Palestine Journal of Mathematics

Vol. 1(2) (2012) , 148–152

A Public-Key Cryptosystem Based on Lucas Sequences

Lhoussain El Fadil

Communicated by Ayman Badawi

MSC2010 Classification: 11T71.

Keywords : Lucas sequences, Public-Key Crypto-system, Semantic security.

Supported by the Spanish ministry (Ref : SB 2006-0128) at CRM of Barcelona

Abstract. Based on Lucas functions, an improved version of Diffie-hellman key distribution, El Gamal public key crypto-system scheme and El Gamal signature scheme are proposed, together with an implementation and computational cost. The security relies on the difficulty of factoring an RSA integer and on the difficulty of computing the discrete logarithm.

Introduction

In [1], Diffie and Hellman introduced a practical solution to the key distribution problem, allowing two parties, Alice and Bob never met, to share a secret key by exchanging information over an open channel. In [2], El Gamal used Diffie-Hellman ideas to design a crypto-system whose security is based on the difficulty of solving the discrete logarithm problem. In [3], It was suggested that linear sequences could be used instead of the standard RSA.

In this paper, based on Lucas sequences, an improved of Diffie-hellman key distribution,El Gamal public key crypto-system and El Gamal digital signature were proposed. This considerably reduces the computation cost of these methods. The security relies on the difficulty of factoring an RSA integer. In section 1, an investigation of cryptographic properties of Lucas sequences, and a computational method to evaluate the k^{th} term of a Lucas sequence are given. In section 2, two cryptographic applications are given, their security and computational cost were analyzed.

1 Lucas sequences

In this section, the main cryptographic properties of Lucas sequences are studied. A computational method to evaluate the k^{th} term are given, together with an analysis of its computational cost.

Throughout this section, p is a prime integer, $a \in \mathbb{Z}$, $f(X) = X^2 - aX + 1\,[p]$ a polynomial in $F_p[X]$, α a root of $f(X)$ in a splitting field of $f(X)$ and $s(a)$ the characteristic sequence generated by a modulo p, where $F_p := \mathbb{Z}/p\mathbb{Z}$ is the finite field of p element. Denote $\bar{}$ the reduction modulo p, $A = F_p[X]/(f(X))$ and $\alpha = \bar{X}$ the class of X modulo the principal ideal of $F_p[X]$ generated by $f(X)$. For every $x \in A$, let l_x be the linear map of A defined by $l_x(y) = xy$, $T(x) = Tr(l_x)$ and $N(x) = det(l_x)$ the trace and norm of x, where $det(l_x)$ is the determinant of the linear map l_x, and $Tr(l_x)$ is its trace. Define a sequence $s(a)$ as follows : $s_k(a) = T(\alpha^k)$. Since $f(\alpha) = 0$ and the map trace is linear, it follows that $s_{k+2}(a) = as_{k+1}(a) - s_k(a)$ modulo p. So, $s(a)$ is a second order linear sequence (Lucas sequence), called the characteristic sequence generated by a.

Remark. Let l_k be the endomorphism of A defined by $l_k(x) = \alpha^k x$, and M_k its matrix with respect to the basis $(1, \alpha)$. Then $M_0 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$, $M_1 = \begin{pmatrix} 0 & -1 \\ 1 & \bar{a} \end{pmatrix}$, and then $s_0(a) = 2$ and $s_1(a) = \bar{a}$.

1.1 Cryptographic properties

The cryptographic applications of Lucas sequences are listed in [3]. For the commodity of the reader, we present some of these results in a more accessible form and with simplified proofs.

Lemma 1. 1. For every integer k, $s_k(a) = \alpha^k + \alpha^{-k}$ and $s_k(a) = s_{-k}(a)$.

Proof. Let K be a splitting field of $f(X)$. Since $f(X)$, the characteristic polynomial of M_1, splits in K, there exists an invertible matrix P in $M_2(K)$ and $x \in K$ such that $M_1 = PTP^{-1}$, where $T = \begin{pmatrix} \alpha & x \\ 0 & \alpha^{-1} \end{pmatrix}$. Let k be an integer. As $M_k = M_1^k$, we have $M_k = PT^kP^{-1}$, where $T^k = \begin{pmatrix} \alpha^k & x_k \\ 0 & \alpha^{-k} \end{pmatrix}$ and $x_k \in K$. Therefore,

$s_k(a) = Tr(M_k) = \alpha^k + \alpha^{-k}$. □

Corollary 1. 2. For every integer k, let $f_k(X) = X^2 - s_k(a)X + 1$. Then $f_k(X) = (X - \alpha^k)(X - \alpha^{-k})$.

