1

Transformations of Cryptographic Schemes Through Interpolation Techniques

Stamatios-Aggelos N. Alexandropoulos, Gerasimos C. Meletiou, Dimitrios S. Triantafyllou, and Michael N. Vrahatis

Abstract The problem of transforming cryptographic schemes using interpolation techniques is studied. Firstly, explicit forms for the discrete logarithm and the Diffie–Hellman cryptographic functions are given. Subsequently, the inverse Aitken and Neville interpolation methods for the discrete logarithm and the Lucas logarithm problems are presented. Next, the representation of cryptographic functions through polynomials or algebraic functions as well as a special case of discrete logarithm problem is given. Finally, a study of cryptographic functions using factorization of matrices is analyzed.

Keywords: Public key cryptography • Discrete logarithm • Diffie Hellman mapping • Polynomial interpolation techniques • Matrix factorization

1 Introduction

A basic task of cryptography is the transformation or encryption, of a given message into another one which appears meaningful only to the intended recipient after the process of decryption. Messages and cryptograms are represented as elements of finite algebraic structures. Encryption and decryption processes are functions over finite structures especially over finite fields.

It is well known that, in a finite field GF(q), where q is a prime power, every function can be represented as a polynomial through the Lagrangian interpolation.

S.-A.N. Alexandropoulos (✉) • M.N. Vrahatis
Computational Intelligence Laboratory (CILab), Department of Mathematics, University of Patras, GR-26110 Patras, Greece
e-mail: alekst@master.math.upatras.gr; vrahatis@math.upatras.gr

G.C. Meletiou
A.T.E.I. of Epirus, P.O. 110, GR-47100 Arta, Greece
e-mail: gmelet@teiep.gr

D.S. Triantafyllou
Hellenic Army Academy (SSE), University of Military Education, GR-16673 Vari, Attica, Greece
e-mail: dtriant@sse.gr

N.J. Daras, M.Th. Rassias (eds.), *Computation, Cryptography, and Network Security*, DOI 10.1007/978-3-319-18275-9_1

Also, for every function, $f : \mathrm{GF}(q) \to \mathrm{GF}(q)$, there exists a unique polynomial $p(x)$ of degree at most $(q-1)$ that coincides with f.

One of the most basic aspects of the numerical analysis, with diverse applications in the field of cryptography, is the interpolation techniques. It is worth noting that the past three decades have witnessed an increasing interest in the application of interpolation techniques of cryptographic functions. The Langrange's, Hermite's, Aitken's and Neville's interpolation methods are widely used for the interpolation process through which the encryption and decryption functions are approximated.

Interpolation is computationally attractive only in the case of a polynomial with small number of nonzero coefficients. Since encryption and decryption functions are defined as functions over finite fields, it is of great importance to attempt to express them as polynomials and perform cryptanalysis by polynomial computation.

In the work at hand we study the problem of transforming cryptographic schemes using interpolation techniques. In the second section we consider explicit forms of cryptographic functions, such as the discrete logarithm and the Diffie–Hellman functions. Subsequently, in the third section we present inverse interpolation methods, such as Aitken's and Neville's methods for the well-known discrete logarithm problem as well as the Lucas logarithm problem. Next, in the fourth section we present the representation of cryptographic functions through polynomials or algebraic functions, while in the fifth section we give a special case of discrete logarithm problem. Finally the chapter ends at the sixth section with a study of cryptographic functions using factorization of matrices.

2 Explicit Forms of Cryptographic Functions

Definition 1. Consider the case of a prime field $\mathbb{Z}_p$, where p is a prime. For a generator g of $\mathbb{Z}_p^*$, $\langle g \rangle = \mathbb{Z}_p^*$, the polynomial:

$$p(x) = \sum_{i=1}^{p-2} \frac{x^i}{1-g^i},$$

represents the ***discrete logarithm*** of x to the basis g, $\forall\ x \in \mathbb{Z}_p^*$.

Remark 1. Surprisingly enough the formulas of the coefficients are very simple [24].

