A PUBLIC KEY ENCRYPTION BASED ON THIRD ORDER LINEAR SEQUENCES

L. EL FADIL

Abstract. Based on third order linear sequences, an improvement version of the Diffie-Hellman distribution key scheme and 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.

1. 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], It was suggested that the linear sequences can be used instead of the standard RSA.

In this paper, based on the third order linear sequences, an improved version of the Diffie-Hellman distribution key and El Gamal 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 2, an investigation of the cryptographic properties of third order linear sequences, and a computational method and cost to evaluate the k^{th} term of a third order linear sequence are given. In section 3, two cryptographic applications, their security and computational cost are analysed.

2. Third order linear sequences

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

Throughout this paper, p is a prime integer, $F_p := \mathbb{Z}/p\mathbb{Z}$ the finite field of p elements, $a \in F_p$ and $f(X) = X^3 - aX^2 + aX - 1$ a polynomial in $F_p[X]$. Denote $A = F_p[X]/(f(X))$ 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

Key words and phrases. Third order linear sequence, Public-Key encryption, Quartic field extensions.

$l_x(y) = xy$, $T(x) = Tr(l_x)$ 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 trace map is linear, it follows that $s_{k+3}(a) = as_{k+2}(a) - as_{k+1}(a) + s_k(a)$. So, $s(a)$ is a third 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 $(1, \alpha, \alpha^2)$. So,

$$M_0 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, M_1 = \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & -a \\ 0 & 1 & a \end{pmatrix} \text{ and } M_2 = \begin{pmatrix} 0 & 0 & a \\ 0 & -a & 1-a^2 \\ 1 & a & a^2-a \end{pmatrix}.$$

It follows that: $s_0(a) = 3$, $s_1(a) = a$, and $s_2(a) = a^2 - 2a$.

2.1. **Cryptographic properties.** Let $f(X) = X^3 - aX^2 + aX - 1 \in F_p[X]$, 1, α_1 and α_2 its roots in a splinting field of $f(X)$, and let $s(a)$ be the characteristic sequence generated by a. Then we have the following properties:

(1) For every integer k, $s_k(a) = 1 + \alpha_1^k + \alpha_2^k$ and $s_k(a) = s_{-k}(a)$.
(2) For every integer k,
let $f_k(X) = X^3 - s_k(a)X^2 + s_k(a)X - 1$. Then $f_k(X) = (X-1)(X-\alpha_1^k)(X-\alpha_2^k)$.
(3) For every integers k and e, $s_e(s_k(a)) = s_{ke}(a)$.
(4) $T = (p^2 - 1)$ is a period of $s(a)$.

Proof.

(1) Since $f(X)$ is the characteristic polynomial of M_1, which splits in a splinting field K of $f(X)$, and 1, α_1 and α_2 are the roots of $f(X)$ in K, there exists an invertible matrix P in $M_3(K)$ such that $M_1 = PTP^{-1}$, where $T = \begin{pmatrix} 1 & x & y \\ 0 & \alpha_1 & z \\ 0 & 0 & \alpha_2 \end{pmatrix}$, where $(x, y, z) \in K^3$. Let k be an integer. As $M_k = M_1^k$, $M_k = PT^kP^{-1}$, where $T^k = \begin{pmatrix} 1 & x_k & y_k \\ 0 & \alpha_1^k & z_k \\ 0 & 0 & \alpha_2^k \end{pmatrix}$, and $(x_k, y_k, z_k) \in K^3$.
Therefore, $s_k(a) = Tr(\alpha^k) = 1 + \alpha_1^k + \alpha_2^k$.
Let k be an integer. Since $\alpha_1\alpha_2 = 1$, $s_{-k}(a) = 1 + \alpha_1^{-k} + \alpha_2^{-k} = 1 + \alpha_2^k + \alpha_1^k = s_k(a)$.
(2) Since, $s_k(a) = 1 + \alpha_1^k + \alpha_2^k$, $1\alpha_1^k + 1\alpha_2^k + \alpha_1^k\alpha_2^k = \alpha_1^k + \alpha_2^k + 1 = s_k(a)$, and $1.\alpha_1^k\alpha_2^k = (1.\alpha_1\alpha_2)^k = 1$.
(3) The roots of the polynomial $f_k(X)$ are 1, α_1^k, α_2^k. So, $s_e(s_k(a)) = 1 + (\alpha_1^k)^e + (\alpha_2^k)^e = s_{ke}(a)$.
(4) Since, α is an element of A of norm 1, α is an invertible element of A. Set $f(X) = (X-1)(X^2 - (a-1)X + 1)$ and $\triangle = (a-1)^2 - 4$ be the disciminant of $P(X) = X^2 - (a-1)X + 1$. Then there are three cases:

(a) p divides $(a-1)^2-4$. Then $a=3$ or $a=-1$ modulo p. If $a=3$ modulo p, then for every k, $s_k(a)=3$. If $a=-1$ modulo p, then for every k, $s_{2k}(a)=3$ and $s_{2k+1}(a)=-1$, and then 2 is the period of $s(a)$.

(b) If $(\frac{(a-1)^2-4}{p})=1$, then $f(X)$ splits in F_p, and $\alpha_1\neq\alpha_2$. Thus $1,\alpha_1,\alpha_2$ are pairwise distinct, and then $A\simeq F_p\times F_p\times F_p$. Hence the exponent of the multiplicative group A^* is $p-1$. So, $\alpha^{p-1}=1$.

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

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 2. 1. *For every integer e such that $gcd(e,p^2-1)=1$, the map*

$$\begin{array}{rccc} Luc_e: & F_p & \longrightarrow & F_p \\ & a & \longrightarrow & s_e(a) \end{array}$$

is a one-one correspondence.

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

2.2. **Computational Method and Cost.**

Lemma 2. 2. *Let $s_k(a)$ be the characteristic sequence generated by a. Then*

$$\begin{cases} i) & s_{2n}(a) = s_n(a)(s_n(a)-2), \\ ii) & s_{2n+1}(a) = s_n(a)s_{n+1}(a)-s_{n+1}(a)-s_n(a)-a+3 \end{cases}$$

Proof.

i) Since $\alpha_1^n\alpha_2^n=1$, $s_n(a)^2=(1+\alpha_1^n+\alpha_2^n)^2=(1+\alpha_1^{2n}+\alpha_2^{2n})+2(\alpha_1^n+\alpha_2^n)+2=s_{2n}(a)+2s_n(a)$.

ii) $s_n(a)s_{n+1}(a)=(1+\alpha_1^n+\alpha_2^n)(1+\alpha_1^{n+1}+\alpha_2^{n+1})=(1+\alpha_1^{2n+1}+\alpha_2^{2n+1})+(\alpha_1^n+\alpha_2^n)+(\alpha_1^{n+1}+\alpha_2^{n+1})+(\alpha_1+\alpha_2)=s_{2n+1}(a)+s_n(a)+s_{n+1}(a)+a-3$. ■

Let $k=2^rm$, where m is an odd integer. To compute $s_k(a)$, first we compute $s_m(a)$, then $s_{2m}(a)=s_m(a)(s_m(a)-2)$, then $s_{4m}(a)=s_{2m}(a)(s_{2m}(a)-2)$,...,$s_k(a)=s_{2^{r-1}m}(a)(s_{2^{r-1}m}(a)-2)$. To compute $s_k(a)$, we need r multiplications modulo p and we need too $s_m(a)$. Let $m=\sum_{i=0}^{l-1}k_i2^{l-1-i}$. 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 $0\leq i<l-1$, assume that $s_{f_{i-1}}(a)$ and $s_{f_{i-1}+1}(a)$ are computed. So, if $k_i=0$, then $s_{f_i}(a)=s_{2f_{i-1}}(a)=$

$s_{f_{i-1}}(a)(s_{f_{i-1}}(a)-2)$ and $s_{f_i+1}(a) = s_{2f_{i-1}+1}(a) = s_{f_{i-1}}(a)s_{f_{i-1}+1}(a) - s_{f_{i-1}}(a) - s_{f_{i-1}+1}(a) - a + 3$ if $k_i = 1$, then

$$s_{f_i}(a) = s_{2f_{i-1}+1}(a) = s_{f_{i-1}}(a)s_{f_{i-1}+1}(a) - s_{f_{i-1}}(a) - s_{f_{i-1}+1}(a) - a + 3$$

and $s_{f_i+1}(a) = s_{2(f_{i-1}+1)}(a) = s_{f_{i-1}+1}(a)(s_{f_{i-1}+1}(a) - 2)$.

