezal@upm.edu.my

ABSTRACT

General Lucas sequences are practically useful in cryptography. In the past quarter century, factoring large RSA modulo into its primes is one of the most important and most challenging problems in computational number theory. A factoring technique on RSA modulo is mainly hindered by the strong prime properties. The success of factoring few large RSA modulo within the last few decades has been due to computing prowess overcoming one strong prime of RSA modulo. In this paper, some useful properties of Lucas sequences shall be explored in factoring RSA modulo. This paper introduces the *S*-index formation in solving quadratic equation modulo *N*. The *S*-index pattern is very useful in designing an algorithm to factor RSA modulo. At any instance in the factoring algorithm, the accumulative result stands independently. In effect, there is no clear direction to maneuver whether to go left or right. The S-index will add another comparative tool to better maneuver in a factoring process. On one hand, it shall remain a theoretical challenge to overcome the strong prime properties. On the other hand, it shall remain a computational challenge to achieve a running time within polynomial time to factor RSA modulo. This paper will propose an avenue to do both using general Lucas sequences.

INTRODUCTION

General Lucas sequences have made significant contribution to the field of cryptography. Lucas sequence V has been proposed to be used for public key cryptosystem (Smith and Lennon, 1994), in a manner similar to the famous RSA (Rivest et. al., 1978), but using Lucas sequences modulo a composite number instead of exponentiation. It has stipulated to have the same security level as RSA for the same size key, but is about twice as slow. A special Lucas sequence has been used to directly factor pseudo prime numbers especially Carmichael numbers (Abu et. al., 2004).

An efficient computation of general Lucas sequences can be found in (Joye and Quisquater, 1996). Zhenxiang Zhang has shown on how to factor an RSA modulo into its primes near both multiples of group orders $P-1$ or $P+1$ and respectively $Q-1$ or $Q+1$ using Lucas sequences. An asymmetric key GM cryptosystem has been developed by Shafi Goldwasser and Silvio Micali in 1982. It is semantically secure based on intractability of the quadratic residue problem modulo $N = PQ$ where P and Q are large primes. The difficulties of decrypting the ciphertext without the key pair (P, Q) is solely based on a comparative interactive challenge on whether a given ciphertext c is a quadratic residue modulo N when the Jacobi symbol for c is +1.

The non-positional nature of Residue Number Systems (RNS) is very efficient in a single arithmetic computing without any hassle of carry propagations. Unlike in the common index number system, RNS has a drawback in comparison. There is no ease general method for magnitude comparison in RNS. This inability to compare two numbers whichever is larger makes it difficult to operate on large modulo efficiently especially in the field of cryptography. (Sousa, 2007). The magnitude comparison in RNS is equivalent to the Comparative *S*-Index in this paper.

CRITERIA OF STRONG RSA PRIMES

Let *N* be the product of two primes, *P* and *Q*. It may be desirable to use strong primes for *P* and *Q*. These are prime numbers with certain properties that make the product *N* difficult to factor by known factoring methods. The selection of *P* and *Q* as strong primes has been recommended, prior to the year 2000, as a way to safeguard the well-known classical factoring algorithm (Rivest and Silverman, 2001). However, these basic strong prime criteria are independently imposed on *P* or *Q*.

Among the properties of strong RSA modulo $N = PQ$ are as follows.

Criterion 1: $P-1$ and $P+1$ consists of a large prime factor.

Let $P-1 = P_0^- \cdot P_1^- \cdot \cdots \cdot P_{k^-}^-$ and $P+1 = P_0^+ \cdot P_1^+ \cdot \cdots \cdot P_{k^+}^+$. The largest prime factors $P_{k^-}^-$ and $P_{k^+}^+$ should be larger than 256-bit for 512-bit *P*.

Criterion 2: $Q-1$ and $Q+1$ consist of a large prime factor.

Let $Q-1 = Q_0^- \cdot Q_1^- \cdot \cdots \cdot Q_{k^-}^-$ and $Q+1 = Q_0^+ \cdot Q_1^+ \cdot \cdots \cdot Q_{k^+}^+$. Respectively, the largest prime factors $Q_{k^-}^-$ and $Q_{k^+}^+$ should be larger than 256-bit for 512-bit *Q*.

