A PUBLIC-KEY CRYPTOSYSTEM BASED ON SECOND ORDER LINEAR SEQUENCES

L. EL FADIL

Abstract. Based on Lucas functions, an improved version of the Diffie-Hellman distribution key scheme and to the ElGamal public key cryptosystem 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], ElGamal used Diffie-Hellman ideas to designe a cryptosystem whose security is based on the difficulty of solving the discrete logarithm problem. In [3, 5, 6], It was suggested that linear sequences can be used instead of the standard RSA.

In this paper, based on second order linear sequences (Lucas functions), an improved version of the Diffie-Hellman distribution key and to the ElGamal public key cryptosystem method are 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 the cryptographic properties of second order linear sequences, and a computational method to evaluate the k^{th} term of a second order linear sequence are given. In section 2, two cryptographic applications are given, their security and computational cost are analysed.

1. Second order linear sequences

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

Key words and phrases. Second order linear sequence (Lucas functions), Public-Key Cryptosystem.

computational cost.

Let $f(X) = X^2 - aX + 1$ be a polynomial in $\mathbb{F}[X]$, where $\mathbb{F}$ is a field. Denote $A = \mathbb{F}[X]/(f(X))$ and $\alpha = \bar{X}$ the class of X modulo the principal ideal of $\mathbb{F}[X]$ 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 $s_{k+2}(a) = as_{k+1}(a) - s_k(a)$. So, $s(a)$ is a second order linear 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 $M_0 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$, $M_1 = \begin{pmatrix} 0 & -1 \\ 1 & a \end{pmatrix}$, and then $s_0(a) = 2$ and $s_1(a) = a$.

1.1. **Cryptographic properties.** The cryptographic applications of Lucas sequences are listed in [3, 4]. For the commodity of the reader, we present some of these results in a more accessible form and with simplified proofs.

Lemma 1. 1. *Let $f(X) = X^2 - aX + 1$ be a polynomial in $\mathbb{F}[X]$, α_1 and α_2 the roots of $f(X)$ in a splitting field of $f(X)$. Then for all integer k, $s_k(a) = \alpha_1^k + \alpha_2^k$, and $s_k(a) = s_{-k}(a)$.*

Proof. Let K be a splitting field of $f(X)$. Since $f(X)$ is the characteristic polynomial of M_1, and splits in K, there exists an invertible matrix P in $M_2(K)$ and $x \in K$ such that $M_1 = PTP^{-1}$, where $T = \begin{pmatrix} \alpha_1 & x \\ 0 & \alpha_2 \end{pmatrix}$. Let k be an integer. As $M_k = M_1^k$, then $M_k = PT^kP^{-1}$ and $T^k = \begin{pmatrix} \alpha_1^k & x_k \\ 0 & \alpha_2^k \end{pmatrix}$, where $x_k \in K$. Therefore, $s_k(a) = Tr(M_k) = \alpha_1^k + \alpha_2^k$.
Let k be an integer. Since $\alpha_1\alpha_2 = 1$, $s_{-k}(a) = \alpha_1^{-k} + \alpha_2^{-k} = \alpha_2^k + \alpha_1^k = s_k(a)$. ■

Corollary 1. 2. *Let $f(X) = X^2 - aX + 1$ be a polynomial in $\mathbb{F}[X]$, α_1 and α_2 be the roots of $f(X)$ in a splitting field of $f(X)$. For every integer k, let $f_k(X) = X^2 - s_k(a)X + 1$. Then $f_k(X) = (X - \alpha_1^k)(X - \alpha_2^k)$.*

Indeed, $s_k(a) = \alpha_1^k + \alpha_2^k$ and $\alpha_1^k\alpha_2^k = 1$. ■

Lemma 1. 3. *Let $f(X) = X^2 - aX + 1$ be a polynomial in $\mathbb{F}[X]$, and $s(a)$ be the characteristic sequence generated by a. Then 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_1^k and α_2^k. So, $s_e(s_k(a)) = (\alpha_1^k)^e + (\alpha_2^k)^e = T(\alpha^{ke}) = s_{ke}(a)$. ■

Lemma 1. 4. *Let $f(X) = X^2 - aX + 1$ be a polynomial in $\mathbb{F}_p[X]$, and $s(a)$ the characteristic sequence generated by a. Then $\pi = p^2 - 1$ is a period of $s(a)$.*

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 $f(X)$. Denote $(\frac{d}{p})$ the Legendre symbol, where d is an integer such that p does not divide d. Then there are three cases :

