IOSR Journal of Mathematics (IOSR-JM)
e-ISSN: 2278-5728, p-ISSN: 2319-765X. Volume 12, Issue 1 Ver. I (Jan. - Feb. 2016), PP 09-16
www.iosrjournals.org

Shank's Baby-Step Giant-Step Attack Extended To Discrete Log with Lucas Sequences

P.Anuradha Kameswari[1], T. Surendra[2], B.Ravitheja[3]
[1]*Department of Mathematics/ Andhra University/Visakhapatnam, Andhra Pradesh/India*
[2]*Department of Mathematics/ GITAM University /Visakhapatnam, Andhra Pradesh/India*
[3]*Department of Mathematics/ Andhra University/Visakhapatnam, Andhra Pradesh/India*

Abstract: *The groups of much attention for which the Diffie - Hellman problem may be hard and used securely are the multiplicative group F_p^*, (Z/nZ)* and the group of rational points on an elliptic curve over a finite field. These groups involve large key sizes or expensive arithmetic operations. In this paper we consider the group of Lucas sequences and describe the generalization of discrete log problem to the group of Lucas sequences and adapt the baby-step giant-step algorithm to the generalization. For the computations we implement fast computing methods proposed by Smith.*
Key word: *Discrete Log Problem, Lucas Sequences, Public Key Cryptography.*

I. Introduction

The development of public key cryptography due to Diffie and Hellman is based on using Discrete Logarithm as one way function. The Discrete Logarithm problem in finite fields Fp is based on the fact that F_p^* is cyclic and if g is any generator every element of F_p^* is g^a for some non negative integer a. For the discrete log problem in (Z/nZ)* we consider the generator g of a cyclic subgroup or a primitive root g mod n. More generally the discrete log problem may be discussed in any group with the group law in place of multiplication. In this paper we first discuss the discrete log problem in the finite groups of the form F_p^* or (Z/nZ)* and some of its attacks and then describe the discrete log in group L(Δ, N) of Lucas sequences and look at the possible extension of the Baby-Giant extension.[2, 4]

1.1 Discrete Log Problem

Public key cryptography based on the difficulty of discrete log problem due to Diffie- Hellman is a protocol used for key exchange in a classical cryptosystem and also used in public key cryptosystems like ElGamal cryptosystem.

1.1.1 Definition Let G be a finite group of the form (Z/nZ)*or F_q^* and b be a fixed element of G, if y is any element of G of the form $y = b^x$ for some x. Then the problem of finding the x given y is called the discrete logarithm problem. We write $x = \log_b y$ and x is called discrete logarithm of y to the base b.[11]

1.1.2 Example Let G = Z_{17}^* and take b=3 the generator of Z_{17}^* then the discrete log of 13 to base 3 in Z_{17}^* is x such that $3^x \equiv 13 \pmod{17}$, note for x=4, $3^4 \equiv 13 \pmod{17}$.

1.1.3 Example Let G = Z_p^* for p = 1999 and take b = 3 the generator of G then the discrete log of 1452 to base 3 is x such that $3^x \equiv 1452 \pmod{1999}$. Note in this example computing that x = 789 is difficult but computing $3^{789} \equiv 1452 \pmod{1999}$ is easy by adapting the modular exponentiation method.

1.2 Diffie-Hellman Key Exchange

The Diffie-Hellman protocol works as follows. A and B wish to agree on a common secret key to communicate over an insecure channel. A chooses a large prime p and an integer g such that $2 \le p \le p-2$ and an integer $a \in \{0,1,\ldots\ldots p-2\}$ randomly, then he computes g^a mod p and makes (p, g, g^a) public. B chooses an integer $b \in \{0,1,\ldots\ldots p-2\}$randomly, then he computes g^b mod p and makes (p, g, g^b) public. Then they agree upon the $k = g^{ab}$ mod p as the common shared secret key.