Criterion 3: Recursively, for each largest factor, $P_{k^-}^- - 1$ and $P_{k^+}^+ - 1$ must also consist of large enough prime factor, namely, $P_{k^{--}}^{--}$ and $P_{k^{+-}}^{+-}$ following the notation in (Rivest and Silverman, 2001).

Criterion 4: Each largest factor of the prime $Q_{k^-}^- - 1$ and $Q_{k^+}^+ - 1$ must also consist of large enough prime factor namely, $Q_{k^{--}}^{--}$ and $Q_{k^{+-}}^{+-}$ respectively.

Factoring the RSA modulo *N* is well known to be unfeasible. Recently, (Boudaoud, 2009) explores another practical approach to surmount this major difficulty by finding the factorization of an integer in a small neighborhood of *N* instead of *N*. (Bakhtiari and Maarof, 2012) pointed out that there are more than one set of decryption key (*d*, *N*) on a given set of RSA encryption key (*e*, *N*). However the distance between them is lcm($P-1$, $Q-1$) which is ruled by the basic strong prime criteria.

Let an elliptic curve be the set of points

$$E(a, b) = \{\ (x, y, z) : y^2 z \equiv x^3 + axz^2 + bz^3 \ (\text{mod p})\ \}$$

By the end of the century, it has been noted to be useless to concentrate on strong primes. It is unnecessary to protect against factoring attacks by building large prime factors into $P-1$ or $P+1$ since the adversary can instead attempt to overcome by finding an elliptic curve $E(a, b)$ whose size

$$P+1-2\sqrt{P} \le \left|E(a,b)\right| \le P+1+2\sqrt{P}$$

is smooth (Rivest and Silverman, 2001).

GENERAL LUCAS SEQUENCES

Given integer parameters $p>2$ and $q>0$, the general Lucas sequences give rise to two functions similar to exponentiation, namely, U_n and V_n.

$U_0 = 0,\ U_1 = 1,\ U_n = p\cdot U_{n-1} - q\cdot U_{n-2}$
$V_0 = 2,\ V_1 = p,\ V_n = p\cdot V_{n-1} - q\cdot V_{n-2}$

Calculating an element of a Lucas sequence can be done in a very similar pattern to exponentiation using a power modulo operation. It may be helpful to think of p as the base and the index n as the exponent. The closed forms of the general Lucas sequences are:

$$U_n = \frac{\alpha^n - \beta^n}{\alpha - \beta} \text{ and } V_n = \alpha^n + \beta^n .$$

where α and β are the two roots of the quadratic polynomial $x^2 - px + q$.

These classical Lucas sequences U_n and V_n are generated from second order recursions with integer variables (p, q) and discriminant $\delta = p^2 - 4q$. In the case of $(p, q) = (1, 1)$, the Lucas sequence U_n is popularly known as Fibonacci numbers, and their companions V_n are the Lucas numbers. The requirement on P and Q, to be strong primes by making $P\pm1$ and $Q\pm1$ to have large prime factors, may no longer appear to be adequately substantiated in the view of the best factorisation algorithms known today.

Pollard Rho Method basically can achieve rapid factorization if $P-1$ consists of only small prime factors. On the other hand, similar result can be said also about $P+1$. This method of integer factorisation is originally described in (Williams, 1982). It can find a large factor P very quickly when $P+1$ is composed of only small factors. (Zhang, 2001) has also shown how the general Lucas Sequence can be employed to exploit any weak primes from both sides, the $P-1$ and $P+1$.

CRITERIA ON GENERAL LUCAS SEQUENCES

Let $N = PQ$. For a given parameters p and q, take $\delta = p^2 - 4q$. Let $\varepsilon_P = \left(\frac{\delta}{P}\right)$ and $\varepsilon_Q = \left(\frac{\delta}{Q}\right)$. The subscript to the epsilon ε is usually left out within the context of known prime P or Q and $\varepsilon_N = \left(\frac{\delta}{N}\right) = \left(\frac{\delta}{P}\right)\cdot\left(\frac{\delta}{Q}\right) = \varepsilon_P \cdot \varepsilon_Q$.