Indeed, $s_k(a) = \alpha^k + \alpha^{-k}$ and $\alpha^k\alpha^{-k} = 1$.

Lemma 1. 3. For every integers k and e, $s_e(s_k(a)) = s_{ke}(a)$.

Proof. From Corollary 1.2, the roots of the polynomial $f_k(X)$ are α^k and α^{-k}. So, $s_e(s_k(a)) = (\alpha^k)^e + (\alpha^{-k})^e = T(\alpha^{ke}) = s_{ke}(a)$. □

Lemma 1. 4. $\pi = p^2 - 1$ is a period of $s(a)$. Especially, if p does not divide $a^2 - 4$, then $p - \epsilon_p$ is the period, where $\epsilon_p = (\frac{a^2-4}{p})$ is the Legendre symbol.

Proof. Since α is an element of A of norm 1, α is an invertible element of A. Let $\triangle = a^2 - 4$ be the disciminant of $f(X)$.

(i) p divides $(a^2 - 4)$. Then $a = \mp 2$ modulo p. If $a = 2$ modulo p, then $s_k(a) = 2$. If $a = -2$ modulo p, then $s_{2k}(a) = 2$ and $s_{2k+1}(a) = -2$.

(ii) If $(\frac{a^2-4}{p}) = 1$, then $f(X)$ splits in F_p, and $A \simeq F_p \times F_p$. Hence the exponent of the multiplicative group A^* is $p - 1$. Thus, $\alpha^{p-1} = 1$.

(iii) If $(\frac{a^2-4}{p}) = -1$, then $A \simeq F_{p^2}$ and $N(\alpha) = \alpha^{p+1} = 1$. Let $x \in N$ be the period of $s(a)$. Then $\alpha^x = 1$. Since$(\frac{a^2-4}{p}) = -1$, $\alpha \notin F_p^*$, and then $x \geq p + 1$.

Corollary 1. 5. For every integer e such that $gcd(e, \pi) = 1$, the map

$$\begin{array}{cccc} Luc_e: & F_p & \longrightarrow & F_p \\ & a & \longrightarrow & s_e(a) \end{array}$$ is a one-one correspondence.

Indeed, since $gcd(e, p^2 - 1) = 1$, let d be the inverse of e modulo π. Then there exists an integer k such that $de = 1 + k\pi$. Hence $s_d(s_e(a)) = s_{de}(a) = s_{1+k\pi}(a) = s_1(a) = a\,[p]$. □

Lemma 1. 6. Let $a \in \mathbb{Z}$ such that p does not divide $a^2 - 4$. Let $\epsilon = (\frac{a^2-4}{p})$, $e \in N$ such that $Gcd(e, \pi) = 1$ and $b = s_e(a)$, where $\pi = p - \epsilon$. Then p does not divide $b^2 - 4$ $(\frac{a^2-4}{p}) = (\frac{b^2-4}{p})$

Proof. Since $a = s_d(b)\,[p]$, where d is the inverse of e modulo $p^2 - 1$, it suffices to show that if $(\frac{a^2-4}{p}) = 1$, then $(\frac{b^2-4}{p}) = 1$ too. Assume that $(\frac{a^2-4}{p}) = 1$, then $\alpha \in F_p$, and then $\alpha^e \in F_p$. Thus, $f_b(X) = X^2 - bX + 1$ splits in F_p $(\alpha^e + \alpha^{-e} = b\,[p])$, i.e., $(\frac{b^2-4}{p}) = 1$. □

1.2 Computational Method and Cost

Lemma 1. 7. $\left\{\begin{array}{lrcl} i) & s_{2n}(a) & = & s_n(a)^2 - 2, \\ ii) & s_{2n+1}(a) & = & s_n(a)s_{n+1}(a) - a \end{array}\right.$

Proof. Let n and m be two integers. $s_n(a)s_m(a) = (\alpha^n + \alpha^{-n})(\alpha^m + \alpha^{-m})$
$= (\alpha^{n+m} + \alpha^{-n-m}) + (\alpha^{n-m} + \alpha_2^{-n+m}) = s_{n+m}(a) + s_{n-m}(a)$. Therefore, $s_{n+m}(a) = s_n(a)s_m(a) - s_{n-m}(a)$
In particular, we have i) and ii). □