To compute the discrete log there are algorithms likes Trial exponentiation, Shanks Baby -Step Gaint-Step Method, Pollard's ρ method, Pohlig-hellman method, Index calculus method etc. In this paper we describe

Shanks Baby -Step Gaint-Step Method for discrete log in F_p^* or (Z/nZ)* and then describe the implementation of Shanks Baby -Step Gaint-Step Method to discrete log with lucas sequences. This gives a wider crossectional view in the study of attacks on extended discrete logarithms. The earliest method for finding the discrete logarithm x from $\alpha=g^x$(DLP) is to check whether x = 0,1,2,3,… satisfy DLP. If one of these x values satisfy then Discrete Logarithm is found. This is the Trial exponentiation. It needs enumeration of x -1 multiplications and x comparisons in the group and the three elements x, g and g^x need to be stored.[8, 1]

1.2.1 Example The Discrete Logarithm of 3 to the base 5 in (Z/2017Z)* with enumeration of 1029 multiplications modulo 2017 yields x = 1030.

1.3 Shanks Baby-Step Giant-Step Algorithm

A notable development of enumeration is the Shanks Baby-step Giant-Step algorithm. This algorithm needs less number of group operations but storage should be greater. This algorithm is described as follows.

Set $\lceil \sqrt{n} \rceil$ where n is the group order and write the unknown Discrete Logarithm as x = qm + r, $0 \le r < m$. Thus r is the remainder and q is the quotient of the division of x by m. The Baby-step Giant-step algorithm calculates q and r.

$$\text{We have } g^{qm+r} = g^x = \alpha$$
$$\Rightarrow (g^m)^q = \alpha g^{-r}$$

We first compute the set of Baby-steps $B = \{(\alpha g^{-r}, r): 0 \le r < m\}$ and if we find a pair (1, r) in the set B; then $\alpha g^{-r}=1$ (i.e. $\alpha=g^r$), then we can set x = r with the smallest of such x. If such a pair not found, we determine $\delta= g^m$.

Then we check for q = 1, 2, 3, … if the group element δ^q is the first component of an element in B, that is we check if there is a pair (δ^q, r) in B, and when it is true, we have

$$\alpha g^{-r} = \delta^q = g^{qm}$$
$$\Rightarrow \alpha = g^{qm+r}$$

Therefore x = qm + r is the discrete logarithm and the elements of δ^q, q=1,2,3,.... are known as Giant steps. We should compare each δ^q with all first components of the Baby-Step set B. To make this comparison effective, the elements of B are stored in a hash table where the key is the first element.

If we use a hash table, then a constant number of comparisons are sufficient to check whether a group element computed as a giant-step is a first component of a baby-step. Therefore, the following result is easy to verify that the baby-step giant-step algorithm requires $O(\sqrt{|G|})$ multiplications and comparisons in G. It needs storage for $O(\sqrt{|G|})$ elements of G.

Time and space requirements of the Baby-step giant-step algorithm are approximately $\sqrt{|G|}$. If $|G|>2^{160}$,then computing discrete logarithms with the baby-step giant-step algorithm is still infeasible.[8]

1.3.1 Example Determine the Discrete Logarithm of 3 to the base 7 in (Z/17Z)*.

We have $\alpha=g^x$(mod17) and $\lceil m \rceil = \lceil 17 \rceil = 4$.

i.e. $\alpha= 3$, g = 7.

The Baby-step is $B = \{(\alpha g^{-r}, r): 0 \le r < m\}$

$B = \{(3\cdot 7^{-r}, r): 0 \le r < 4\} = \{(3,0),(15,1),(16,2),(1,3)\}$

Therefore $(\alpha g^{-r}, r) = (1, 3)$

$(\alpha g^{-3}, 3) = (1, 3)$

$\alpha g^{-3}=1$

$\Rightarrow 3(7^{-3}) = 1$

$3 \equiv 7^3 \pmod{17}$

x = 3 is the discrete log of 3 to the base 7.

In this example there is no need to go for Giant-step, because solution is found in Baby-step itself.