For instance,

$$\varepsilon_P = \left(\frac{\delta}{P}\right) = \begin{cases} +1, & \delta \text{ is quadratic residue mod } P \\ -1, & \delta \text{ is non-quadratic residue mod } P \end{cases}$$

Here the criteria of general Lucas sequences are being compactly summarised. They are very practical tools in factoring process.

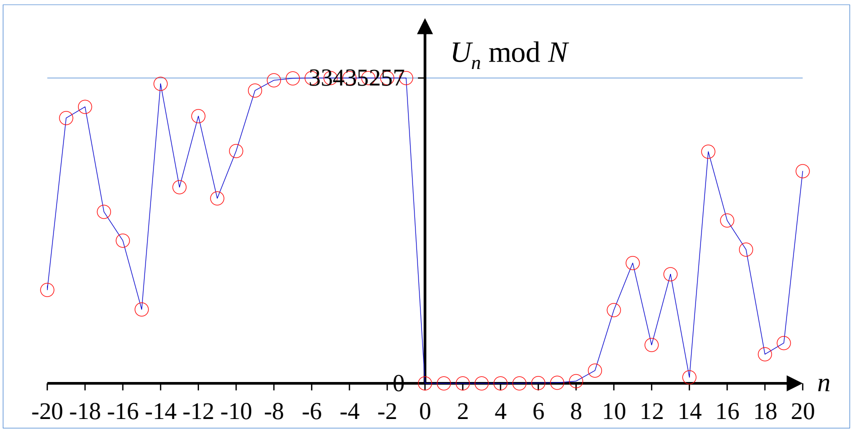

Figure 1: U_n mod N sequence is odd with respect to the center period C.

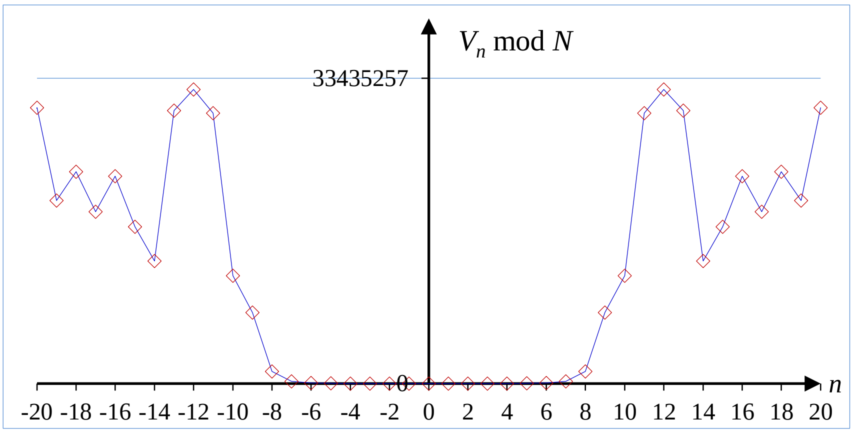

Figure 2: V_n mod N sequence is even with respect to the center period C.

Criterion 1: All the operations here are done modulo N. The maximum period of the general Lucas sequences U and V modulo N of parameters p and q is C = lcm($P - \varepsilon_P$)($Q - \varepsilon_Q$). This criteria has been regarded as a generalisation of the Euler totient function for Lucas functions, the Lehmer totient function (Lehmer, 1930).

