Applied Mathematics and Computation 209 (2009) 52–56

Contents lists available at ScienceDirect

Applied Mathematics and Computation

journal homepage: www.elsevier.com/locate/amc

Aitken and Neville inverse interpolation methods for the Lucas logarithm problem

E.C. Laskari [a,c,*], G.C. Meletiou [b,c], M.N. Vrahatis [a,c]

[a] Computational Intelligence Laboratory, Department of Mathematics, University of Patras, GR-26110 Patras, Greece
[b] A.T.E.I. of Epirus, P.O. Box 110, GR-47100 Arta, Greece
[c] University of Patras Artificial Intelligence Research Center (UPAIRC), University of Patras, GR-26110 Patras, Greece

ARTICLE INFO

Keywords:
Lucas function
Lucas logarithm problem
Inverse interpolation
Aitken interpolation
Neville interpolation
Discrete logarithm

ABSTRACT

The Lucas function is a recently proposed one-way function used in public key cryptography. The security of cryptosystems based on the Lucas function relies on the difficulty of solving the Lucas logarithm problem. In this paper, the Lucas logarithm problem is studied using interpolation techniques. In particular, the inverse Aitken and the inverse Neville interpolation methods are applied to values of the Lucas sequence to obtain a polynomial that interpolates the Lucas logarithm. The results indicate that in all the considered instances of the problem a polynomial of low degree that interpolates the desired values exists.

1. Introduction

The essential part of public key cryptography is the use of *one-way* functions. These functions can be easily computed in one direction, but their inverses are hard to compute. One-way functions are usually derived from computationally difficult number theoretic problems, like the discrete logarithm problem and the factorization problem, among others. The best known and most widely used one-way function is the modular exponentiation which constitutes the basis for various public key cryptosystems [1,2].

Smith and Lennon in [3,4] proposed a new one-way function to design public key cryptosystems, namely the Lucas function, which is defined as follows. Let p be an odd prime and let $\mathbb{F}_p$ be the finite field of order p. For a fixed element $m \in \mathbb{F}_p$, consider the following second-order linear recurrence relation:

$$\begin{aligned} &V_0(m) = 2, \quad V_1(m) = m, \\ &V_t(m) = mV_{t-1}(m) - V_{t-2}(m), \quad t \geqslant 2. \end{aligned} \tag{1}$$

The sequence $\{V_t(m)\}, t = 0, 1, \ldots$, is called *Lucas sequence generated by* m and the mapping,

$$t \mapsto V_t(m), \quad t \geqslant 0,$$

is called *Lucas function*.

It was shown in [3] that $V_t(m) = \mu^t + \mu^{-t}, t \geqslant 0$, where μ and μ^{-1} are the roots of the characteristic polynomial of Eq. (1), i.e. $f(X) = X^2 - mX + 1$. The roots μ and μ^{-1} are given by the formulae,

[*] Corresponding author. Address: Computational Intelligence Laboratory, Department of Mathematics, University of Patras, GR-26110 Patras, Greece.
E-mail addresses: elena@math.upatras.gr (E.C. Laskari), gmelet@teiep.gr (G.C. Meletiou), vrahatis@math.upatras.gr (M.N. Vrahatis).

0096-3003/$ - see front matter © 2008 Elsevier Inc. All rights reserved.
doi:10.1016/j.amc.2008.06.014

$$\mu = \frac{m + \sqrt{m^2 - 4}}{2}, \quad \mu^{-1} = \frac{m - \sqrt{m^2 - 4}}{2}.$$

If $m^2 - 4$ is a quadratic residue modulo p or zero, then both roots are in $\mathbb{F}_p$, otherwise both roots are in $\mathbb{F}_{p^2} \setminus \mathbb{F}_p$.

The security of cryptosystems based on the Lucas function relies on the difficulty of addressing the *Lucas logarithm problem to the base m* which is defined as follows. Given a prime p, any $m \in \mathbb{F}_p$ and $z \in \{V_t(m)\}$, where $\{V_t(m)\}$, with $t \geqslant 0$, is the Lucas sequence generated by m, find an integer x such that $V_x(m) = z$.