Proposition 1 ([17]). *Using the discrete Fourier transform, we can also derive the following matrix representation:*

$$\log_g(x) = -(1\ 2\ \ldots\ p-1)\left(g^{-ij}\right)\begin{pmatrix} x \\ x^2 \\ \vdots \\ x^{p-1} \end{pmatrix},$$

where $(-g^{-ij})$, $1 \leqslant i,j \leqslant p-1$ *is an* $(p-1)\times(p-1)$ *matrix.*

It seems natural to generalize these results to logarithms where the base is not necessarily a primitive element in a field of prime power order. To this end, we recall the following result [16–20]:

Theorem 1. *Let $g \in \mathbb{F}_{p^n}^*$, g generator of the multiplicative group of the field, that is $\langle g \rangle = \mathbb{F}_{p^n}^*$, $g^z = x \in \mathbb{F}_{p^n}^*, 1 \leqslant z \leqslant p^n - 1$. Suppose that the numeral system with p as a basis is used:*

$$z = \sum_{s=0}^{n-1} d_s p^s, \qquad 0 \leqslant d_s \leqslant m.$$

Then, it holds that:

$$d_s = \sum_{i=1}^{p^n-2} \frac{x^i}{(1 - g^i)^{p^s}} \ .$$

Concerning the representations of the Diffie–Hellman key function Meidl and Winterhof in [15] gave the following result:

Theorem 2. *Assume that $g \in \mathbb{F}_{p^n}^*$, $|\langle g \rangle| = m$, m divides $p^n - 1$ and $1 \leqslant a, b \leqslant m$. Then the polynomial:*

$$f(x, y) = m^{-1} \sum_{i,j=1}^{m} g^{i \cdot j} x^i y^j,$$

satisfies the relation:

$$f\left(g^a, g^b\right) = g^{ab}.$$

Proposition 2. *Using the discrete Fourier transform, we can also derive the following matrix representation:*

$$f(x, y) = m^{-1} (y \ y^2 \ \ldots \ y^m) \left(g^{-ij}\right) \begin{pmatrix} x \\ x^2 \\ \vdots \\ x^m \end{pmatrix},$$

where (g^{-ij}) is an $m \times m$ matrix, $1 \leqslant i, j \leqslant m$.

3 Interpolation and Inverse Interpolation Methods

Aitken's and Neville's interpolation techniques, as well as the Lagrange interpolation method, are well known and they are considered as the state of the art for transforming of cryptographic functions over finite fields. In contrast to the Lagrange method, Aitken's and Neville's methods are constructive in a way that permits the addition of a new interpolation point directly and with low computational cost. Thus, the interpolation procedure 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, have motivated the investigation of their performance over finite fields. In this section, we study the inverse Aitken and the inverse Neville interpolation methods over finite fields for the discrete logarithm and the Lucas logarithm function.

3.1 The Aitken and Neville Interpolation and Inverse Interpolation Methods

We study the Aitken and Neville interpolation methods by considering a function $f(x)$ defined on a field $\mathbb{F}$ and $x_i \in \mathbb{F}$ be mutually different interpolation points. Also, we assume that $f_i = f(x_i)$, with $i = 0, 1, \ldots, n$. Then, the ***Aitken polynomial*** is defined as follows:

$$P_{0,1,\ldots,m,i}(x) = \frac{1}{(x_i - x_m)} \begin{vmatrix} P_{0,1,\ldots,m}(x) & x_m - x \\ P_{0,1,\ldots,m-1,i}(x) & x_i - x \end{vmatrix},$$

where $m = 0, 1, \ldots, n-1$, $i = m+1, \ldots, n$ and $x_0, x_1, \ldots, x_k$ are the interpolated points.

Similarly, the ***Neville interpolation*** formula is given by:

$$P_{i,1+i,\ldots,i+m}(x) = \frac{1}{(x_{i+m} - x_i)} \begin{vmatrix} P_{i,i+1,\ldots,i+m-1}(x) & x_i - x \\ P_{i+1,i+2,\ldots,i+m}(x) & x_{i+m} - x \end{vmatrix},$$

where $m = 1, 2, \ldots, n$, $i = 0, 1, \ldots, n-m$ and where $x_i, x_{i+1}, \ldots, x_{i+k}$ are the interpolated points.

