Journal of Computer Science 8 (4): 523-527, 2012
ISSN 1549-3636

Computation of Private Key Based on Divide-By-Prime for Luc Cryptosystems

Zulkarnain Md Ali and Nawara Makhzoum Alhassan Makhzoum
School of Computer Science, Faculty of Information Science and Technology,
University Kebangsaan Malaysia, 43600 Bangi, Selangor Darul Ehsan, Malaysia

Abstract: Problem statement: One of the public key cryptosystem is Luc cryptosystems. This system used Lucas Function for encryption and decryption process. Lucas Function is a special form of second-order linear recurrence relation. An encyption process is used to encrypt an original message to ciphertext by using public key. A decryption process is the process to decrypt a ciphertext into original message using private key. The existing algorithm on computing private key computation involved some redundant computations. **Approach:** In this study, an efficient algorithm to compute private key for Luc cryptosystem is developed. The Extended Euclidean Algorithm will be enhanced by implementing Divide-By-Prime in its computations. The comparison is focused on the computation time by the existing and new algorithms. The more efficient algorithm means the better computation time. The shorter computation time the better algorithm. **Results:** A new algorithm shows better computation time. In all experiments, the computation time by new algorithm is always better than the existing algorithm. **Conclusion:** The new computation algorithm that based on Divide-By-Prime provided better efficiency of decryption process compared to the existing algorithm.

Key words: Luc cryptosystem, decryption process, private key

INTRODUCTION

Public key cryptosystem is a way that is used a secret communication between the sender and receiver, without needing for a secret key exchange and it can used for create a digital signature (Diffie and Hellman, 1976).

Public key cryptosystem is a widely used technology around the world, which enables information to be transmission in a secret channel on the Internet. An encryption process is the process that used to obtain ciphertext C from original message P using public key e. While, in reverse, the decryption process is obtained original message P by decryption the ciphertext C using private key d.

In fact, the encryption of P is relatively easy, since the plaintext P and public key e are known publically. The knowledge of two primes p and q is not important, because the value of these two primes is known as N where N is the product of p and q.

On the other hand, the decryption process is not easy as the encryption; the reason is that, the private key is hidden from the public. It is difficult to obtain it. The strength of cryptosystem depends on the length of public key e and the two prime's p and q. In fact, the increasing of the size in these parameters will also be increased the time required for the decryption computation.

Diffie and Hellman (1976) introduced the concept of public key cryptography, which opened up a whole new research field within the cryptographic community. One of the first public key cryptosystem technique and probably widely used is the RSA (Rivest *et al*., 1978). In RSA using a modular exponentiation of message block for a very large power after that, reducing this number to modulo N, where n equal to multiply of two large primes p and q.

Smith and Lennon (1993) then introduced a new technique of public key cryptosystem based on Lucas Function, which is believed offers better alternative to the RSA. It is used Lucas function to perform the processing of encryption and decryption instead of using exponentiation technique.

There are some researcher, interested on using Luc cryptosystem and also introduced fast computation algorithms (Horster *et al*., 1996; Ali *et al*., 2007; Othman *et al*., 2008).

There is a difficult mathematical problem in Luc as in RSA, In RSA the mathematical problem is known as the Discrete Logarithm (DL). Although, Luc use Lucas functions, it's still based on analogous to the DL problem (Smith and Lennon 1993). Sometimes, implementation of Lucas Functions

Corresponding Author: Zulkarnain Md Ali, School of Comptuer Science, Faculty of Information Science and Technology, University Kebangsaan Malaysia, 43600 Bangi, Selangor Darul Ehsan, Malaysia

ciphers has large complication in timing consumer.

In this study, the proposed algorithms will compare with an existing algorithm which is proposed in (Ali *et al*., 2009). There are three set of data are tested on each algorithm. These data could be categories in different size of messages, public keys and two relatively primes.