The importance of the Lucas logarithm for modern cryptography is well-known [3–7]. Furthermore, the Lucas function can be viewed as a generalization of the exponentiation function as it is shown in [5] to be a special form of the Dickson polynomial. This fact triggered research on the computational equivalence of the Lucas logarithm and the discrete logarithm problems [6,7] and it was recently shown in [7] that the security of the Lucas function is polynomial-time equivalent to the generalized discrete logarithm problems. The hardness of the discrete logarithm problem is supported by a large amount of research work [8–18] and as shown in [8,16,17] there are no low degree interpolation polynomials of the discrete logarithm for a large set of given data. Similar work on polynomial representations and lower bounds on the degree of interpolation polynomials of a function related to the Lucas problem is presented in [5]. Specifically, the exact formula of a polynomial representing the Lucas logarithm is deduced and lower bounds on the degree of interpolation polynomials of the Lucas logarithm for subsets of given data are proved.

Finding a polynomial of low degree or low sparcity which represents the Lucas logarithm comprises a method for its fast computation. To this end, in this contribution we study the Lucas logarithm problem through a different approach. Specifically, we employ the Aitken and Neville inverse interpolation methods to address the problem at hand. An important characteristic of these methods is that they internally determine the domain of the interpolated function. The results indicate that the computational cost for addressing the problem in this setting remains high, but the appearance of low degree polynomials that interpolate the Lucas logarithm value constitutes an interesting finding.

The rest of the paper is organized as follows. In Section 2, for completeness purposes, the Aitken and Neville inverse interpolation methods are briefly presented. In Section 3 the presented interpolation methods are applied for the Lucas function, the experimental setup is described and results are reported. Finally, conclusions are derived in Section 4.

2. The Aitken and Neville inverse interpolation methods

Aitken and Neville interpolation methods, as well as the Lagrange method, are well-known and they are considered as the state-of-the-art for interpolation of functions over real numbers [19]. In contrast to the Lagrange method, Aitken and Neville methods are constructive in a way that permits the addition of a new interpolation point directly and with low computational cost. Thus, interpolation is initially applied to a small number of points and unless the required polynomial is found, new interpolation points are added sequentially to the previously obtained polynomial with low cost. This advantage over the Lagrange interpolation method and the fact that Aitken's and Neville's interpolation formulae can be applied in any field, has motivated the investigation of their performance over finite fields [20].

Consider a function $f(x)$ defined on a field $\mathbb{F}, x_i \in \mathbb{F}, i = 0, 1, \ldots, n$, be mutually different interpolation points and denote $f_i = f(x_i)$. Since in several cases, the values $f_i, i = 0, 1, \ldots, n$ of a function $y = f(x)$ at the corresponding points x_i, $i = 0, 1, \ldots, n$, are given and the point x^*, such that $f(x^*) = y^*$, is required, inverse interpolation methods for the function f

Table 1
Instances of the problems considered for the case of roots of the characteristic polynomial of the Lucas sequence in $\mathbb{F}_p$ (Roots in $\mathbb{F}_p$) and in $\mathbb{F}_{p^2}$ (Roots in $\mathbb{F}_p^2$)

# Problem	p	g	Roots in $\mathbb{F}_p$		Roots in $\mathbb{F}_p^2$	
			m	z	m	z
1	101	2	3	82	4	14
2	599	7	3	365	5	23
3	1759	6	5	1701	4	14
4	2003	5	4	14	3	7
5	2411	6	3	7	6	34
6	2801	3	3	7	4	14
7	3001	14	4	2261	5	23
8	3313	10	4	902	5	23
9	3631	15	3	3419	4	14
10	4001	3	3	2990	4	14
11	4751	19	5	2178	11	119
12	5003	2	5	23	3	7
13	5881	31	4	3362	9	79
14	6007	3	5	1681	3	7
15	7001	3	3	5739	4	14
16	7841	12	3	6924	4	14
17	8011	14	3	7	5	23
18	8929	11	4	545	6	34
19	9001	7	4	8403	5	23
20	10,007	5	4	2627	5	23