1.3.2 Example Determine the Discrete Logarithm of 3 to the base 5 in (Z/19Z)*.

We have $\alpha= g^x$(mod19) and $\lceil m \rceil = \lceil 19 \rceil = 4$.

i.e. $\alpha = 3$, $g = 5$.
The Baby-step is $B = \{(\alpha g^{-r}, r): 0 \le r < m\} = \{(3 \cdot 5^{-r}, r): 0 \le r < 4\} = \{(3,0),(12,1),(10,2),(2,3)\}$
Here we do not find a pair (1,r), so we have to determine

$$\delta^q = g^{mq}, \text{ where } q = 1,2,3,\ldots$$
$$\delta^q = 5^{4q} \pmod{19}$$

The Giant-steps are 13, 7,12,…

We have (12,1) in the baby-step set. Therefore, $\alpha g^1 = 12 + 19Z$. Since 12 has been found as the third giant-step, we obtain $g^{3*4} = \alpha g^{-1} \Rightarrow g^{3*4+1} = \alpha$.
$x = 3*4 + 1 = 13$ is the discrete log of 3 to the base 5.

1.3.3 Observation To compute Baby-step set, 4 multiplications mod 19 were necessary, where as to compute giant steps, 3 multiplications mod 19 were necessary.

II. Lucas Sequences

Lucas sequences are widely used in cryptography. There are RSA like Elgamal like cryptosystems based on Lucas sequences. Lucas sequences are also used in the cryptanalytic study.[5]

2.1 Introducing Lucas sequences

2.1.1 Definition Let a and b be two integers and α a root of the polynomial $x^2 - ax + b$ in $Q(\sqrt{\Delta})$ for $\Delta = a^2 - 4b$ a non square, writing $\alpha = \frac{a + \sqrt{\Delta}}{2}$ and its conjugate $\beta = \frac{a - \sqrt{\Delta}}{2}$ we have $\alpha+\beta = a$, $\alpha\beta = b$, $\alpha-\beta = \sqrt{\Delta}$ and the Lucas sequences $\{V_n(a,b)\}$ and $\{U_n(a,b)\}$, $n \ge 0$ are defined as

$$\begin{cases} V_n(a,b) = \alpha^n + \beta^n \\ U_n(a,b) = \dfrac{\alpha^n - \beta^n}{\alpha - \beta} \end{cases}$$

2.2 Lucas Sequences Satisfy The Following Relations

1. $V_{2n}(a,b) = (V_n(a,b))^2 - 2b^n$
2. $V_{2n-1}(a,b) = V_n(a,b)V_{n-1}(a,b) - ab^{n-1}$
3. $V_{2n+1}(a,b) = a(V_n(a,b))^2 - bV_n(a,b)V_{n-1}(a,b) - ab^n$
4. $(V_n(a,b))^2 = \Delta(U_n(a,b))^2 + 4b^n$
5. $V_{km}(a,b) = V_k(V_m(a,b), b^k)$
6. $U_{km}(a,b) = U_k(V_m(a,b), b^k)U_m(a,b)$
7. $V_{k+m}(a,b) = \frac{1}{2}(V_k(a,b)V_m(a,b) + \Delta U_k(a,b)U_m(a,b))$
8. $U_{k+m}(a,b) = \frac{1}{2}(U_k(a,b)V_m(a,b) + \Delta U_m(a,b) V_k(a,b))$

2.3 Modular Computations With The Lucas Sequences

2.3.1 Definition [3,5] Let $N = p_1^{e_1} \ldots p_r^{e_r}$, p_i's odd primes and define the function

$$S(N) = \operatorname{lcm}\left\{ p_i^{e_i - 1}\left(p_i - \left(\frac{\Delta}{p_i}\right)\right) \right\}_{i=1}^{r}.$$

The following theorem is an analogue to Euler Fermat theorem.

2.3.2 Theorem For $N = p_1^{e_1} \ldots p_r^{e_r}$ and p_i does not divide Δ, then

$$\begin{cases} U_{S(N)t}(a,1) = U_0(a,1) \equiv 0 \bmod N \\ V_{S(N)t}(a,1) = V_0(a,1) \equiv 2 \bmod N \quad \textit{for some integer } t \textit{ and } p_i \textit{ does not divide } 2\Delta. \end{cases}$$

2.4 Group Structure on Lucas Sequence

Let N be an integer with gcd(Δ,N)=1 and write V_k *for* $V_k(a,1)$, consider the set L(Δ,N)={(V_k, U_k) mod N: k≥0} then for all (V_k,U_k), (V_m,U_m) ∈L(Δ,N) we define an operation ⋆ on L(Δ,N) as follows $(V_k,U_k)\star(V_m,U_m)=(V_{k+m},U_{k+m})\text{mod}N$.