The inverse interpolation problem [12] can be approached through Aitken's and Neville's interpolation techniques using the corresponding formulae [3]. Specifically, the corresponding formulae of the ***inverse Aitken interpolation method*** and the ***inverse Neville interpolation method*** are given as follows:

$$P_{0,1,\ldots,m,i}(y) = \frac{1}{(y_i - y_m)} \begin{vmatrix} P_{0,1,\ldots,m}(y) & y_m - y \\ P_{0,1,\ldots,m-1,i}(x) & y_i - y \end{vmatrix},$$

where $m = 0, 1, \ldots, n-1$, $i = m+1, \ldots, n$ and:

$$P_{i,1+i,\ldots,i+m}(y) = \frac{1}{(y_{i+m} - y_i)} \begin{vmatrix} P_{i,i+1,\ldots,i+m-1}(y) & y_i - y \\ P_{i+1,i+2,\ldots,i+m}(y) & y_{i+m} - y \end{vmatrix}.$$

An interesting point is the approach on the values of the ***shifted exponential function***:

$$f(x) = \alpha^x - b \,(\mathrm{mod}\, p), \quad \text{for } p \text{ prime and } \alpha \in \mathbb{Z}_p,$$

using the inverse Aitken and the inverse Neville interpolations method. Selected points of the function f are used to construct a polynomial that interpolates the value $f(x^*) = 0 \,(\mathrm{mod}\, p)$. The resulting polynomial is evaluated at zero by interpolating two random values of x in the beginning. Every new point becomes a new interpolate point, unless the value is the discrete logarithm of b over $\alpha \,(\mathrm{mod}\, p)$.

As it has been presented in [12] the computational cost for tackling the problem of discrete logarithm through both methods is high. Overall, Aitken's method proved slightly better than the Neville's method. The performance of two methods implies that the resulting polynomials were most often of low degree and in most cases there exists a low degree polynomial that interpolates the discrete logarithm.

3.2 Inverse Interpolation Methods for the Lucas Logarithm Problem

The Lucas function is a 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 subsection the Lucas logarithm problem is studied using the inverse Aitken and Neville interpolation methods. These methods are applied to values of the Lucas function to obtain a polynomial that interpolates the Lucas logarithm.

Definition 2. Suppose that p is 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:

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

Then the sequence $\{V_t(m)\}_{t=0}^{\infty}$ is called ***Lucas sequence*** generated by m and the mapping:

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

is called ***Lucas function***. Furthermore, given a prime p any $m \in \mathbb{F}_p$ and $z \in \{V_t(m)\}$ then, the integer x which satisfies the relation $V_x(m) = z$ is called the ***Lucas logarithm*** of z.

Remark 2. The security of cryptosystems based on the Lucas function relies on the difficulty of addressing the Lucas logarithm problem.

Remark 3. It was shown in [20] that $V_t(m) = \mu^t + \mu^{-t}$, $t \geqslant 0$, where μ and μ^{-1} are the roots of the characteristic polynomial of the above second-order linear recurrence relation.

Remark 4. The roots of the following equation:

$$f(X) = X^2 - mX + 1,$$

are given by the expressions:

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

and if $m^2 - 4$ is zero or a quadratic residue modulo p, then both μ and $-\mu$ are in $\mathbb{F}_p$, otherwise they are in the extension field $\mathbb{F}_{p^2}$.

Let us study the inverse Aitken and the inverse Neville interpolation methods over the ***shifted Lucas function***:

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

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_0 is verified by computing $f(t_0)$. If $f(t_0) = 0$, then t_0 is the Lucas logarithm to the base m and the procedure is terminated, otherwise the value $f(t_0)$ becomes a new interpolation point.

As it has been presented in [13] through several experiments, both Aitken's and Neville's methods have similar behavior in finding the polynomial that interpolates the Lucas logarithm value and require about one third of the field cardinality for verifications to obtain the polynomial, which is not small.

In comparison with the results for the discrete logarithm problem [12], in the case of Lucas logarithm problem the number of verifications required to find the proper polynomial is smaller than the corresponding one for the discrete logarithm

problem. Concerning the polynomial degree, the degrees of the polynomials that interpolate the discrete logarithm value are higher [12] than that of the polynomials that interpolate the Lucas logarithm value.