Table 1: The values of general Lucas sequences U_n *mod* N and V_n *mod* N near the center C.

n	U_n	V_n
-20	10216491	30209367
-19	29036528	20045158
-18	30261649	23191067
-17	18792338	18795473
-16	15621865	22711257

-15	8068338	17166298
-14	32788163	13416017
-13	21484355	29894547
-12	29247453	32210237
-11	20259335	29625847
-10	25438043	11803817
-9	32063152	7761798
-8	33199841	1331714
-7	33394866	228486
-6	33428327	39202
-5	33434068	6726
-4	33435053	1154
-3	33435222	198
-2	33435251	34
-1	33435256	6
0	0	2
1	1	6
2	6	34
3	35	198
4	204	1154
5	1189	6726
6	6930	39202
7	40391	228486
8	235416	1331714
9	1372105	7761798
10	7997214	11803817
11	13175922	29625847
12	4187804	32210237
13	11950902	29894547
14	647094	13416017
15	25366919	17166298
16	17813392	22711257
17	14642919	18795473
18	3173608	23191067
19	4398729	20045158
20	23218766	30209367

Criterion 2: The Lucas sequence U is odd while V is even with respect to the period as shown in the Figures 1 and 2 above, i.e. $U_{kC-n} = -U_{kC+n}$ and $V_{kC-n} = V_{kC+n}$ for any integer k and positive integer n from the center period C=0.

Let the parameters of general Lucas sequences be $(p, q) = (6, 1)$. The values of both Lucas sequences have been listed in Table 1. The graphs in Figures 1 and 2 above show typical characteristics of an odd sequence U_n (mod N) and an even sequence V_n mod N for $N = PQ = 4073 \cdot 8209 = 33435257$. This criterion has made Lucas sequence V appear to be a better reference than U in the LUC public-key system.

Criterion 3: The center values of the general Lucas sequences U and V modulo RSA primes are as follows;

i. $U_{k(P-\varepsilon)} \equiv 0 \pmod{P}$ for any positive integer k.

ii. $V_{k(P-\varepsilon)} \equiv 2q^{\frac{k(1-\varepsilon)}{2}} \pmod{P}$ for any positive integer k.

iii. $U_{k(Q-\varepsilon)} \equiv 0 \pmod{Q}$ for any positive integer k.

iv. $V_{k(Q-\varepsilon)} \equiv 2q^{\frac{k(1-\varepsilon)}{2}} \pmod{Q}$ for any positive integer k.

Preferably the second parameter q is set to be one(1) so that the sequence V will always have consistent output 2 modulo N at a multiple instance of period C.

Criterion 4: These following characteristics have been observed based on the previous research on general Lucas sequences. Most researchers insist on Criterion 3 as a more practical form for factoring purposes. Nevertheless, these criteria are more flexible in factoring angles to choose from.

i. $U_{j(P-\varepsilon)+L} - U_{k(P-\varepsilon)+L} \equiv 0 \pmod{P}$ for some positive integers j and k.
ii. $V_{j(P-\varepsilon)\pm L} - V_{k(P-\varepsilon)\pm L} \equiv 0 \pmod{P}$ for some positive integers j and k.
iii. $U_{j(Q-\varepsilon)+L} - U_{k(Q-\varepsilon)+L} \equiv 0 \pmod{Q}$ for some positive integers j and k.
iv. $V_{j(Q-\varepsilon)\pm L} - V_{k(Q-\varepsilon)\pm L} \equiv 0 \pmod{Q}$ for some positive integers j and k.

It is a necessary condition that $j \neq k$ for integer $-R < L < R$ where R is typically referred to the absolute difference between the primes P and Q. This last criterion is the most useful but by far the most elusive characteristic of the general Lucas sequences in designing a factoring algorithm. It is also noted that Criterion 4 is useful for factoring algorithm if it does not happen simultaneously i.e. the sequence U or V is not equal to the ones modulo N.

Criterion 5: Alternatively, all the criteria above may be summarised in terms of primes P and Q as follows. There are integers $0 \leq a_j, b_k < Q$ and $0 \leq c_j, d_k < P$ such that

i. $U_{j(P-\varepsilon)+L} = a_j \cdot P + U_L \pmod{N}$
ii. $V_{k(P-\varepsilon)\pm L} = b_k \cdot P + V_L \pmod{N}$
iii. $U_{j(Q-\varepsilon)+L} = c_j \cdot Q + U_L \pmod{N}$
iv. $V_{k(Q-\varepsilon)\pm L} = d_k \cdot Q + V_L \pmod{N}$

for every integer L. Thus, an RSA prime can be extracted respectively by taking the greatest common divisor as follows;

i. $P = \gcd(U_{j(P-\varepsilon)+L} - U_L, N)$
ii. $P = \gcd(V_{k(P-\varepsilon)\pm L} - V_L, N)$
iii. $Q = \gcd(U_{j(Q-\varepsilon)+L} - U_L, N)$
iv. $Q = \gcd(V_{k(Q-\varepsilon)\pm L} - V_L, N)$