are used. Both Aitken and Neville interpolation methods can be applied for the inverse interpolation problem by simply exchanging x_i and y_i in the corresponding formulae [19]. Thus, the formula of the inverse Aitken interpolation method is,

$$P_{0,1,\dots,m,i}(y) = \frac{1}{(y_i - y_m)} \begin{vmatrix} P_{0,1,\dots,m}(y) & y_m - y \\ P_{0,1,\dots,m-1,i}(y) & y_i - y \end{vmatrix}, \quad \text{for } \begin{cases} m = 0, \dots, n-1, \\ i = (m+1), \dots, n, \end{cases}$$

where, in general, $P_{0,1,\dots,k}$ denotes the Aitken polynomial that interpolates all points $y_0, y_1, \dots, y_k$, while the corresponding inverse Neville interpolation formula is,

$$P_{i,i+1,\dots,i+m}(y) = \frac{1}{(y_{i+m} - y_i)} \begin{vmatrix} P_{i,i+1,\dots,i+m-1}(y) & y_i - y \\ P_{i+1,i+2,\dots,i+m}(y) & y_{i+m} - y \end{vmatrix}, \quad \text{for } \begin{cases} m = 1, \dots, n, \\ i = 0, \dots, (n-m), \end{cases}$$

where, in general, $P_{i,i+1,\dots,i+k}$ denotes the Neville polynomial that interpolates all points $y_i, y_{i+1}, \dots, y_{i+k}$.

3. Inverse Aitken and Neville interpolation for the Lucas logarithm

In this contribution we study the inverse Aitken and the inverse Neville interpolation methods over a shifted Lucas function, i.e. the function

$$f(t) = V_t(m) - z, \quad t \geqslant 0,$$

Table 2
Results of the inverse Aitken and inverse Neville methods over 100 independent experiments for the case of roots in the field $\mathbb{F}_p$