Let $k = 2^r m$, where m is an odd integer. To compute $s_k(a)$, first we compute $s_m(a)$, then $s_{2m}(a) = (s_m(a))^2 - 2$, then $s_{4m}(a) = (s_{2m}(a))^2 - 2$,...,$s_k(a) = s_{2^{r-1}m}(a)^2 - 2$. Then to compute $s_k(a)$, we need r multiplications modulo p and we need $s_m(a)$. Let $m = \sum_{i=0}^{l-1} k_i 2^{l-1-i}$. For every $0 \leq i < l - 1$, let $f_{i+1} = 2f_i + k_{i+1}$ and $f_0 = k_0$. Then $f_{l-1} = k$. For $0 \leq i < l - 1$ and assume that, $s_{f_{i-1}}(a)$ and $s_{f_{i-1}+1}(a)$ are computed. Then

if $k_i = 0$, then $\left\{\begin{array}{rcl} s_{f_i}(a) & = & s_{2f_{i-1}}(a) = (s_{f_{i-1}}(a))^2 - 2 \\ s_{f_i+1}(a) & = & s_{2f_{i-1}+1}(a) = s_{f_{i-1}}(a)s_{f_{i-1}+1}(a) - a \end{array}\right.$

if $k_i = 1$, then $\left\{\begin{array}{rcl} s_{f_i}(a) & = & s_{2f_{i-1}+1}(a) = s_{f_{i-1}}(a)s_{f_{i-1}+1}(a) - a \\ s_{f_i+1}(a) & = & s_{2(f_{i-1}+1)}(a) = (s_{f_{i-1}+1}(a))^2 - 2 \end{array}\right.$

Computational Algorithm.

In put $k = 2^r \sum_{i=0}^{l-1} k_i 2^i$ and a, where $k_0 \neq 0$ and $k_{l-1} \neq 0$.
Out put s_k.

```
Algorithm
$s_0 = 2$, $s_1 = a$,
for $i$ from 0 to $l - 1$ do
    if $k_i = 0$ then $s_1 = s_1 s_0 - a$, $s_0 = s_0^2 - 2$
    else then $s_0 = s_1 s_0 - a$, $s_1 = s_1^2 - 2$
End
return $(s_0)$.
$s = s_0$, for $i$ from 1 to $r$ do $s = s^2 - 2$.
End
return $(s)$.
```

This method ensures that s_k can be computed in about the same length of time as the k^{th} power is computed in the RSA method. But in the computation of $s_m(a)$, having to compute *two numbers at each stage does slow* the computation down a little, but there are optimizations in the calculation which mean that the total amount of computation is only about *half less* than the amount needed for the RSA system. Therefore, to compute $s_k(a)$, the total number of multiplications modulo p is $log_2(k)$.

2 Main results

In this section we describe some applications of Lucas sequences, in more details : Lucas Diffie-Hellman, Lucas El Gamal encryption scheme and Lucas El Gamal signature scheme.

Let $n = pq$ be an RSA integer, $a \in N$ such that $(\frac{a}{p}) = (\frac{a}{q}) = -1$ and $s(a)$ the Lucas sequence generated by a defined by : $s_0(a) = 2\,[n]$, $s_1(a) = a\,[n]$ and $s_{k+2}(a) = as_{k+1}(a) - s_k(a)\,[n]$. Let and $f(X) = X^2 - aX + 1$ modulo n, $A = \mathbb{Z}_n[X]/(f(X))$ and $\alpha = \bar{X}$ the class of X modulo the principal ideal $(f(X))$ (we have $s_k(a) = \alpha^k + \alpha^{-k} = T(\alpha^k)$).

2.1 Lucas Diffie-Hellman

Let a be an integer. Suppose that Alice and Bob, who both have access to the Lucas sequences public key data (n, a), want to agree on a shared secret key K_{AB}.

(i) User Alice selects $0 < x_A < n$ as her private key. She then computes $y_A = s_{x_A}(a)$ as her public key from the system public parameters (n, a).

(ii) User Bob selects $0 < x_B < n$ as his private key. He then computes $y_B = s_{x_B}(a)$ as his public key from the system public parameters (n, a).

(iii) Key-Distribution Phase : $K_{AB} = s_{x_A}(y_B) = s_{x_B}(y_A)$ is their common secret key.