2.4.1 Theorem L(Δ,N) forms an abelian group with respect to operation ⋆ defined as $(V_k,U_k)\star(V_m,U_m)=(V_{k+m},U_{k+m})$

Proof

1. ⋆ is closed

L(Δ,N) is closed w.r.t ⋆ follows by definition

2.⋆ is associative

For any (V_k, U_k), (V_m, U_m), (V_l, U_l)∈ L(Δ, N) we have by the definition

$(V_{k+m},U_{k+m})\star(V_l,U_l)=(V_{(k+m)+l},U_{(k+m)+l})$

$(V_{k+m},U_{k+m})\star(V_l,U_l)=(V_{k+(m+l)},U_{k+(m+l)})$

$=(V_k,U_k)\star(V_{m+l},U_{m+l})\text{mod}N$

Therefore L(Δ,N) is associative.

3.(V_0,U_0) is the identity

For any (V_k, U_k)∈L(Δ,N) we have (V_0, U_0)∈L(Δ,N) such that

$(V_k,U_k)\star(V_0,U_0)=(V_{k+0},U_{k+0})=(V_k,U_k)$

$=(V_0,U_0)\star(V_k,U_k)$

Therefore (V_0,U_0) is the Identity.

4. Inverse of (V_k,U_k)

For any (V_k, U_k)∈L(Δ,N), we have ($V_{S(N)-1)k}$, $U_{S(N)-1)k}$)∈L(Δ,N), is the inverse of (V_k,U_k) and

$(V_k,U_k)\star(V_{(S(N)-1)k},U_{(S(N)-1)k})=(V_{k+(S(N)-1)k},U_{k+(S(N)-1)k})\text{mod}N$

$=(V_{kS(N)},U_{kS(N)})\text{mod}N$

$= (2,0)$ by theorem 2.3.2

$=(V_0,U_0)\text{mod}N$

Therefore $(V_{(s(N)-1)k},U_{(s(N)-1)k})$ is the inverse of (V_k,U_k)

5.⋆ is commutative

$(V_m,U_m)\star(V_n,U_n)=(V_{m+n},U_{m+n})$

$=(V_{n+m},U_{n+m})$

$=(V_n,U_m)\star(V_m,U_n)$

Therefore L(Δ,N) is an abelian group.

III. Fast Computation Method For V_e

We describe the fast computation method to compute V_e suggested by P.Smith for Lucas sequences, this method directly leads to the computation of V_e with no ambiguity of adding or doubling at each stage right from V_1 by using the above recursive formulas. [3,12] We give an algorithm for this fast computation in this section.

For any integer m, we have the binary expression given as $e=\sum_{t=0}^{t} x_i 2^{t-i}, x_0=1, x_i=0$ or 1, for i ≥ 0.

Let $e_k = \sum_{i=0}^{k} x_i 2^{k-i}$, for $0 \le k \le t$, then e_t=e, e_0=1.

3.1 Theorem $e_{k+1} = \begin{cases} 2e_k & if\ x_{k+1} = 0 \\ 2e_k + 1 & if\ x_{k+1} = 1 \end{cases}$

3.2 Remark $e_{k+1} + 1 = \begin{cases} 2e_k + 1 & if\ x_{k+1} = 0 \\ 2(2e_k + 1) & if\ x_{k+1} = 1 \end{cases}$

$$e_{k+1} - 1 = \begin{cases} 2e_k - 1 & if\ x_{k+1} = 0 \\ 2e_k & if\ x_{k+1} = 1. \end{cases}$$

3.3 Remark V_e are computed by evaluating V_{e_k} for k=0,1,...,t by using recursive formulas for V_{2e_r+1},V_{2e_r-1} and V_{2e_r} for r ≤k.