The proposed algorithms and the existing algorithm will be tested on every set of data. In addition the computation time will be recorded for each algorithm. The computation time of each algorithm can decided which algorithm is better in term of speed and efficiency.

MATERIALS AND METHODS

Lucas Functions: Two functions V_n and U_n are defined in Lucas sequences as follows:

$V_0 = 2, V_1 = P; V_1 = PV_{n-1}\text{-}QV_{n-2}$ for $n \geq 2$
$U_0 = 0, U_1 = 1; U_n = PU_{n-1}\text{–}QU_{n-2}$ for $n \geq 2$

The computations of V_n need huge computations in view of the fact that the nature of Lucas Functions is a recurrence relation.

The computation of V_n requests two previous values in Lucas Functions computation. The primary values have to be V_0 and V_1.

Encryption and Decryption processes for luc cryptosystem: The ciphertext C is obtained by encrypting the plain text, P by: Enc (P) = Ve (P, 1) (mod N) = C (mod N). Where, e is a public key while Ve is a Lucas Function.

On the other hand, the decryption process by : Dec (C) = $V_d(C,1) = V_d (V_e (P,1),1) = V_{ed}(P,1)$ = P (mod N). Where d is private key and V_d is a Lucas Function.

Lucas functions properties: There are properties of Lucas Function which useful for encryption and decryption process (Smith and Lennon, 1993) and (Horster *et al*., 1996) Eq. 1-4:

$$V_n = PV_{n-1}\text{-}V_{n-2}, \quad (1)$$

$$V_{2n} = V_n^2 - 2Q_n, \quad (2)$$

$$V_{2n-1} = V_nV_{n-1} - MQ_{n-1}, \quad (3)$$

$$V_{2n+1} = P_{Vn2} \text{-}QV_nV_{n-1} - PQ_n \quad (4)$$

The initial values are $V_0 = 2$ and $V_1 = P$. Where, $D = P^2 - 4Q$ is a discriminant.

Important Number Theories Techniques: The very basic and important number theories techniques are required. The following section will discussed the features briefly.

Legendre Symbol (LS): Legendre symbol is a multiplicative function with values 1, −1, or 0, if (**a)** is an integer number and (**p)** is an odd prime, the Legendre Symbol $\left(\frac{a}{p}\right)$ is:

- 0 if p divides a; else
- 1 if a is a quadratic residue modulo p,
- - 1 if a is a quadratic non-residue modulo p.

Some properties of Legendre Symbol which can be speed up its computation.

- Let p be an odd prime, then $\left(\frac{ab}{p}\right)=\left(\frac{a}{p}\right)\left(\frac{b}{p}\right)$
- If $a \equiv b \pmod p$, then $\left(\frac{a}{p}\right)=\left(\frac{b}{p}\right)$
- For b prime to a, $\left(\frac{ab^2}{p}\right)=\left(\frac{a}{p}\right)$
- $\left(\frac{a}{p}\right)=1$
- $\left(\frac{-1}{p}\right)=(-1)^{\frac{(p-1)}{2}}$
- If p and q are odd primes then $\left(\frac{a}{p}\right)=\left(\frac{a}{p}\right)(-1)^{\left(\frac{p-1}{2}\right)\left(\frac{q-1}{2}\right)}$

Least Common Multiple (LCM): The Least Common Multiple of two integers x and y are the smallest positive integer which is divisible by x of y and it is multiple. It could be divided by x and y with a remainder (Knuth, 1981). For example to find LCM by using Division by primes:

- Divide all the numbers by the smallest prime which could divide any of them at the same time
- Then continues in the same way until all prime numbers
- The last step multiply all prime with the last remainder from each number

Let find the Least Common Multiple for 1092 and 1170. Refer to Fig. 1 that has explanation on using the Divide-By-Prime. The LCM for 1092 and 1170 is 2.3.13.14.15 = 16380. See the example below.