Remarks

(i) In [3], it was given a Diffie-Hellman scheme based on lucas functions defined in F_q. Here, it is the same version but with Lucas sequences modulo an RSA integer.

(ii) $K_{AB} = s_{x_A x_B}(a)$.

(iii) In each exchange session, the computational cost of each user is $2log_2(n)$.

(iv) From [5], the security of Lucas sequences is polynomial-time equivalent to the generalized discrete logarithm problem over F_p. Thus, the security level of this scheme is at least the security level of the standard Diffie-Hellman scheme.

2.2 Lucas El Gamal

Now, we explain our version of the public key system. It is based on El Gamal system, which is defined by Lucas sequences.

Suppose that Bob is the owner of the Lucas public key data (p, q, a).
Bob selects a small integer e such that $Gcd(d, (p^2 - 1)(q^2 - 1)) = 1$ and a secret integer $0 < x \leq n$. He computes d the inverse of e modulo $(p^2 - 1)(q^2 - 1)$, $y = s_x(a)$ and makes public (e, y).
Given Bob's public data (n, a, e, y), Alice can encrypt a message m, where $0 \leq m < n$, intended for Bob using the following Lucas version of the El Gamal encryption scheme :

Algorithm

(i) Public key : (n, a, y, e)

(ii) Private key : (p, q, d, x).

(iii) Encryption : For a message $0 \leq m < n$, Alice chooses a (secret) random number $0 < k < n$, and sends Bob the cipher $c = (c_1, c_2)$, where $c_1 = s_k(a)$, $c_2 = K + s_e(m)$ and $K = s_k(y)$.

(iv) Decryption : For a cipher $c = (c_1, c_2)$, Bob computes $K = s_x(c_1)$, and then $m = s_d(c_2 - K)$, where (p, q, d, x) is its private key.

Note that

(1) All computations are performed in $\mathbb{Z}_n$.
(2) Let $\epsilon_p = (\frac{m^2-4}{p})$, $\epsilon_q = (\frac{m^2-4}{q})$ and $\pi = Lcm(p - \epsilon_p, q - \epsilon_q)$. Then $s_x(c_1) = s_x(s_k(a)) = s_{xk}(a) = s_k(s_x(a)) = s_k(y) = K$, and then $c_2 - K = s_e(m)$.
Since $ed = 1$ modulo π, there exists an integer l such that $ed = 1 + l\pi$.Thus, $s_d(c_2 - K) = s_d(s_e(m)) = s_{ed}(m) = s_{1+l(p^2-1)(q^2-1)}(m) = s_1(m) = m$ modulo p.

Security analysis

Definition 2. 8. Given a cipher $C = K + s_e(m)$ $(m \in \{m_0, m_1\})$, where K is randomly chosen modulo n, the problem of deciding whether m equals m_0 or m_1 is called decision problem based on Lucas sequences.
If any analyzer can not decide which one of m_0 or m_1 is corresponding to the cipher c in polynomial time, then the encryption scheme is semantical secure.

First, since k is randomly chosen, K is considered random too. On the other hand, since $n = pq$ is an RSA integer, it is very hard to factorise the period $T = (p - \epsilon_p)(q - \epsilon_q)$. It follows that, the decision problem based on Lucas sequences is intractable: Given $c = s_e(m)$ it is very hard to calculate neither e nor m (this problem is equivalent to the discrete logarithm problem (see [5])). Consequently, the proposed Lucas El Gamal scheme is semantical secure.

Computational cost

As in the standard RSA public key system, Bob chooses a small integer e and Alice chooses a relatively small integer k such that the computational cost for evaluating $s_k(a)$ and $s_e(m)$ are low. For example $e = 5$, we need 3 multiplications modulo n for computing $s_3(m)$, $log_2(k)$ multiplications modulo n for computing $s_k(a)$, i.e., totally, we need $3 + log_2(k)$ multiplications modulo n for enciphering.
For deciphering, once d and y are computed, we need $log_2(x)$ multiplications modulo n for computing $K = s_x(c_1)$, and $log_2(d)$ multiplications modulo n for computing $s_d(c_2 - K)$. As $d < n^2$, we need $log_2(n)$ multiplications modulo n for deciphering. Totally, we need $4log_2(n)$ on average.