4 Interpolation of Cryptographic Functions for a Given Set of Data

Another approach is to represent the cryptographic functions with polynomials or algebraic functions coinciding with the functions over proper subsets of the domain. However it has been shown that polynomials approximating cryptographic transformations on sufficiently large sets must be of sufficiently large degree and sparsity. To this end, lower bounds on the degrees and the sparsity (i.e., the number of the nonzero coefficients) of polynomials interpolating the cryptographic functions can be obtained.

It has been shown that even for polynomial representations of the discrete logarithm over quite thin sets, the degree is still required to be high. These results support the assumption of hardness of the aforementioned functions if the parameters are properly chosen. The term "approximation" has been used for polynomials which coincide with the cryptographic function over a subset of its domain.

Concerning the discrete logarithm we have the result given by Coppersmith and Shparlinski [7] and Shparlinski [21]:

Theorem 3. *Let p be a prime, $g \in \mathbb{Z}_p^*$. Consider the subset $S \subset \{1, 2, \ldots, p-1\}$, $|S| = p - 1 - s$, $F(X) \in \mathbb{Z}_p[X]$ a polynomial satisfying $F(g^x) = x$, $\forall\, x \in S$. Then it holds that:*

$$\deg(F) \geqslant p - 2 - 2s \qquad \textbf{(lower bound)}.$$

Similar results can be derived for the Diffie–Hellman mapping:

Theorem 4. *Let q be a prime power, $g \in \mathbb{F}_q^*$. Consider the subset $A \subset [N+1, N+h] \times [N+1, N+h]$, where $2 \leqslant h \leqslant q-1$ and $|A| \geqslant 10h^{8/5}$. Assume that $F(U, V) \in \mathbb{F}_q[X, Y]$ satisfies $F(g^x, g^y) = g^{xy}$ for all $(x, y) \in A$. Then it holds that:*

$$\deg(F) \geqslant \frac{|A|^2}{128h^3} \qquad \textbf{(lower bound)}.$$

El Mahassni and Shparlinski in [10] gave for the decision Diffie–Hellman key problem the following result:

Theorem 5. *Let q be a prime power, $g \in \mathbb{F}_q^* = \langle g \rangle$. Consider the subset $A \subset [N+1, N+h] \times [N+1, N+h]$, where $2 \leqslant h \leqslant q-1$. The three variable polynomial $F(U, V, T) \in \mathbb{F}_q[X, Y, Z]$ satisfies $F(g^x, g^y, g^{xy}) = 0$ for all $(x, y) \in A$. Then it holds that:*

$$\deg(F) \geqslant \frac{|A|}{3h^{8/5}} \qquad \textbf{(lower bound)}.$$

Furthermore, lower bounds have been computed for functions related to the integer factoring problem and the RSA cryptosystem [1] as well as the Lucas logarithm [2].

5 The Double Discrete Logarithm and the Root of the Discrete Logarithm

Definition 3. Let G be a cyclic group of order t, $|\langle g \rangle| = |G| = t$ and $h \in \mathbb{Z}_t^*$ be an element of order $|\langle h \rangle| = m$. The ***double discrete logarithm*** of an element $z = g^{h^x} \in G$ to the bases g and h is the unique $x : 0 \leqslant x < m$.

Remark 5. The parameters G, t, g, and h should be chosen such that computing discrete logarithms in G to the base g and in $\mathbb{Z}_t^*$ to the base h are infeasible.

Remark 6. The double discrete logarithm is used as one-way function in several cryptographic schemes, in particular in group signature schemes and publicity verifiable secret sharing schemes.

The verifiable encryption of discrete logarithms is a typical example. Specifically we have [22]:

1. Assume that $|\langle g \rangle| = |G| = p$, p is prime, $p = 2q + 1$, $h \in \mathbb{Z}_{p'}^*$, $|\langle h \rangle| = q$, q prime.
2. A private key $z \in \mathbb{Z}_q$ is randomly chosen and the public-key $y \equiv h^z (\mathrm{mod}\, p)$ is published.
3. A message v is encrypted as (A, B), $A \equiv h^\alpha (\mathrm{mod}\, p)$ and $B \equiv v^{-1} y^\alpha (\mathrm{mod} p)$ (El Gamal's public key cryptosystem [9]).
4. The element $w = g^v$ becomes public.
5. Verifying that a pair (A, B) encrypts the discrete logarithm of a public element $w = g^v$ of the group G is equivalent to verifying that the discrete logarithm of A to the base h is identical to the double discrete logarithm of w^B to the bases g and y.