We give in the following an algorithm for fast computation method for computing the Lucas sequences $V_e(a,1)$ mod N. Let a be an integer modulo N, we initialize with $V_1(a,1)=a$ to obtain the result $V_e(a,1)$.

3.4 Algorithm

Step 1: Write the binary expression of e as $e = \sum_{i=0}^{t} x_i 2^{t-i}$, x_0=1.

Step 2: Initialize the values $V_c = V_1(a,1)$

$$V_{c_+} = V_1^2 - 2 \quad V_{c_-} = 2$$

Step 3: For i from 0 to t do

c←2c
c-←2c−1
$V_{2c} \leftarrow V_c^2 - 2$
$V_{2c-1} \leftarrow V_c V_{c-} - V_1$
$V_c \leftarrow V_{2c}$
$V_{c-} \leftarrow V_{2c-1}$
if x_i=1
then c←2c+1
c-←2c
$V_{2c+1} \leftarrow V_1 V_c^2 - V_c V_{c-} - V_1$
$V_{2c} \leftarrow V_c^2 - 2$
$V_c \leftarrow V_{2c+1}$
$V_{c-} \leftarrow V_{2c}$
else c←2c
c− ← 2c−1

IV. Discrete Log with Lucas Sequences

Let L(Δ, N) be the group of Lucas sequences with $D=a^2-4$ mod n, then for any (α,β)∈L(Δ, N) if $\alpha=V_m(a)$ given α and a to find m is the Discrete Log problem of Lucas sequences and m may be called the discrete log of α to the base a.[6]

4.1 Extended Diffie Hellmann Protocol With Lucas Sequences

L(Δ, N) be a group of Lucas sequences for $\Delta=a^2-4$ mod N.

1. A and B generate m, n $\in Z_{S(N)}$ the secret keys respectively and calculate their respective public keys P_A *and* P_B as $P_A=V_m(a)$ and $P_B=V_n(a)$.
2. A and B exchange their public keys (a,P_A) and (a,P_B) .

3. A receives P_B from B and calculates $K = V_m(P_B)=V_m(V_n(a))=V_{mn}(a)$.
4. B receives P_A from A and calculates $K = V_n(P_A)=V_n(V_m(a))=V_{nm}(a)=V_{mn}(a)$.
5. Then A and B agree upon $K=V_{mn}(a)$ as the shared secret key.

4.2 To Extend Shank's Baby-step Giant-Step attack on Discrete Log problem with Lucas Sequences

The attacker can gather the shared secret key if he computes *m* or *n* from the public information P_A or P_B, so we employ Shank's Baby-step Giant-step method to the Discrete Log problem with Lucas sequences as follows.

In this method we compute *m* from the public key $\alpha V_m(a)$. Let $n=S(N)$ be the order of the group $L(\Delta,N)$ and $t=\sqrt{n}$ for $n=S(N)$. Now by division algorithm for *t* and *m* we write the unknown discrete logarithm as $m=qt+r$, for $0\le r<t$. Thus *r* is the remainder and *q* is the quotient of the division of *m* by *t*. The Baby-step Giant-step algorithm calculates *q* and *r*.

We have $V_m(a) = \alpha$

$\Rightarrow V_{tq+r}(a) = \alpha$, now by group operation å on L(Δ, N)

we have $V_{tq+r}(a) = V_{tq}$ å $V_r(a)=\alpha$

$\Rightarrow V_{tq} = \alpha$ å $(V_r(a))^{-1}$

We first compute the set of Baby-Steps B = $(\alpha$ å $(V_r(a))^{-1}$, r): $0\le r<t$ and if we find a pair (V_0,r) in B, if exists then

$$a\star(V_r(a))^{-1}\equiv V_0 \bmod N$$
$$\Rightarrow \alpha\equiv V_0 \text{ å } (V_r(a)) \bmod N$$
$$\Rightarrow \alpha\equiv (V_r(a)) \bmod N$$