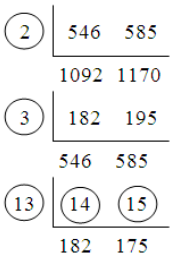

Fig. 1: The using of division by prime to find least common multiple.

Input: a, b and two vectors (1,0) and (0,1)
1: Divide the larger of the two numbers by the smaller - Quotient (q)
2: Subtract q times the smaller from the larger
3: Subtract q times the vector corresponding to the smaller from the vector corresponding to the larger.
4. Repeat from 1 to 3 until the result is zero
Output: Print the preceding results as GCD(a, b)

Fig. 2: Extended Euclid Algorithm

Extended Euclidean Algorithm (EEA): The Extended Euclidean Algorithm is used to find the Greatest Common Divisor (GCD) of two integers a and b and it is an extension to the Euclidean Fig. 2.

It is also can be used to find the integers x and y in ax+by = GCD (a.b). This is a useful technique when a and b are co primes, because x is the modular multiplicative inverse of a modulo b. (Silverman, 2006; Knuth, 1981).

Current Methods: A private key d for Luc cryptosystem could be computed by following these steps:

- de ≡ 1 (mod r), e is a public key
- r = LCM (x , y)
- x = p - LS(p) and y = q - LS(q)
- LS(p) = D/p and LS(q) = D/q
- D = C^2- 4

Note that LCM is least common multiple, LS is Legendre Symbol, D is discriminant, C is Cipher text, e is a public key and d is private key.

This technique suffers lots of computations and requires more computation time. The computation of private key d used the slower computation technique such as Least Common Multiple (LCM) which used Greater Common Division (GCD) and Legendre Symbol (LS) which used computing of power for calculation.

The enhancement of computation of Legendre Symbols can be use in designing a proposed technique of computing private key. The new approach on computing Legendre Symbols is shown in detail in Fig. 4 below.

Input : e, C, p and q
Calculate D=C2 – 4
Calculate LS(p) = LS
Calculate LS(q) = LS
X = p – L(p)
Y = q – L(q)
r = LCM(x, y))
d=EEA(r, e)
Output : d

Fig. 3: An existing algorithm for computing private key

Input : C, p and q
Calculate D = C2 – 4
Calculate L(p)= and L(q)=
X = p – L(p)
Y = q – L(q)
R=LCM=Division by Prime(X, Y)
d=Extended Euclid Algorithm (r, e)
Output : d

Fig. 4: Existing algorithm to compute private key d.

```
Check if a mod p = 0 then
return 0
EndIf
x = a, y = p, L = 1
While true
   x = x mod y
   If x > y/2 then
     x = y-x
     If y %4 =3 then
        L := L *(-1)
     EndIf
   EndIf
   If x = 0 then
     return –1
   EndIf
While x %4= 0
       x = x/4
EndWhile
If x mod 2=0 then
     x = x/2
     t = y mod 8
      If  t = 5 or t = 3 then
         L := L *( -1)
      EndIf
EndIf
If  x = 1 then
    return L
EndIf
If  x, y  mod 4= 3 then
    L := L *( -1)
EndIf
     t = x, x = y, y = t
EndWhile
```

Fig. 5: New Approach of Computing Legendre Symbols (LS)

Proposed methods: The weakness of the existing algorithm was because using the slower algorithm to compute LCM and LS. Moreover, the time consuming for computation private key of Luc Cryptosystem are been reduced. By this fact, the performance of decryption process can be improved. The result of these proposed algorithms would be compared to the existing Fig. 3.