Definition 4. Let G be a cyclic group of order t, $|\langle g \rangle| = |G| = t$, $Y \in G$ be an element of the group G. A kth ***root of the discrete logarithm*** of Y to the base g is an integer satisfying $x : 0 \leqslant x < t$ satisfying $Y = g^{x^k}$ if such an x exists.

Remark 7. Existence and uniqueness of the kth root of the discrete logarithm are not guaranteed. In the case $\left|\{x : g^{x^k} = y\}\right| \geqslant 2$ branches of the kth root of the discrete logarithm are defined.

Remark 8. Group G and parameters g and t can be chosen in such a way that computing discrete logarithms to the base g is infeasible. Also, it can be chosen such that obtaining kth roots modulo t is hard.

Remark 9. The kth root of the discrete logarithm is used as one-way function [4, 5, 14] in group signature schemes, publicly verifiable secret sharing schemes, electronic cash, offline electronic cash systems, anonymity control in multi-bank e-cash system, in history-based signatures, etc.

The following proposition gives an insight for the lower bounds of the polynomial representation of the double discrete logarithm:

Proposition 3. *Let $t \geqslant 3$ be an integer, p be a prime, $p \equiv 1 (\mathrm{mod}\ t)$, $g \in \mathbb{F}_p^*$ an element of order $m \geqslant 2$, $S \subseteq \{0, 1, \ldots, m-1\}$ a set of order $|S| = m - s$ and $f(x) \in \mathbb{F}_p[x]$ a polynomial satisfying the following relation:*

$$f(g^{h^n}) = n, \quad \forall\ n \in S.$$

Then it holds that:

$$\deg(f) \geqslant \frac{m - 2s}{2v} \qquad \textbf{(lower bound)},$$

where v is the smallest integer in the set $\{h^n (\mathrm{mod}\ t) : 1 \leqslant n \leqslant m\}$.

Similar results can be obtained in the case of the multiplicative group of fields of prime power order and groups derived from elliptic curves. Lower bounds can also be computed for the degree of the polynomial which represents the root of the discrete Logarithm:

Proposition 4. *Let p be a prime number, $g \in \mathbb{Z}_p^*$, $|\langle g \rangle| = t$ and let $k \geqslant 1$ be an integer s.t. $\gcd(k, \phi(t)) = 1$. Let $S \subset \mathbb{Z}_t^*$ be a subset of order $|S| = \phi(t) - s$. We assume the existence of a polynomial $F(X) \in \mathbb{Z}_p[X]$ s.t. $F(g^{x^k}) = x$, $\forall\, x \in S$. Then it holds that:*

$$\deg(F) \geqslant \frac{\phi(t) - 2s}{2} \qquad \textbf{(lower bound)}.$$

Remark 10. The exponent k is odd and relatively prime to $\phi(t)$ and the kth root function becomes a bijection.

Remark 11. The main motivation stems from RSA. In this case k is the encryption exponent e. In some applications the message m is encrypted as $c \equiv m^e (\mathrm{mod}\ \ N)$ and g^{m^e} becomes public. Recovering m from g^{m^e}, or verifying properties of m is the problem. For proofs of knowledge of roots of discrete logarithms, we refer the interested reader to [4].

6 Matrix Factorization in Cryptography

Before we proceed to methods for the matrix representation of cryptographic functions, we give some necessary definitions and theorems.

Definition 5. An $m \times n$ matrix whose row-entries are terms of a geometric progression is called ***Vandermonde matrix*** and has the following expression:

$$V = \begin{pmatrix} 1 & a_1 & a_1^2 & \cdots & a_1^{n-1} \\ 1 & a_2 & a_2^2 & \cdots & a_2^{n-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & a_{m-1} & a_{m-1}^2 & \cdots & a_{m-1}^{n-1} \\ 1 & a_m & a_m^2 & \cdots & a_m^{n-1} \end{pmatrix}.$$

In order to extract useful pieces of information for a matrix, including the rank, the eigenvalues and eigenvectors as well as the determinant among others, its factorization can be used. In matrices with real or complex entries, the use of orthogonal transformations such as Householder's transformations for computing the QR factorization or the singular value decomposition (SVD) [8] improves the stability of the algorithms increasing simultaneously the floating point operations. Non-orthogonal techniques such as LU factorization with partial or complete pivoting [8] reduce the required computational complexity giving a higher, but acceptable bound, for the norm of the error.