NEW PROPOSAL ON RSA FACTORING

On one hand, it shall remain a theoretical challenge to overcome the strong prime properties. On the other hand, it shall remain a computational challenge to keep the running time within polynomial time to factor RSA modulo.

According to the Proposition 3.3 in (Khadir, 2008) Let N be the product of two prime factors P and Q, $2 < P < Q$. If we can compute efficiently two odd integers r and s such that $s < P$ and $|sQ - rP| \leq 2^{\frac{K+5}{4}}$ where K is the bit-size of the integer rsN, then we can compute the factors P and Q.

In this paper, a more relaxed requirement shall be made.

Suppose $\varepsilon_N = \left(\frac{c}{N}\right) = \left(\frac{c}{P}\right)\cdot\left(\frac{c}{Q}\right) = \varepsilon_P \cdot \varepsilon_Q = (+1)(+1) = 1.$ Let $R < P < Q$ such that $R = Q - P$.

$$\begin{aligned} N-1 &= (P-1)(Q-1) + (P-1)+(Q-1) \\ &= (P-1)(Q-1) + 2(P-1) + R = (P-1)(Q-1) + 2(Q-1) - R \end{aligned}$$

For a given odd w,

$$\begin{aligned} N-1 + w &= (P-1)(Q-1) + 2(P-1) + (R+w) \\ &= (P-1)(Q-1) + 2(Q-1) - (R-w) \end{aligned}$$

and

$$\begin{aligned} N-1 - w &= (P-1)(Q-1) + 2(P-1) + (R-w) \\ &= (P-1)(Q-1) + 2(Q-1) - (R+w) \end{aligned}$$

Preferably, $w = 1$ is a good starting point.

Let V_n be the special Lucas sequence with parameters $(p, q) = (p, 1)$ so that $p^2 - 4$ is a quadratic residue of N. Then we need to set a special even Lucas sequence such that $V_0 = 2$, $V_1 = p$, $V_2 = p^2 - 2$ and $V_3 = p\cdot V_2 - V_1 = p\cdot(p^2-2) - p = p^3-3p$.

Let $N_0 = N-1$. Suppose an odd indexed sequence only is readily available. Nevertheless, it is sufficient to generate the values of V sequences along other large odd indexes. Since N_0-w and N_0+w are odd, V sequence modulo N can be computed using a special algorithm below. The running time of this textbook Algorithm 1 is still $O(n^3)$ compared to the running time of general Lucas sequences.

Algorithm 1: A textbook algorithm to compute an odd Lucas sequence V.

```
Function Vodd ( $p$, $K$, $N$)
Set $K=b_{n-1}b_{n-2}\ldots b_2b_1b_0$ be odd such that $b_{n-1}=1$ and $b_0=1$.
Left = $V_1$, Right = $V_3$.
for $i=n-2$ down to 1,
  if $b_i=0$,
          Right = Left*Right – $p$ mod $N$,
          Left  = Left$^2$ – 2 mod $N$.
if $b_i=1$,
          Left = Left*Right – $p$ mod $N$,
          Right = Right$^2$ – 2 mod $N$.
end(*for*)
return Left.
```

Following the Lucas sequence V criterion 5, there are integers a, b, c and d such that

$$V_{(N-1)-w} = aP + V_{R-w} = bQ + V_{R+w} \tag{1}$$
$$V_{(N-1)+w} = cP + V_{R+w} = dQ + V_{R-w} \tag{2}$$