(1) p divides $(a^2 - 4)$. Then $a = \mp 2$ modulo p. If $a = 2$ modulo p, then for every k, $s_k(a) = 2$. If $a = -2$ modulo p, then for every k, $s_{2k}(a) = 2$ and $s_{2k+1}(a) = -2$, and then 2 is the period of $s(a)$.
(2) If $(\frac{a^2-4}{p}) = 1$, then $f(X)$ splits in $\mathbb{F}_p$, and $\alpha_1 \neq \alpha_2$. Thus $A \simeq \mathbb{F}_p \times \mathbb{F}_p$. Hence the exponent of the multiplicative group A^* is $p - 1$. So, $\alpha^{p-1} = 1$.
(3) If $(\frac{a^2-4}{p}) = -1$, then $A \simeq \mathbb{F}_{p^2}$. Let σ be a primitive element of the multiplicative group $\mathbb{F}_{p^2}^*$. Set $\alpha = \sigma^k$. Then $N(\alpha) = \sigma^{k(p+1)} = 1$. Therefore, $p^2 - 1$ divides $k(p + 1)$, i.e., $p - 1$ divides k, and then there exists an integer l such that $\alpha = \sigma^{l(p-1)}$. So, $\alpha^{p+1} = 1$.

Consequently, $\alpha^\pi = 1$. Let k and m be two integers, $s_{m+k\pi}(a) = T(\alpha^{m+k\pi}) = T(\alpha^m(\alpha^\pi)^k) = T(\alpha^m) = s_m(a)$. Hence π is a period of the sequence $s(a)$.■

Corollary 1. 5. *Let $f(X) = X^2 - aX + 1$ be a polynomial in $\mathbb{F}_p[X]$, and $s(a)$ the characteristic sequence generated by a. Then for every integer e such that $gcd(e, \pi) = 1$, the map* $\begin{array}{rccc} Luc_e : & \mathbb{F}_p & \longrightarrow & \mathbb{F}_p \\ & a & \longrightarrow & s_e(a) \end{array}$ *is a one-one correspondence.*

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

1.2. **Computational Method and Cost.**

Lemma 1. 6. *Let $f(X) = X^2 - aX + 1$ be a polynomial in $\mathbb{F}_p[X]$, and $s_k(a)$ the characteristic sequence generated by a. Then*

$$\begin{cases} i) & s_{2n}(a) & = & s_n(a)^2 - 2, \\ ii) & s_{2n+1}(a) & = & s_n(a)s_{n+1}(a) - a \end{cases}$$

Proof. Let n and m be two integers. $s_n(a)s_m(a) = (\alpha_1^n + \alpha_2^n)(\alpha_1^m + \alpha_2^m)$ $= (\alpha_1^{n+m} + \alpha_2^{n+m}) + (\alpha_1^{n-m} + \alpha_2^{n-m}) = s_{n+m}(a) + s_{n-m}(a)$.
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 $\begin{cases} s_{f_i}(a) & = & s_{2f_{i-1}}(a) = (s_{f_{i-1}}(a))^2 - 2 \\ s_{f_i+1}(a) & = & s_{2f_{i-1}+1}(a) = s_{f_{i-1}}(a)s_{f_{i-1}+1}(a) - a \end{cases}$

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

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 more 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 result

In this section we describe some applications of Lucas functions, in more details : Diffie-Hellman distribution key method and the ElGamal encryption scheme.

Let $n = pq$ be an RSA integer, $f(X) = X^2 - aX + 1$ a polynomial in $\mathbb{Z}_n = \mathbb{Z}/n\mathbb{Z}$, $A = \mathbb{Z}_n[X]/(f(X))$ and $\alpha = \bar{X}$ the class of X modulo the principal ideal $(f(X))$. Let $s(a)$ be the characteristic sequence generated by a, defined in $\mathbb{Z}_n$ by : $s_k(a) = T(\alpha^k)$.

Lemma 2. 7. *Let $\pi = (p^2-1)(p^2-1)$. Then $\alpha^\pi = 1$ modulo n. In particular, $\alpha^\pi = 1$ is a period of $s(a)$.*

Indeed, $\alpha^{\alpha^\pi} = (\alpha^{(p^2-1)})^{(q^2-1)} = 1$ modulo p and $\alpha^{\alpha^\pi} = (\alpha^{(q^2-1)})^{(p^2-1)} = 1$ modulo q. ∎

2.1. **Lucas Diffie-Hellman.** Let a be an integer. Suppose that Alice and Bob, who both have access to the Lucas function public key data (n, a), want to agree on a shared secret key K_{AB}. recall that in [3], there is a Diffie-Hellman scheme based on lucas functions defined in $\mathbb{F}_q$. Here, we give the same version but with lucas functions on $\mathbb{Z}_n$.