In the case of finite fields there is no error, thus the use of non-orthogonal methods which are faster is more suitable. Since in cryptography the required storage capacity of a method should not be greater than that of the initial data, the QR factorization is not preferable. The LU factorization does not require extra storage capacity and has less computational complexity.

Below we present the LU factorization with/without partial/complete pivoting of a matrix.

Theorem 6 (LU Factorization without Pivoting [8]). *Let A be an $m \times n$ matrix. Then there are a lower triangular $m \times m$ matrix L with ones in its main diagonal and an upper triangular $m \times n$ matrix such that $A = L \cdot U$.*

Theorem 7 (LU Factorization with Partial Pivoting [8]). *Let A be an $m \times n$ matrix. Then there are an $m \times m$ row permutation matrix P, a lower triangular $m \times m$ matrix L with ones in its main diagonal and an upper triangular $m \times n$ matrix such that $P \cdot A = L \cdot U$.*

Theorem 8 (LU Factorization with Complete Pivoting [8]). *Let A be an $m \times n$ matrix. Then there are an $m \times m$ row permutation matrix P, an $n \times n$ column permutation matrix Q, a lower triangular $m \times m$ matrix L with ones in its main diagonal, and an upper triangular $m \times n$ matrix such that $P \cdot A \cdot Q = L \cdot U$.*

Proposition 5. *The required floating point operations of LU factorization of an* $m \times n$ *matrix is* $O(n^2(m - \frac{n}{3}))$.

Below, we present the error analysis for the LU factorization with partial pivoting.

Proposition 6. *The LU factorization is the exact factorization of the slightly disturbed initial matrix A:*

$$A + E = L \cdot U, \qquad \|E\|_\infty \leqslant n^2 \, \rho \, u \, \|A\|_\infty,$$

where ρ *is the growth factor (in case of row pivoting) and u the unit round off.*

Remark 12. The theoretical bound of the norm of the error matrix is unfortunately large due to the growth factor.

Remark 13. It has been proved that in the case of Gaussian elimination with partial pivoting holds that [8, 25]:

$$\rho \leqslant 2^{n-1},$$

while in the case of Gaussian elimination with complete pivoting holds that:

$$\rho \leqslant \left(n \cdot 2^1 \cdot 3^{1/2} \cdot 4^{1/3} \cdots n^{1/(n-1)}\right)^{1/2}.$$

Remark 14. Although the theoretical bound for the norm of the error matrix in LU factorization with partial pivoting is too high, in practice there are only a few examples for which the error is not satisfactory. Thus, the LU factorization with partial pivoting is one of the most popular matrix-factorization methods.

Next we present a high level description of the LU factorization with partial pivoting algorithm:

Algorithm LU factorization with partial pivoting [8]

for $k = 1 : \min\{m - 1, n\}$
  Find $r : |a_{r,k}| = \max_{k \leqslant i \leqslant m}\{|a_{i,k}|\}$
  Interchange rows k and r
  $m_{ik} = -a_{ik}/a_{kk}, \; i = k + 1 : m$
  $a_{ij} = a_{ij} + m_{ik} \, a_{kj}, \; i = k + 1 : m, j = k + 1 : n$
  Set $a_{i,j} = 0$ if $|a_{i,j}| \leqslant \epsilon_t, i = k : m + n, j = k : m + n$

Row interchanges can be saved in a vector p, where p_i is the number of row which is the maximum element in absolute value in column i for the rows $i, i + 1, \ldots, m$ in step i of the algorithm. Let P_i be the permutation matrix in step i and $P = P_{n-1} \cdots P_2 \cdot P_1$, then the LU factorization with partial pivoting is $P \cdot A = L \cdot U$.

6.1 Vandermonde Matrices

The Vandermonde matrices can be used for the representation of the discrete logarithm function as well as the Diffie–Hellman mapping. These matrices are derived from the interpolation process.