Problem	# Verifications					# Points Used					Pol. Degree				
	Mean	Med	SD	Min	Max	Mean	Med	SD	Min	Max	Mean	Med	SD	Min	Max
Aitken															
1	46.97	30.00	51.12	0.00	326.00	32.47	28.50	23.48	1.00	97.00	4.73	4.00	3.77	0.00	14.00
2	134.58	105.00	126.94	0.00	661.00	112.55	96.50	85.81	1.00	407.00	7.66	5.50	6.34	0.00	31.00
3	464.38	314.50	500.02	0.00	3495.00	365.39	293.50	302.84	1.00	1524.00	13.61	11.00	10.70	0.00	52.00
4	84.05	52.50	102.85	0.00	584.00	80.43	53.00	90.57	1.00	478.00	5.75	5.00	4.66	0.00	24.00
5	601.71	396.50	557.74	2.00	2849.00	489.61	368.50	370.75	4.00	1677.00	18.50	16.00	14.20	1.00	74.00
6	243.02	164.50	231.45	0.00	898.00	225.30	162.50	203.31	1.00	764.00	9.36	7.00	7.89	0.00	37.00
7	66.05	44.00	65.70	0.00	335.00	65.67	45.00	62.01	1.00	314.00	5.35	4.00	4.67	0.00	19.00
8	414.79	291.00	399.04	15.00	2483.00	372.23	279.00	313.86	17.00	1744.00	14.34	11.00	10.95	1.00	68.00
9	574.65	439.00	520.08	2.00	2558.00	502.32	420.00	403.84	4.00	1803.00	16.19	13.00	12.13	1.00	61.00
10	81.65	52.00	88.25	0.00	575.00	80.91	53.50	83.38	1.00	534.00	6.13	5.00	4.65	0.00	20.00
11	1115.85	594.00	1318.30	0.00	7572.00	876.33	565.50	828.60	1.00	3786.00	19.45	14.00	18.84	0.00	87.00
12	1237.94	1036.50	1023.22	2.00	4789.00	1021.75	941.00	727.05	4.00	3078.00	23.59	20.00	16.60	1.00	74.00
13	1529.05	959.00	1694.05	3.00	7310.00	1187.00	891.00	1087.87	5.00	4202.00	21.93	16.00	19.67	1.00	98.00
14	1527.72	947.50	1852.38	8.00	9698.00	1176.03	887.00	1051.08	10.00	4769.00	23.84	18.50	20.97	1.00	107.00
15	1671.58	1029.00	1724.95	0.00	8898.00	1343.99	957.00	1132.74	1.00	5041.00	26.37	21.00	23.21	0.00	120.00
16	3715.67	3104.00	3289.47	65.00	18856.00	2597.46	2553.50	1722.80	67.00	7104.00	36.50	28.00	31.03	1.00	135.00
17	401.51	247.00	454.12	1.00	2715.00	380.71	245.00	404.40	3.00	2295.00	12.99	10.00	10.67	0.00	47.00
18	2371.20	1656.50	2296.85	19.00	10017.00	1883.87	1518.00	1521.90	21.00	6039.00	28.75	21.00	25.50	1.00	109.00
19	1048.29	651.50	1199.67	31.00	7100.00	926.81	627.00	924.16	33.00	4871.00	19.92	15.00	17.25	1.00	82.00
20	2353.83	1845.50	1898.68	47.00	8071.00	1965.46	1692.00	1398.36	49.00	5515.00	27.60	22.0	22.80	1.00	102.00
Neville															
1	47.61	47.50	49.23	0.00	278.00	33.02	31.50	23.10	1.00	89.00	4.61	3.50	3.89	0.00	16.00
2	181.90	152.00	176.20	0.00	968.00	142.16	114.00	107.05	1.00	478.00	9.24	7.00	7.28	0.00	28.00
3	443.11	380.50	435.98	0.00	1958.00	355.42	269.50	294.37	1.00	1192.00	11.37	9.00	8.83	0.00	42.00
4	88.90	90.00	86.86	0.00	438.00	86.28	59.00	79.70	1.00	395.00	5.25	3.50	4.57	0.00	18.00
5	627.15	554.00	519.77	13.00	3125.00	514.05	474.00	357.92	15.00	1760.00	15.23	11.50	12.00	1.00	46.00
6	240.76	171.00	238.60	0.00	1092.00	223.14	147.50	207.55	1.00	901.00	9.63	7.00	7.86	0.00	34.00
7	64.05	56.00	53.57	0.00	265.00	63.49	53.00	50.62	1.00	246.00	5.83	4.00	4.65	0.00	18.00
8	406.36	367.50	434.64	2.00	2431.00	360.96	245.00	346.01	4.00	1689.00	11.96	9.50	10.60	1.00	48.00
9	594.15	535.50	489.96	2.00	2448.00	523.27	417.50	391.60	4.00	1801.00	17.59	15.00	14.02	1.00	56.00
10	75.33	46.00	83.51	0.00	419.00	74.92	44.50	79.66	1.00	398.00	5.60	4.00	4.56	0.00	21.00
11	1203.58	990.50	1413.61	0.00	7020.00	926.54	635.00	874.09	1.00	3645.00	19.15	15.00	17.59	0.00	117.00
12	1139.67	837.00	1305.51	2.00	9144.00	911.31	648.50	790.61	4.00	4167.00	22.20	18.50	19.20	1.00	99.00
13	1465.15	1181.50	1509.38	6.00	8697.00	1169.27	901.00	972.77	8.00	4533.00	22.94	17.50	18.60	1.00	76.00
14	1540.68	1527.00	1379.59	18.00	6942.00	1250.55	1023.00	932.83	20.00	4072.00	24.29	20.50	19.00	1.00	79.00
15	1974.82	1880.00	2009.95	0.00	10388.00	1534.43	1291.00	1281.68	1.00	5434.00	29.09	21.00	23.76	0.00	94.00
16	3835.49	2762.00	3987.39	16.00	23017.00	2560.39	2389.50	1850.69	18.00	7430.00	39.72	32.50	33.01	1.00	138.00
17	357.01	286.50	375.17	1.00	2473.00	341.58	238.00	339.51	3.00	2122.00	10.96	8.00	10.22	0.00	58.00
18	2405.76	1619.50	2704.83	6.00	18320.00	1862.98	1382.00	1557.72	8.00	7764.00	32.41	25.00	27.33	1.00	175.00
19	978.87	627.00	1087.30	11.00	7372.00	875.50	677.00	844.18	13.00	5012.00	17.90	14.00	17.09	1.00	112.00
20	2476.01	2530.00	2554.21	18.00	19746.00	1994.78	2044.50	1551.64	20.00	8634.00	28.85	25.50	22.83	1.00	89.00