Let us compute

$$\begin{aligned} S &= V_{(N-1)-w} + V_{(N-1)+w} \equiv V_{R-w} + V_{R+w} \pmod N \\ T &= V_{(N-1)-w} \cdot V_{(N-1)+w} \equiv V_{R-w} \cdot V_{R+w} \pmod N \end{aligned}$$

Let us scan for a candidate of x of V_r and y of V_s. respectively the satisfy the conditions

$$x + y \equiv S \pmod N \tag{3}$$
$$x \cdot y \equiv T \pmod N \tag{4}$$

From (3), let $y = S - x$, equation (4) will become,

$$x \cdot y = x \cdot (S - x) \equiv T \pmod{\mathrm{N}} \tag{5}$$

Consequently, the problem has been reduced down to solving the quadratic equation modulo N. We shall search for the root of the function

$$f(x) = x \cdot (S - x) - T \pmod{\mathrm{N}}.$$

Let us take the $(2m+1)$ terms at one time as the error function,

$$g(x) = \sum_{i=x-m}^{x+m} f(i)$$

A sample case for N= 4073·8209 = 33435257 is made here. Let the Lucas sequence parameters $(p, q) = (6, 1)$, m=1 and w=3. From (1) and (2),

$$V_{(N-1)-3} = 146 \cdot P + V_{R-3} = -146 \cdot Q + V_{R+3}$$
$$V_{(N-1)+3} = 1561 \cdot P + V_{R+3} = -1561 \cdot Q + V_{R-3}$$

The strategy is to locate the values of V_{R-3} and V_{R+3}. The error function has been plotted within the surrounding region of $V_{(N-1)+3}$ in the Figure 3. We would like to collect the points near zeros.

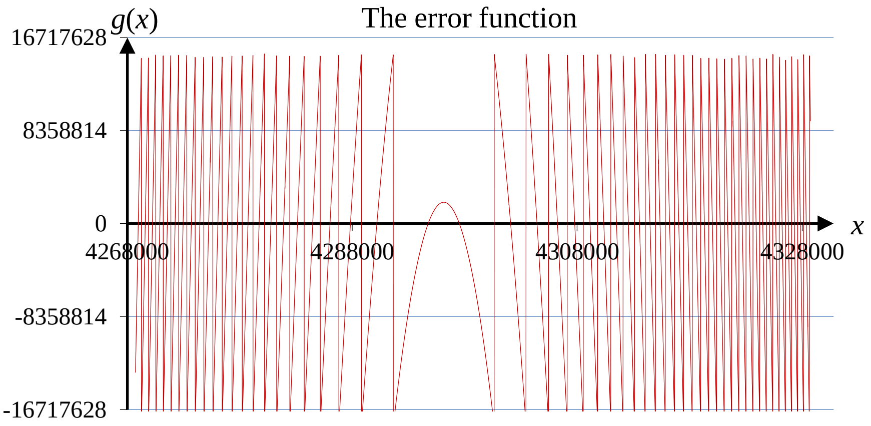

Figure 3: The error function near the zero value.

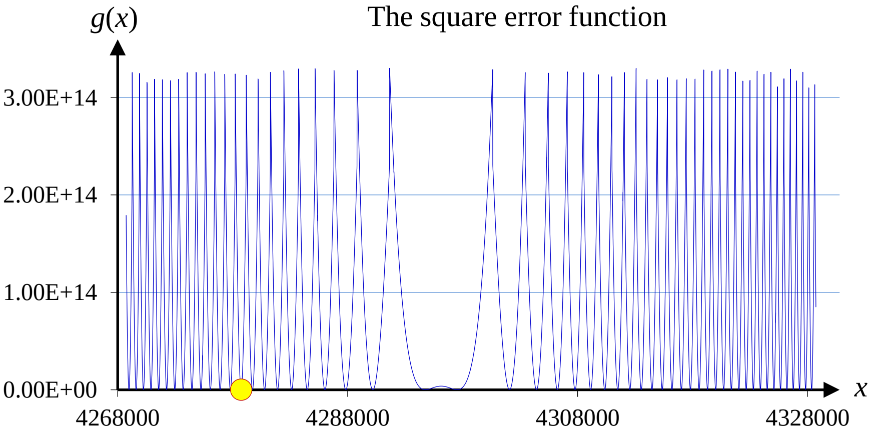

Figure 4: Taking the square on the error function.