In [11, 18] LU-decomposition for Vandermonde matrices through Newton polynomial has been elaborated and new forms of both these problems have been provided. These new forms constitute an alternative approach to view and study the equivalence of the two problems and evidence new ideas for the generation of new cryptographic functions. The symmetric $(p-1)\times(p-1)$ Vandermonde matrix W is used:

$$W = \{W_{ij}\},\ \ i \leqslant i,j \leqslant p-1, \quad \text{with} \quad W_{ij} = w^{(i-1)(j-1)},$$

where $w = g^{-1}$. The matrix W is a ***discrete Fourier transform***, thus explicit forms for the cryptographic function of Sect. 2 can be written as follows:

$$\log_g(x) = -(p-1,1,2,\ldots,p-2)\ W\,(x^{\,p-1},x,\ldots,x^{\,p-2})^{\top}, \tag{1}$$

and

$$K(x,y) = -(x^{\,p-1},x,x^2,\ldots,x^{\,p-2})\ W\,(y^{\,p-1},y,\ldots,y^{\,p-2})^{\top}, \tag{2}$$

respectively. Then, using LU-decomposition, the matrix W can be factorized to $W = L\cdot U$, which equals to

$$U = \begin{pmatrix} 1 & 1 & 1 & 1 & \ldots & 1 \\ 0 & w-1 & w^2-1 & w^3-1 & \ldots & w^{p-2}-1 \\ 0 & 0 & (w^2-1)(w^2-w) & (w^3-1)(w^3-w) & \ldots & (w^{p-2}-1)(w^{p-2}-w) \\ 0 & 0 & 0 & \Pi_{j=0}^{2}(w^3-w^j) & \ldots & \Pi_{j=0}^{2}(w^{p-2}-w^j) \\ \vdots & \vdots & \vdots & \ldots & \ddots & \vdots \\ 0 & 0 & 0 & \ldots & 0 & \Pi_{j=0}^{p-3}(w^{p-2}-w^j) \end{pmatrix}.$$

Since the matrix W is symmetric, the upper triangular matrix U can also be factorized to $U = D\cdot L^{\top}$, where $D = \mathrm{diag}(U)$.

Thus, the matrix L assumes the form:

$$U=\begin{pmatrix} 1 & 0 & 0 & 0 \ldots 0 \\ 1 & 1 & 0 & 0 \ldots 0 \\ 1 & (w^2-1)(w-1)^{-1} & 1 & 0 \ldots 0 \\ 1 & (w^2-1)(w-1)^{-1} & (w^3-1)(w^3-w)(w^2-1)^{-1}(w^2-w)^{-1} & 1 \ldots 0 \\ \vdots & \vdots & \vdots & \vdots \ddots \vdots \\ 1 & (w^{p-2}-1)(w-1)^{-1} & \ldots & \ldots\ldots 1 \end{pmatrix}.$$

By setting $F(x) = L^\top x$, with $x^\top = (x^{p-1}, x, \ldots, x^{p-2})$ and by using the previous factorization of the matrix W and taking into consideration the Eqs. (1) and (2), then the ***discrete logarithm function*** can be written as follows:

$$-\eta^\top LDL^\top x = -\eta^\top LDF(x),$$

where $\eta^\top = (p-1, 1, 2, \ldots, p-2)$. Also, the ***Diffie–Hellman key function*** can be written as follows:

$$-y^\top LDL^\top x = -F^\top(y)LDF(x),$$

where $y^\top = (y^{p-1}, y, y^2, \ldots, y^{p-2})$. In the case of the ***Diffie–Hellman mapping*** (where $x = y$), we obtain the following quadratic form:

$$-x^\top LDL^\top x = -F^\top(x)DF(x),$$

which is computationally equivalent to the Diffie–Hellman function. The Diffie–Hellman mapping can also be written as follows:

$$-c^\top LDL^\top y, \quad \text{where} \quad c^\top = \left(g^0, g^{1^2}, g^{2^2}, \ldots, g^{(p-2)^2}\right).$$

6.2 LU Factorization in Cryptography

The LU factorization with partial pivoting can be applied in order to encrypt a message [6, 23]. Let $A \in \mathbb{R}^{m\times n}$ (or $A \in \mathbb{C}^{m\times n}$) be a matrix containing the initial message. If L and U are lower and upper triangular matrices, respectively, and P is a row permutation matrix as described previously, such that $P \cdot A = L \cdot U$, then the initial message is efficiently encrypted in L and U. It has been proved that the problem of restoring the initial message even though the matrix L or the matrix U is known constitutes an NP-hard problem, i.e., it cannot be solved in a practical amount of time [6]. If L is known from one person and U is known from another one, then

the two persons have to meet together and multiply their matrices in order to decrypt the initial message. Alternatively, the LU factorization with complete pivoting can be applied in order to enforce the stability of the algorithm.