with $z \in \mathbb{F}_p$, which is not a bijection. Specifically, a polynomial that interpolates the function value $f(t^*) = 0$ is required. Both methods are constructive, thus the interpolation procedure begins by interpolating two function values of the function $f(t)$ for two random values of t. The resulting polynomial is evaluated at zero and the obtained value t_o is verified by computing $f(t_o)$. If $f(t_o) = 0$ then t_o is the Lucas logarithm to base m and the procedure is terminated, otherwise the value $f(t_o)$ becomes a new interpolation point.

For the computation of the function $f(t)$ two cases are considered with respect to the roots of the characteristic polynomial of the Lucas sequence. If the roots are in the field $\mathbb{F}_p$ then all computations are performed in the field $\mathbb{F}_p$. Alternatively, if the roots are in the field $\mathbb{F}_{p^2}$, then the modular exponentiations required to obtain the value $V_t(m)$ are performed in $\mathbb{F}_{p^2}$.

3.1. Experimental setup and results

The inverse interpolation methods for the Lucas logarithm problem were tested for several primes p, primitive roots g modulo p and base values m and z, for both cases of roots in $\mathbb{F}_p$ and $\mathbb{F}_{p^2}$, respectively. The instances of the problems considered are reported in Table 1, where “Roots in $\mathbb{F}_p$” values correspond to instances for the case of roots in the field $\mathbb{F}_p$ and “Roots in $\mathbb{F}_p^2$” values correspond to the case of roots in $\mathbb{F}_{p^2}$.

The results over 100 independent experiments for each instance of the problem are given in Tables 2 and 3. In both tables, the number of verifications (# Verifications), the number of points used (# Points Used), and the degree of the resulting polynomials (Pol. Degree) are reported. Verifications refers to the number of times that a polynomial evaluation at zero is per-

Table 3
Results of the inverse Aitken and inverse Neville methods over 100 independent experiments for the case of roots in the field $\mathbb{F}_{p^2}$