then take m=r and $V_m(a)=\alpha= V_r(a)$ and if such pair in B is not found we determine $\delta=V_t(a)$ and evaluate $V_q(\delta)$ for all $q=1,2,3,\ldots$ until for some *q* there is a pair $(V_q(\delta), r)$ in B for some *r*, then we have

$$V_q(\delta)=\alpha \text{ å } (V_r(a))^{-1}$$
$$\Rightarrow \alpha=V_q(\delta) \text{ å } (V_r(a))$$
$$=V_q(V_t(a))\star V_r(a)$$
$$=V_{qt}(a)\star V_r(a)$$
$$=V_{qt+r}(a)$$
$$=V_m(a)$$

therefore $m=qt+r$ is the discrete log of α to base *a*.

4.2.1 Example Let $N=17$ and $a=5$ then $\Delta=a^2-4 = 5^2-4 =21$ and $S(N)=N-(\Delta/N)=17-1=16$. In $L(\Delta,N)$ given $\alpha=12$ to find the discrete log of 12 to the base 5. We have $t=\sqrt{n}=\sqrt{16}=4$ and by division algorithm m = tq+r, $0\le r < t$ i.e. $m = 4q + r$; $r = 0,1,2$ *or* 3. To find $V_m(a)=a$, we have to find $V_{4q+r}(a)=\alpha$, in this context we compute the pairs $(\alpha$ å $V_r(a)^{-1}, r)$ in B for $r=0,1,2,3$; and $a=5$. We have $(V_r(a))^{-1}=V_{(S(N)-1)r}$ and using the fast computing method

$$V_0(5)=2;(V_0(5))^{-1}=V_0(5)=2$$
$$V_1(5)=5;(V_1(5))^{-1}=V_{15}(5)=5$$
$$V_2(5)=6;(V_2(5))^{-1}=V_{14}(5)=6$$
$$V_3(5)=8;(V_3(5))^{-1}=V_{13}(5)=8$$

$$\Rightarrow B=\{(12 \text{ å } (V_0(5))^{-1},0);\ (12 \text{ å } (V_1(5))^{-1},1);\ (12 \text{ å } (V_2(5))^{-1},2);(12 \text{ å } (V_3(5))^{-1},3)\}$$
$$=\{(12 \text{ å } (V_0(5)),0);\ (12 \text{ å } (V_{15}(5)),1);\ (12 \text{ å } (V_{14}(5)),2);\ (12 \text{ å } (V_{13}(5)),3)\}.$$

Now we can compute these values using the product $V_k\star V_m=V_{k+m}$ in the group $(L(\Delta,N);\star)$ and apply the formula

$$V_{k+m}(a,b)=\frac{1}{2}(V_k(a,b)V_m(a,b)+\Delta U_k(a,b)U_m(a,b))$$

by computing the corresponding U_i's by using formula $V_n^2(a,b) = \Delta U_n^2(a,b) + 4b^n$.

$$V_n^2(a,b) = \Delta U_n^2(a,b) + 4b^n.$$

So, for a=5 we have Δ =4; and

$$U_0^2(5) = \frac{V_0^2(5)-4}{\Delta} = \frac{2^2-4}{4} = 0(mod17) = 0$$

$$U_{15}^2(5) = \frac{V_{15}^2(5)-4}{\Delta} = \frac{5^2-4}{4} = \frac{21}{4} = 1(mod17) = 1$$

$$U_{14}^2(5) = \frac{V_{14}^2(5)-4}{\Delta} = \frac{6^2-4}{4} = \frac{32}{4} = \frac{15}{4} = 8(\text{mod}17) = 8$$

$$U_{13}^2(5) = \frac{V_{13}^2(5)-4}{\Delta} = \frac{8^2-4}{4} = \frac{60}{4} = 15(\text{mod}17) = 15.$$