When the computation of Legendre Symbols is done the computation of finding private key can be continued with computation of Least Common Multiple.

```
Input x and y
Initialized  k=1, r1 , r2 ,q1 ,q2 ,
x_size=x_size/2, a=x, b=y
q1=x/2
r1=x mod y
q2=y/2
r2=y mod 2
While (r1=r2=0)
K=k*2
X=q1
Y=q2
End while
N=3,I=0
While (I < x_size)
q1=x/n
r1=x mod y
q2=y/n
r2=y mod 2
While (r1=r2=0)
      K=k*2
      X=q1
      Y=q2
End While
N=n+2
I=i+2
End While
Output R=LCM(x, y)=k
```

Fig. 6: Proposed Algorithm For Least Common Multiple (LCM) Using (DbP)

This technique is base on the method of Divide-By-Prime and it is called DbP.

The detail of this technique is shown in Algorithm 5. Remember that in Fig. 5, x = LS and y = e. Where LS is found from Fig. 4 and e is the public key. The result of Fig. 5 and 6 is R and R is the private key.

RESULTS

The sender uses computing of ciphertext C from the original message, P. Let consider that P = 11111, p = 1093, q = 1171 and e = 1109. To compute C means the computation of V_{1109} (11111,1) = 15407. Meanwhile, C = 15407.To get back the plaintext P, the receiver should compute ciphertext, C. The following steps display how the proposed Algorithms work:

- Ciphertext C = 15407
- Discriminant computation is D = C^2- 4
- Legendre Symbol for (D/p) is (D/1093) = 1
- Legendre Symbol for (D/q) is (D/1171) = 1

Calculate r where:

- r = LCM((1093-1), (1171-1))
- By using Division by Prime to calculate r
- r = LCM(1092,1170) = 2.3.13.14.15 = 16380
- By using EEA to find private key d by e d = 1 (mod 1279903)
- In addition, 1109*d = 1 (mod 1279903)
- Finally, d = 6809
- Calculate plaintext
- $v_d(c,1) = v_{6809}(15407,1) = 11111$

There are three data set used three different experiments by changing the size of one variable and fixing the others. Three set of data are different size of public key e, different size of primes and different size of message. All details criteria on each set of data are explained here.

Set 1: Different sizes of public key e are 99, 159, 199, 339,539 digits while the size of p and q are 100 digits. In addition the size of plaintext p is 5 digits.
Set 2: Change P size (20, 80, 100, 160, 200), while p and q size are 100 digits and e size is 19 digits.
Set 3: Using different size of primes p and q; 40, 60,100 digits, where the size of e is 159 and the size of p is 20 digits.

In the following tables explain the decryption computation time for both algorithm the existing algorithm and the proposed algorithm in different situations.

Table 1 shows the decryption computation time on different size of public key. From this table, it is clearly shown that the increasing of the size of public key can also increase the computation time for both algorithms.

Table 1: Decryption computation time on different public keys size

e	p & q	P	d	Existing (Seconds)	DbP (Seconds)
99	100	5	199	47.59	35.24
159	100	5	199	47.69	35.67
199	100	5	199	60.76	36.49
339	100	5	199	85.85	67.61
539	100	5	199	91.28	68.28

Table 2: Decryption computation time on different size of messages

P	p & q	e	d	Existing (Seconds)	DbP (Seconds)
20	100	19	198	89.71	66.86
80	100	19	198	90.21	67.75
100	100	19	198	90.38	67.87
160	100	19	198	90.56	67.98
200	100	19	199	98.16	68.14

Table 3: Decryption computation time on different size of primes

p & q	P	e	d	Existing (Seconds)	DbP (Seconds)
40	20	159	77	12.8	5.51
60	20	159	119	34.73	13.07
80	20	159	159	41.14	20.26
90	20	159	178	45.17	26.20
100	20	159	198	53.01	30.92

From three tables above, the existing algorithm is suffered huge time computation for the decryption process, meanwhile the proposed algorithm is required a small computation time.

The results in Table 1-3 above are based on the running time for each algorithm in C language in Windows 7 Environment, Intel Core^{tm}2 Duo Processor P8700 (2.53 GHz) and 3GBof RAM. All computation times are in seconds.