Problem	# Verifications					# Points Used					Pol. Degree				
	Mean	Med	SD	Min	Max	Mean	Med	SD	Min	Max	Mean	Med	SD	Min	Max
Aitken															
1	8.46	5.50	10.04	0.00	52.00	8.79	7.00	7.70	1.00	38.00	1.78	2.00	1.68	0.00	7.00
2	259.60	217.50	216.44	0.00	1029.00	188.77	183.50	125.91	1.00	485.00	11.70	9.50	8.61	0.00	39.00
3	792.54	625.50	666.09	2.00	2592.00	568.34	537.50	378.71	4.00	1367.00	18.39	16.50	12.93	1.00	47.00
4	1002.74	725.50	1053.57	6.00	7133.00	669.78	616.50	461.80	8.00	1948.00	19.34	15.00	17.28	1.00	112.00
5	1221.30	816.50	1236.63	0.00	8447.00	819.35	703.00	555.48	1.00	2333.00	21.43	18.50	16.93	0.00	73.00
6	1216.80	800.50	1328.68	2.00	7059.00	828.70	692.50	642.93	4.00	2583.00	19.48	17.00	15.38	1.00	84.00
7	1306.25	894.50	1394.61	5.00	10001.00	906.64	777.00	658.66	7.00	2904.00	24.78	23.00	20.42	1.00	131.00
8	460.96	325.00	444.91	2.00	2506.00	406.35	311.00	350.57	4.00	1775.00	14.92	13.00	11.39	1.00	52.00
9	1852.59	968.50	2190.82	65.00	13735.00	1169.41	850.50	920.04	66.00	3557.00	28.13	20.50	23.27	1.00	107.00
10	517.33	367.00	491.07	3.00	3244.00	462.29	350.50	377.75	5.00	2190.00	14.25	12.00	10.63	1.00	48.00
11	641.66	490.50	710.80	4.00	5491.00	562.10	469.00	508.02	6.00	3246.00	16.57	15.00	12.05	1.00	56.00
12	2322.47	2016.00	2028.76	3.00	11154.00	1637.83	1645.00	1089.07	5.00	4450.00	36.11	31.50	25.07	1.00	128.00
13	3349.57	2301.50	3346.93	25.00	16341.00	2124.32	1896.50	1499.86	27.00	5527.00	35.01	27.50	27.40	1.00	131.00
14	2314.08	1489.50	2559.03	31.00	15257.00	1640.25	1329.50	1285.18	33.00	5537.00	27.70	23.00	24.11	1.00	126.00
15	3167.17	2457.50	3003.33	9.00	13716.00	2205.43	2077.50	1568.08	11.00	6065.00	35.16	25.50	29.45	1.00	130.00
16	3875.90	3470.50	3187.13	54.00	15872.00	2716.24	2797.00	1649.76	56.00	6795.00	41.02	32.50	34.22	1.00	158.00
17	943.05	691.00	862.04	5.00	3881.00	852.02	665.50	722.69	7.00	3117.00	20.60	14.50	17.55	1.00	68.00
18	2272.53	1853.50	2076.25	10.00	11965.00	1841.48	1679.50	1397.48	12.00	6589.00	32.23	29.00	26.49	1.00	112.00
19	3751.20	2800.50	3565.80	33.00	17037.00	2673.58	2415.00	1959.90	35.00	7579.00	37.28	26.50	34.72	1.00	168.00
20	4972.11	3962.50	4461.65	2.00	16878.00	3384.46	3297.50	2388.80	4.00	8165.00	50.69	39.50	41.17	1.00	174.00
Neville															
1	7.36	8.00	7.57	0.00	33.00	7.93	6.00	6.17	1.00	29.00	1.77	2.00	1.58	0.00	6.00
2	269.56	164.00	343.32	0.00	2083.00	178.21	147.50	137.31	1.00	581.00	10.07	7.00	7.89	0.00	37.00
3	851.42	900.50	840.22	3.00	5562.00	580.66	463.00	403.11	5.00	1677.00	21.04	15.50	18.03	1.00	102.00
4	1133.77	1005.00	1110.86	4.00	5971.00	729.59	666.00	493.48	6.00	1901.00	22.38	18.00	17.16	1.00	85.00
5	1316.51	1312.50	1225.28	0.00	5497.00	868.04	790.00	574.28	1.00	2157.00	21.65	17.50	16.85	0.00	75.00
6	1634.80	1316.50	1623.25	57.00	6694.00	1022.62	921.50	736.33	57.00	2526.00	24.51	21.00	20.67	1.00	91.00
7	1502.49	1171.00	1403.31	5.00	6676.00	1014.02	950.00	709.49	7.00	2684.00	23.92	19.50	17.91	2.00	79.00
8	365.08	334.50	368.11	2.00	1871.00	330.31	249.00	301.13	4.00	1439.00	12.18	10.00	9.37	1.00	46.00
9	1825.26	1353.00	2055.20	18.00	9359.00	1166.45	863.50	912.58	20.00	3335.00	28.32	20.50	26.26	1.00	122.00
10	729.39	660.50	715.29	7.00	3595.00	619.75	506.50	516.94	9.00	2365.00	18.89	16.50	14.89	1.00	74.00
11	841.00	761.00	770.95	5.00	3295.00	724.86	633.00	591.19	7.00	2399.00	20.08	16.00	15.18	1.00	69.00
12	2179.27	1727.50	2342.17	2.00	12466.00	1490.78	1290.50	1154.39	4.00	4572.00	26.64	20.00	23.14	1.00	112.00
13	3227.29	2515.00	2925.22	97.00	16257.00	2159.17	1978.00	1312.52	98.00	5489.00	35.71	31.00	28.66	1.00	147.00
14	2952.24	1982.00	3670.16	10.00	21967.00	1852.00	1375.00	1521.13	12.00	5844.00	29.04	21.00	24.87	1.00	112.00
15	3352.11	3112.00	3736.95	32.00	22103.00	2218.47	1899.50	1638.08	34.00	6685.00	33.77	29.00	28.86	1.00	158.00
16	3952.63	3453.50	4181.73	29.00	25482.00	2603.54	2453.50	1891.57	31.00	7552.00	39.48	31.50	32.31	1.00	150.00
17	951.18	1125.50	939.90	3.00	5047.00	851.16	668.00	767.83	5.00	3723.00	21.73	16.50	18.31	1.00	97.00
18	2132.81	1954.50	2372.73	25.00	18035.00	1709.60	1362.50	1370.63	27.00	7732.00	25.69	19.50	21.77	1.00	102.00
19	4891.85	3395.00	5014.92	94.00	27638.00	3171.10	2881.50	2126.46	96.00	8569.00	48.10	39.50	39.87	1.00	206.00
20	4221.56	4179.50	4096.85	35.00	20638.00	2999.15	2767.00	2112.28	37.00	8723.00	40.68	31.50	31.38	1.00	146.00