Therefore $U_0(5)=0; U_{15}(5)=1; U_{14}(5)=5; U_{13}(5)=7$, now from the formula of $V_{k+m}(a,b)$, *wehave*

$$12 \text{ å } V_0(5) = 12 \text{ å } 2 = \frac{1}{2}(12.2 + 4.1.0) = 12(\text{mod}17) = 12$$

$$12 \text{ å } V_{15}(5) = 12 \text{ å } 5 = \frac{1}{2}(12.5 + 4.1.1) = 32(\text{mod}17) = 15$$

$$12 \text{ å } V_{14}(5) = 12 \text{ å } 6 = \frac{1}{2}(12.6 + 4.1.5) = 46(\text{mod}17) = 12$$

$$12 \text{ å } V_{13}(5) = 12 \text{ å } 8 = \frac{1}{2}(12.8 + 4.1.7) = 62(\text{mod}17) = 11$$

$\Rightarrow$B={$\alpha\star(V_r^{-1}$, r): r=0,1,2,3}={(12,0), (15,1), (12,2), (11,3)}

The pair (V_0,r) does not exists in B, so we take $\delta=V_t(a)= V_4(5)=0$ and evaluate $V_q(\delta)$ for all q=1,2,3,… until for some q there is a pair $(V_q(\delta), r)$ in B for some r, we have $V_1(0) = 0(mod17) = 0; V_2(0) = -2(mod17) = 15$ for $q = 2, r = 1$.

Therefore $m = tq + r = 4.2 + 1 = 9$, also $V_9(5) = 12$.

V. Conclusion

The generalization of discrete log problem to Lucas sequences is obtaining $a^{\pm k}$ from $v_k\ (a,1) \bmod p$ as a root of the irreducible polynomial $x^2-v_k(a,1)x+1$ over F_p of degree 2 and retrieving ±k from $a^{\pm k}$ i.e. it is a discrete log problem in F_p^2. The discrete log problem with Lucas sequences is originally considered as equivalent to problems with Dickson polynomials. In this paper we depicted the discrete log problem on Lucas sequences as generalization of discrete log problem to group of Lucas sequences and adapt the Shank's Baby-step Giant-step attack and described the fast computation method with an algorithm that may be used in the computations of Lucas sequences.

References

Journal papers

[1]. W. Diffie and M.E. Hellman, "New directions in cryptography", IEEE Trans. Inform. Theory, IT-22(6):644-654, 1976.

[2]. T. ElGamal, "A public key cryptosystem and a signature scheme based on discrete logarithms", IEEE Trans. Inform. Theory, 31(4):469-472, 1985.

[3]. Marc Gysin, "The Discrete Logarithm Problem for Lucas Sequences and a New Class of Weak RSA Moduli", The University of Wollongong, NSW 2522.

[4]. Peter J. Smith and Michael J.J. Lennon, "A New Public Key System", LUC Partners, Auckland UniServices Ltd, The University of Auckland, Private Bag 92019m Auckland, New Zealand.

[5]. B.Ravitheja, "RSA-Like Cryptosystem based on the Lucas Sequence", dissertation, 2015.

[6]. P.J.Smith, G.J.J.Lennon, "LUC: a new public key cryptosystem", Ninth IFIP Sympoiusm on Computer Science Security, Elsevier Science Publictions (1993)103-117.

Books

[7]. Tom M. Apostol, "Introduction to Analytic Number Theory", Springer-Verlag, New York Inc.
[8]. J. Buchmann, "Introduction to cryptography", Springer-Verlag 2001.
[9]. P.B.Bhattacharya, S.K. Jain, S.R. Nagpaul "Basic Abstract Algebra", 2nd edition, Cambridge University press.
[10]. A.J.Menezes, P.C. van Oorschot, and S.A. Vanstone, "Hand book of Applied Cryptography." CRC Press Series on Discrete Mathematics and its Applications. CRC Press, Boca Raton, FL, 1997. With a foreword by Ronald L. Rivest.
[11]. Neal Koblitz "A course in number theory and cryptography ISBN 3-578071-8,SPIN 10893308 ".
[12]. Song Y. Yan, "Number Theory for computing", 2nd edition, Springer, ISBN: 3-540-43072-5.

Proceedings in Papers

[13]. G. Maze, C. Monico, and J. Rosenthal, "A public key cryptosystem based on actions by semigroups". In the proceedings of the 2002 IEEE International Symposium on Information Theory, page XY, Lausanne, Switzerland, 2002.