2.3 Lucas El Gamal signature

We now explain our version of El Gamal signature scheme, defined by Lucas sequences.
First,we need the following lemma :

Lemma 2. 9. Let $s(a)$ be a Lucas sequence generated by a, m, n and k three integers such that $m + n = k$. Then $s_k^2(a) + s_{2m}(a) + s_{2n}(a) = s_m(a)s_n(a)s_k(a)$.

Proof. In the proof of Lemma 1.6, we have shown that $s_{m+n}(a) = s_m(a)s_n(a) - s_{m-n}(a)$. Thus, $s_k(a) = s_m(a)s_n(a) - s_{m-n}(a)$, and then $s_k^2(a) = s_m(a)s_n(a)s_k(a) - s_{m+n}(a)s_{m-n}(a) = s_m(a)s_n(a)s_k(a) - (s_{2n}(a) + s_{2m}(a))$. Therefore, $s_k^2(a) + (s_{2n}(a) + s_{2m}(a)) = s_m(a)s_n(a)s_k(a)$. □

Suppose Alice is the owner of the Lucas public key data (p, q, a). She computes the secrets $\epsilon_p = (\frac{a^2-4}{p})$ and $\epsilon_q = (\frac{a^2-4}{q})$. She selects an integer x, $1 \leq x < (p-1)(q-1)$, computes $y = s_x(a)$ and makes public (n, a, y).
Given Alice's public data (n, a, y), Alice can sign a message m, where $0 \leq m < n$, intended for Bob using the following Lucas version of El Gamal signature scheme :

(i) Alice selects at random an integer k such that $gcd(k, (p - \epsilon_p)(q - \epsilon_q)) = 1$ and she computes $r = s_k(a)$.

(ii) She computes $s = k^{-1}(m - xr)$ modulo $(p - \epsilon_p)(q - \epsilon_q)$, where k^{-1} is the inverse of k modulo $(p - \epsilon_p)(q - \epsilon_q)$.

(iii) (r, S) is the signature for the message m, where $S = s_s(r)$.

To verify the authenticity of the signature, any verifier can check if

$$(E) \,:\, s_m^2(a) + s_r^2(y) + S^2 - 4 = S s_r(y) s_m(a)$$

.
Indeed, since $m = xr + ks$, it follows that $s_m^2(a) + s_{2rx}(a) + s_{2sk}(a) = s_{rx}(a)s_{sk}(a)s_m(a) = s_r(y)s_s(r)s_m(a)$.

Security analysis.

This scheme hides the linear equation $ks + xr = m$ modulo T. Thus even if the owner uses twice the same random k, the analyzer can not calculate the secret x. On the other hand, the security of Lucas sequences is polynomial-time equivalent to the generalized discrete logarithm problem over F_p. It follows that the security level of this scheme is at least the security level of El Gamal digital signature.

Computational Cost.

To sign the message m, we will compute : $s_k(a)$, k^{-1} modulo $(p - \epsilon_p)(q - \epsilon_q)$ and $k^{-1}(m - xr)$. Thus we need, $ln_2(n) + 3,5$ multiplications to sign the message m such that $0 \leq m < n$.
To verify, we will compute : $s_r(y)$, $s_s(r)$, $s_m(a)$, $s_{2r}(y)$, $s_{2s}(r)$ and $s_m^2(a)$. As $s_{2r}(y) = s_r^2(y) - 2$ and $s_{2s}(r) = s_s^2(r) - 2$, to verify the authenticity of the signature, we need $2ln_2(n) + 3$ multiplications.

References

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

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

[3] P. Smith et M. J. J. Lennon, LUC : A new public key system. In Proc. of the Ninth IFIP Int. Symp. on Computer Security, 103-117 (1993).

[4] A.K. Lenstra, E.R. Verheul, The XTR public key system, Proceedings of Crypto 2000, LNCS 1880, 1-19(2000).

[5] Chi-Sung Laih, Fu-Kuan Tu, Wen-Chun Tai, On the security of the Lucas function, Information Processing Letters 53, 243-247 (1995).

[6] Douglas R. Stinson, Cryptography Theory and Practice, Third edition 2006, Chapman, Hall/CRC, Taylor and Francis Group.

Author information

Lhoussain El Fadil, Dep. of Math., College of Sciences, King Khalid University, Abha, Saudi Arabia.
E-mail: lhouelfadil@hotmail.com

Received: March 12, 2012

Accepted: June 4, 2012