formed, to verify whether the outcome is equal to the Lucas logarithm to base m. Each verification requires one evaluation of the function $f(t)$, for a specific value of t.

The quantity "Points Used" refers to the number of interpolation points, $y = f(t)$, employed by the procedure (and thus evaluated) until the required polynomial is found. However, due to the constructive character of the interpolation methods, the final polynomial may not interpolate all these points. Finally, "Pol. Degree" corresponds to the degree of the resulting polynomial that interpolates the value of the Lucas logarithm. The number of interpolation points that are used to construct the resulting polynomial is Pol. Degree + 1.

For each experiment, the two initial interpolation points are chosen randomly in the range $[1, p-1]$. For each one of the three quantities computed (Verifications, Point Used and Pol. Degree), the mean value (Mean), the median (Med), the standard deviation (SD), as well as, the minimum (Min) and the maximum (Max) values are reported.

The results indicate that both Aitken and Neville methods have similar behavior in finding the polynomial that interpolates the Lucas logarithm value (for both cases of roots origin) and require about one third of the field cardinality for verifications to obtain the polynomial, which is not small. However, an interesting finding that was also observed for the discrete logarithm problem in [20], is the low degree of the resulting polynomials, i.e. a low degree polynomial that interpolates the Lucas logarithm value exists.

In comparison to the results for the discrete logarithm problem [20], in the case of the Lucas logarithm problem the number of verifications required to find the proper polynomial is smaller than the corresponding one for the discrete logarithm problem. The same holds for the polynomial degree, i.e. the degrees of the polynomials that interpolate the discrete logarithm value are higher than that of the polynomials that interpolate the Lucas logarithm value. However, this is not the case for the number of points used, as more points are evaluated for the Lucas logarithm problem.

4. Conclusions

The Lucas logarithm problem is of great importance for modern cryptography as the security of many cryptosystems relies on it. Finding a polynomial of low degree or low sparcity that represents the Lucas logarithm is crucial. To this end, in this contribution Aitken and Neville inverse interpolation methods are employed to tackle the Lucas logarithm problem. Specifically, we apply inverse interpolation over values of the Lucas function to obtain a polynomial that interpolates the Lucas logarithm value. Both Aitken and Neville methods have a constructive character enabling us to begin with two interpolation points and to add new interpolation points directly with low cost. The results reported indicate that the computational cost for addressing the Lucas logarithm problem is high. However, low degree polynomials that interpolate the Lucas logarithm value exist in every tested instance. Finally, the results compared with the corresponding ones for the case of the discrete logarithm [20] provide an interesting insight with respect to the computational equivalence of the two problems.