(1) User Alice selects $0 < x_A \leq n$ as her private key. She then computes $y_A = s_{x_A}(a)$ as her public key from the system public key n and $f(X) = X^2 - aX + 1$.
(2) User Bob selects $0 < x_B \leq n$ as his private key. He then computes $y_B = s_{x_B}(a)$ as his public key from the system public key n and $f(X) = X^2 - aX + 1$.
(3) Key-Distribution Phase : $K_{AB} = s_{x_A}(y_B) = s_{x_B}(y_A)$ is their common secret key.

Remarks (1) $K_{AB} = s_{x_A x_B}(a)$.
(2) In each exchange session, the computational cost of each user is $2log_2(n)$.
(3) If an attacker tries to compute Alice's private key x from her public key $y = s_x(a)$, a polynomial $f_y(X) = X^2 - yX + 1$ is formed. According to Lemma 1.2, α_1^x and α_1^{-x} are the roots of $f_y(X)$. As a result, once α_1^x and α_1 are known, solving the exponent x is equivalent to solving the discret logarithm problem in $\mathbb{Z}_n$. Since $\mathbb{Z}_n \simeq \mathbb{F}_p \times \mathbb{F}_q$, let $\alpha_1 = (\alpha_{11}, \alpha_{12})$, then $\alpha_1^x = (\alpha_{11}^x, \alpha_{12}^x)$. Consequently, solving the discret logarithm problem in $\mathbb{Z}_n$ is much harder than solving the discret logarithm problem in $\mathbf{F}_p$, and then this method improves that presented in [3].

2.2. **Lucas ElGamal.** We now explain our version of the public key system. It is based on ElGamal system, which is defined by Lucas functions.

Suppose 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 $0 < x \leq n$. Computes d the inverse of e modulo $(p^2-1)(q^2-1)$ and $y = s_x(a)$, 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 ElGamal encryption scheme :
Algorithm

(1) Public key : (n, a, y, e)
(2) Private key : (p, q, d, x).

(3) Encryption : For a message $0 \leq m < n$, Alice chooses a (secret) random number $0 < k < n$, and she sends Bob the ciphertext $c = (c_1, c_2)$, where $c_1 = s_k(a)$ and $c_2 = K + s_e(m)$, where $K = s_k(y)$.
(4) Decryption : For a ciphertext $c = (c_1, c_2)$, 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) $s_x(c_1) = s_x(s_k(a)) = s_{xk}(a) = s_k(s_x(a)) = s_k(y) = K$, and then $c_2 - K = s_e(m)$.
Since $ed = 1$ modulo $(p^2-1)(q^2-1)$, there exists an integer l such that $ed = 1 + l(p^2-1)(q^2-1)$. As $(p^2-1)(q^2-1)$ is a period of $s(m)$, $s_d(c_2 - K) = s_d(s_e(m)) = s_{ed}(m) = s_{1+l(p^2-1)(q^2-1)}(m) = s_1(m) = m$.

2.3. **Security.** If an attacker tries to compute m from $c = (c_1, c_2)$ and (a, y, e, n), he will compute K and d, i.e., he will compute the secret parameters k and d. The first one is equivalent to braking standard the El Gamal scheme. For the second one, because the properties of Lucas functions mirror those of exponentiation, public key and private key processes can be developed in an exactly analogous manner to the RSA system, this enables us to prove that any successful attack on this system would give a successful attack on the standard RSA system [3]. Thus the security of the method relies on the difficulty of factoring an RSA integer and on the difficulty of computing the discrete logarithm in $\mathbb{Z}_n$.

2.4. **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.

Acknowledgments
I would like to thanks the "Centre de Recerca Matematica" of Barcelona for their extraordinary hospitality and facilities for doing this work. As well as the professor E. Nart for his valuable comments and suggestions.

References

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

[2] T. ElGamal, "A public key cryptosystem and a signature scheme based on discrete logarithms," IEEE Trans. Inform. Theory, vol. IT- 31, pp. 469-472, July 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, p. 103-117, 1993.

[4] D. H. Lehmer, An extended theory of lucas functions, Annals of Maths, 31 (1930), pp 419-448.

[5] G. Castagnos, An efficient probabilistic public-key cryptosystem over quadratic fields quotients., Finite Fields Appl. 13 (2007), no. 3, 563-576.

[6] G. Gong et L. Harn : Public-Key Cryptosystems Based on Cubic Finite Field Extensions. In IEEE Trans. Inform. Theory, vol. 45, p. 2601-2605, 1999.

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

Faculté Polydisciplinaire de Ouarzazat, PO. Box 638, Ouarzazat-Morocco
E-mail address: lhouelfadil@hotmail.com