Let us take the square of the error function so that we can see the error function value near zeros as depicted in Figure 4. The yellow dot is the target value for $V_{(N-1)+w}$. The touchdown points have been observed here as shown in Figure 5. The errors are probabilistically getting larger as the points are moving away from the center critical point. They are much easier to locate as the points of local minima as shown in Figure 4. The green dot is the target value for $V_{(N-1)+3}$.

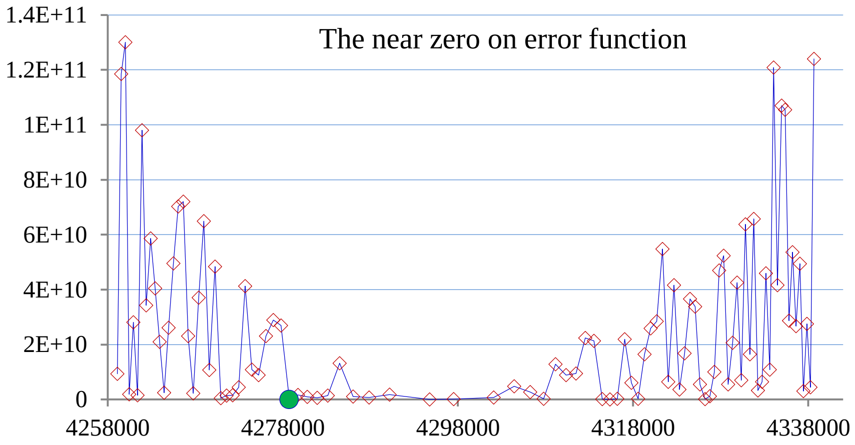

Figure 5: The point of local minima on the error function.

It has also been observed that the distances between the local minima is getting smaller as the points go further away from the center. The list of points x has been plotted in the Figure 6 which form the S pattern.

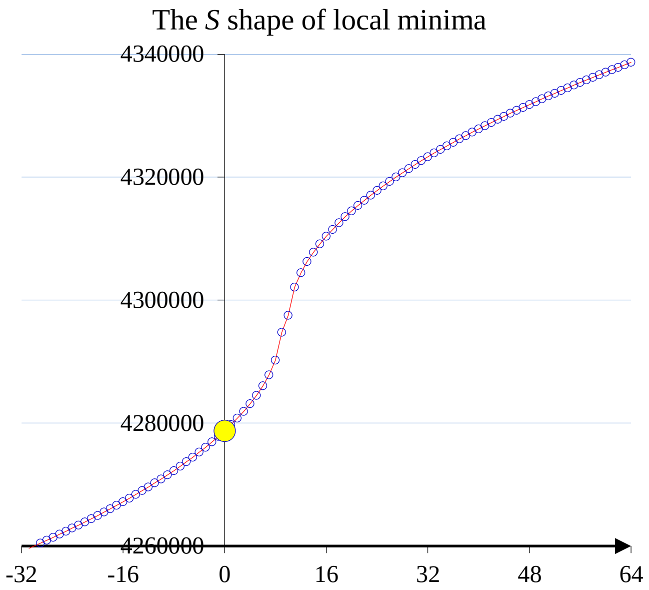

Figure 6: The point of local minima on the error function forms the *S* shape.

According to basic Calculus, a point x to the left of the critical inflection point z, is said to be concaved up and to the right of the critical inflection point z is concaved down respectively.