References

[1] W. Diffie, M. Hellman, New directions in cryptography, IEEE Transactions on Information Theory 22 (1976).

[2] T. El Gamal, A public key cryptosystem and a signature scheme based on discrete logarithms, IEEE Transactions on Information Theory 31 (1985) 469–472.

[3] P. Smith, M. Lennon, LUC: a new public key system, in: Proceedings of the Ninth IFIP International Symposium on Computer Security, North-Holland, Amsterdam, 1993, pp. 103–117.

[4] P. Smith, C. Skinner, A public key cryptosystem and a digital signature system based on the Lucas function analogue to the discrete logarithms, Lecture Notes in Computer Science 917 (1995) 355–364.

[5] H. Aly, A. Winterhof, Polynomial representations of the Lucas logarithm, Finite Fields and their Applications 12 (2006) 413–424.

[6] D. Bleichenbacher, W. Bosma, A.K. Lenstra, Some remarks on Lucas-based cryptosystems, in: CRYPTO'95: Proceedings of the 15th Annual International Cryptology Conference on Advances in Cryptology, Springer-Verlag, London, UK, 1995, pp. 386–396.

[7] C.-S. Laih, F.-K. Tu, W.-C. Tai, On the security of the Lucas function, Information Processing Letters 53 (1995) 243–247.

[8] D. Coppersmith, I. Shparlinski, On polynomial approximation of the discrete logarithm and the Diffie–Hellman mapping, Journal of Cryptology 13 (2000) 339–360.

[9] E. Kiltz, A. Winterhof, Polynomial interpolation of cryptographic functions related to Diffie–Hellman and discrete logarithm problem, Discrete Applied Mathematics 154 (2006) 326–336.

[10] S. Konyagin, T. Lange, I. Shparlinski, Linear complexity of the discrete logarithm, Designs Codes and Cryptography 28 (2003) 135–146.

[11] T. Lange, A. Winterhof, Interpolation of the discrete logarithm in F_q by Boolean functions and by polynomials in several variables modulo a divisor of $q-1$, Discrete Applied Mathematics 128 (2003) 193–206.

[12] W. Meidl, A. Winterhof, Lower bounds on the linear complexity of the discrete logarithm in finite fields, IEEE Transactions on Information Theory 47 (1) (2001) 2807–2811.

[13] W. Meidl, A. Winterhof, A polynomial representation of the Diffie–Hellman mapping, Applicable Algebra in Engineering Communication and Computing 13 (4) (2002) 313–318.

[14] G.C. Meletiou, Explicit form for the discrete logarithm over the field $GF(p,k)$, Archiv der Mathematik (Brno) 29 (1–2) (1993) 25–28.

[15] G.L. Mullen, D. White, A polynomial representation for logarithms in $GF(q)$, Acta Arithmetica 47 (1986) 255–261.

[16] H. Niederreiter, Incomplete character sums and polynomial interpolation of the discrete logarithm, Finite Fields and their Applications 8 (2002) 184–192.

[17] A. Winterhof, Polynomial interpolation of the discrete logarithm, Designs Codes and Cryptography 25 (1) (2002) 63–72.

[18] A. Winterhof, A note on the linear complexity profile of the discrete logarithm in finite fields, Progress in Computer Science and Applied Logic 23 (2004) 359–367.

[19] R.L. Burden, J.D. Faires, Numerical Analysis, Brooks/Cole Publishing Company, 1997.

[20] E.C. Laskari, G.C. Meletiou, M.N. Vrahatis, Aitken and Neville inverse interpolation methods over finite fields, Applied Numerical Analysis and Computational Mathematics 2 (1) (2005) 100–107.