Algorithm of computational.

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

```
Algorithm
Begin
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 − s_1 − s_0 − a + 3,
  s_0 = s_0^2 − 2s_0 else then
  s_0 = s_1 s_0 − s_1 − s_0 − a + 3, s_1 = s_1^2 − 2s_1
end for
s = s_0,
for i from 1 to r do s = s^2 − 2s.
end for
return (s).
End.
```

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 $log_2(k)$.

3. Main result

In this section we describe some applications of third order linear sequences, in more details : Diffie-Hellman distribution key and ElGamal public key encryption scheme.

Let $n = pq$ be an RSA integer, $f(X) = X^3 - aX^2 + aX - 1$ 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 3. 3. *$\alpha^{(p^2-1)(q^2-1)} = 1$ modulo n. In particular, $(p^2-1)(q^2-1)$ is a period of $s(a)$.*

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

3.1. **Diffie-Hellman distribution key.** Suppose that Alice and Bob who both have access to the Lucas function public data (n, a) want to agree on a shared secret key K_{AB}. This can be done using the following Lucas version of the Diffie-Hellman protocol:

(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 $f(X) = X^3 - aX^2 + aX - 1$.
(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 $f(X) = X^3 - aX^2 + aX - 1$.
(3) Key-Distribution Phase: $K_{AB} = s_{x_A}(y_B) = s_{x_B}(y_A)$ is their common secret key.

Remark. That

(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 try to compute the Alice's private key x from her public key $y = s_x(a)$, a polynomial $f_y(X) = X^2 - (y-1)X + 1$ is formed. According to the 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 discrete logarithm in $\mathbb{Z}_n$, which is much harder than solving the discrete logarithm in $\mathbf{F}_p$, and then this method improves that presented in [3].

3.2. **ElGamal public key encryption.** We now explain our improvement version of the public key system. It is based on ElGamal system, which is defined by third order linear sequences.

Suppose Bob is the owner public data (p, q, a). Bob selects a small integer e such that $gcd(e, (p^2-1)(q^2-1)) = 1$ 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 made public (e, y) as his public key.

Given the Bob public data (n, a, e, y), Alice can encrypt a message m, where $0 \leq m < n$, intended for Bob using the following version of the ElGamal encryption scheme as follows:

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 computational are 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 $c_2 - K = s_e(m)$. (3) 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 $s(m)$, $s_d(c_2 - K) = s_d(s_e(m)) = s_{ed}(m) = s_{1+l(p^2-1)(q^2-1)}(m) = s_1(m) = m$.

Security analysis. Because the properties of third order linear sequences 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 RSA and ElGamal systems [3]. Thus the security of the method relies on the difficulty of factoring an RSA integer and on the difficulty of solving the discrete logarithm in $\mathbb{Z}_n$.

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 $s_3(m)$, in average $log_2(k)$ multiplications modulo n for computing $s_k(a)$. Totally, $4+log_2(k)$ 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, $3log_2(n)$ multiplications modulo n for deciphering. Totally, we need $4log_2(n)$ on average.

Acknowledgments

I would like to thanks the "Centre de Recerca Matemtica" 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. El Gamal, "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, pp. 103–117, 1993.

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

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

[6] Chi-Sung Laih, Fu-Kuan Tu, Wen-Chun Tai, "On the security of the Lucas function", Information Processing Letters 53(1995), pp. 243–247.
[7] Douglas R. Stinson, "Cryptography Theory and Practice", Third edition 2006, Chapman, Hall/CRC, Taylor and Francis Group.

Polydisciplinary Faculty of Ouarzazate
Ouarzazate-Morocco
E-mail address: lhouelfadil@hotmail.com