DISCUSSION

Although the new algorithm is reduced the time consumer which led to speed up the computing time for decryption still requires more computation time. The calculation of private d is done by the calculation of modular equation ed = 1 (mod r), since e is a public key which is known by everyone and r is the Least Common Multiple two Legendre Symbols.

The Least Common Multiple is computed by division by prime method. In this study, the private key computation is possible, because the value of primes p and q is known.

Then, the product of p and q can be use to find Legendre Symbols, Least Common Multiple of two Legendre Symbols. The most important fact is that the Extended Euclidean Algorithm need both public key and N = p*q. This fact is the most crucial in finding the private key for decryption process.

The decryption processes then continues with the computation of private key. Then it continues with decryption process to find the plain text, P.

The computation time of every step in decryption is recorded including the time for calculation Legendre Symbol LS, Lease Common Multiple (LCM) and Extended Euclid Algorithm (EEA).

The proposed algorithm clearly shows that it can compute faster than the existing algorithm. Table 1, 2 and 3 concluded that by implementing Divide-By-Prime (DbP) in Least Common Multiple can speed up the computation of EEA. Furthermore, DbP has an ability to skip some redundant computation found in the existing algorithm.

CONCLUSION

A new algorithm can speed up the process of decryption. It is done by find another algorithm for computing private key. The new enhancement is made by a new approach of finding Legendre symbol and Least Common Multiple.

The comparison between new and existing algorithm shows that the new approach is better in computation speed. It is clearly shown in Table 1, 2 and 3. As a conclusion, the new algorithm makes the decryption process more efficient by reducing the time computing for calculate Legendre Symbols and Least Common Multiple.

ACKNOWLEDGEMENT

The first author would like to express gratitude to Universiti Kebangsaan Malaysia for a research grant UKM-GUP-2011-244.

REFERENCES

Ali, Z.M., M. Othman, M.R.M. Said and M.N. Sulaiman, 2007. Two fast computation algorithms for LUC cryptosystems. Proceeding of The International Conference on Electrical Engineering and Informatics (ICEEI2007), ITB, 2: 434-437.

Ali, Z.M., M. Othman, M.R.M. Said and M.N. Sulaiman, 2009. Computation of private key for luc cryptosystem. Proceedings of the International Conference on Electrical Engineering and Informatics, Aug. 5-7, IEEE Xplore Press, Selangor, pp: 418-422. DOI: 10.1109/ICEEI.2009.5254700

Diffie, W. and M. Hellman, 1976. New directions in cryptography. IEEE Trans. Inform. Theory, 22: 644-654. DOI: 10.1109/TIT.1976.1055638

Horster, P., M. Michels and H. Petersen, 1996. Digital signature schemes based on Lucas functions. Proceedings of the 1st International Conference on Communications and Multimedia Security, (ICCMS' 96), Chapman and Hall, Graz, pp: 178-190.

Knuth, D.E. 1981. The Art of Computer Programming: Seminumerical Algorithms. 2nd Edn., Addison-Wesley, ISBN-10: 0201038226, pp: 688.

Othman, M., E.M. Abulhirat, Z.M. Ali, M.R.M. Said and R. Johari, 2008. A new computation algorithm for a cryptosystem based on lucas functions. J. Comput. Sci., 4: 1056-1060. DOI: 10.3844/jcssp.2008.1056.1060

Rivest, R.L., A. Shamir and L. Adleman, 1978. A method for obtaining digital signatures and public-key cryptosystems. Commun. ACM, 21: 120-126. DOI: 10.1145/359340.359342

Silverman, J.H., 2006. A Friendly Introduction to Number Theory. 3rd Edn., Pearson Prentice Hall, Upper Saddle River, ISBN-10: 0131861379 pp: 434.

Smith, P.J. and M.J.J. Lennon, 1993. LUC: A New Public Key System. 9th IFIP Symposium on Computer Security in E.G Douglas, (SEGD' 93), pp: 103-117.