DISCUSSION

Checking on 3 consecutive ‘touch-down’ at any given point x, will give us a good estimate of the concavity of the surrounding region. A major hurdle in reducing the sub-exponential running time in breaking RSA down to super polynomial running time is the comparative mechanism. At any one time in the factoring algorithm, there has been no mechanism to compare the current position and where to go next. In effect, there is no direction to maneuver whether to go left or right. The *S* index pattern is very useful in designing an algorithm to factor RSA modulo. For instance, in order to determine the quadratic residue on ciphertext c of N, it suffices to predict whether the Lucas sequence V follow the *S* index pattern case 0 or case 1. The *S*-index pattern follows the similar behaviour on all root of the quadratic equation (5) at $V_{(N-1)-3}$, $V_{(N-1)+3}$, V_{R-3} and V_{R+3}. Rather than locating the periodic center of general Lucas sequences U and V as shown in Figures 1 and 2, it is much easier and we stand better chances in locating the *S* pattern on the quadratic equation (5) modulo N.

CONCLUSION

Factoring large integers into primes is one of the most important and most difficult problems in computational number theory. A factoring technique on RSA modulo has been previously hindered by the strong prime properties. Few algorithms have overcome the strong prime criteria of RSA modulo. Nevertheless, they are still subjected to the size of the primes. In this paper, some useful properties of general Lucas sequences have been explored in factoring RSA modulo. A major hurdle in reducing the sub-exponential running time in breaking RSA down to super polynomial running time is the comparative mechanism. At any instance in the factoring algorithm, the accumulative result stands independently. In effect, there is no clear direction to maneuver whether to go left or right. This paper has introduced the *S* index formation in solving quadratic equation modulo N. The *S* index pattern is very useful in designing an algorithm to factor RSA modulo. It shall remain a computational challenge to see whether the running time of factoring RSA modulo can be reduced down to a super polynomial time.

ACKNOWLEDGEMENT

We would like to thank anonymous reviewers of CRYPTOLOGY 2016 for giving valuable suggestions and corrections to this paper. This research paper is supported by Fundamental Research Grant Scheme FRGS/1/2015/ICT05/FTMK/02/F00293 funded by the Ministry of Higher Education, Malaysia.

REFERENCES

Abu, N. A., Suryana, N. and Sahib, S. 2004. Factoring Carmichael Numbers using General Lucas Sequences, Jurnal Matematika, 4(1):131–136.

Bakhtiari, M. and Maarof, M. A., 2012. Serious Security Weakness in RSA Cryptosystem, International Journal of Computer Science Issues, 9.1(3): 175−178.

Boudaoud, Abdelmadjid, 2009. Decomposition of Terms in Lucas Sequences, Journal of Logic & Analysis, 1(4):1–23.

Goldwasser, S. and Micali, S., 1984. Probabilistic Encryption, Journal of Computer and System Sciences, 28:270−299.

Joye, M. and Quisquater, J.-J., 1996. Efficient Computation of Full Lucas Sequences, Electronics Letters, 32(6):537–538.

Khadir, Omar, 2008. Algorithm for Factoring some RSA and Rabin Moduli. J. Discrete Math. Sci. Cryptography, 11(5):537−543.

Lehmer, D. H., 1930. An Extended Theory of Lucas' Functions, Annals of Mathematics, Second Series, 31(3): 419−448.

Smith, Peter J. and Lennon, Michael J. J., 1994. LUC: A New Public Key System, Proceedings of the 9th IFIP International Symposium on Computer Security '93, pp. 097−111, Elsevier Science Publications.

Rivest, R., Shamir, A. and Adleman L., 1978. A Method for Obtaining Digital Signatures and Public Key Cryptosystems, Communications of the ACM, 21(2):120−126.

Rivest, Ronald L. and Silverman, Robert D., 2001. Are Strong Primes Needed for RSA? IACR Cryptology ePrint archive, paper 2001/007, 30 January 2001.

Sousa, L., 2007. Efficient Method for Magnitude Comparison in RNS Based on Two Pairs of Conjugate Moduli, 18th IEEE Symposium on Computer Arithmetic (ARITH '07), 25-27 June 2007, pp. 240 – 250.

Williams, H. C., 1982. A P+1 Method of Factoring, Mathematics of Computation, 39(159): 225−234.

Zhang, Z., 2001. Using Lucas Sequences to Factor Large Integers near Group Orders, Fibonacci Quarterly, 39(3):228–237.