Below, we present an example implementing the LU factorization with complete pivoting in order to encrypt an initial message. Then, the matrix multiplications is used in order to restore the message.

Example 1. Let us assume the following matrix:

$$A = \begin{pmatrix} 0.5688 & 0.1622 & 0.1656 & 0.6892 \\ 0.4694 & 0.7943 & 0.6020 & 0.7482 \\ 0.0119 & 0.3112 & 0.2630 & 0.4505 \\ 0.3371 & 0.5285 & 0.6541 & 0.0838 \end{pmatrix}.$$

We apply the LU factorization with complete pivoting to A.

Step 1:

The maximum element in absolute value in A is 0.7943 in the second row and second column.
Interchange rows 1 and 2 and columns 1 and 2 of A.
Compute the multipliers $A_{i,1} \equiv L_{i,1} \equiv m_{i,1} = \frac{A_{i,1}}{A_{1,1}}, i = 2, 3, 4$
Update the elements of A: $A_{i,j} = A_{i,j} - A_{i,1} \cdot A_{1,j},\ i = 2, 3, 4,\ j = 1, 2, 3, 4$

$$A^{(1)} = \begin{pmatrix} 0.7943 & 0.4694 & 0.6020 & 0.7482 \\ 0 & 0.4730 & 0.0427 & 0.5365 \\ 0 & -0.1720 & 0.0271 & 0.1574 \\ 0 & 0.0248 & 0.2535 & -0.4140 \end{pmatrix}.$$

$$L = \begin{pmatrix} 1.0000 & 0 & 0 & 0 \\ 0.2042 & 1.0000 & 0 & 0 \\ 0.3918 & 0 & 1.0000 & 0 \\ 0.6654 & 0 & 0 & 1.0000 \end{pmatrix}.$$

Step 2:

The maximum element in absolute value in $A^{(1)}_{i,j}, i = 2, 3, 4,\ j = 2, 3, 4$ is 0.5365 in the second row and fourth column.
Interchange columns 2 and 4 of A.
Compute the multipliers $A_{i,2} \equiv L_{i,2} \equiv m_{i,2} = \frac{A_{i,2}}{A_{2,2}}, i = 3, 4$
Update the elements of A: $A_{i,j} = A_{i,j} - A_{i,1} \cdot A_{1,j},\ i = 3, 4,\ j = 2, 3, 4$

$$A = \begin{pmatrix} A^{(2)} = 0.7943 & 0.7482 & 0.6020 & 0.4694 \\ 0 & 0.5365 & 0.0427 & 0.4730 \\ 0 & 0 & 0.0146 & -0.3108 \\ 0 & 0 & 0.2865 & 0.3898 \end{pmatrix}.$$

$$L = \begin{pmatrix} L = 1.0000 & 0 & 0 & 0 \\ 0.2042 & 1.0000 & 0 & 0 \\ 0.3918 & 0.2934 & 1.0000 & 0 \\ 0.6654 & -0.7718 & 0 & 1.0000 \end{pmatrix}.$$

Step 3:

The maximum element in absolute value in $A_{i,j}^{(2)}, i = 3, 4,\ j = 3, 4$ is 0.3898 in the fourth row and fourth column.
Interchange rows 3 and 4 and columns 3 and 4 of A.
Interchange rows 3 and 4 of L except the diagonal entries.
Compute the multipliers $A_{i,3} \equiv L_{i,2} \equiv m_{i,3} = \frac{A_{i,3}}{A_{3,3}}, i = 4$
Update the elements of A: $A_{i,j} = A_{i,j} - A_{i,1} \cdot A_{1,j},\ i = 4,\ j = 3, 4$

$$A = \begin{pmatrix} U \equiv A^{(3)} = 0.7943 & 0.7482 & 0.4694 & 0.6020 \\ 0 & 0.5365 & 0.4730 & 0.0427 \\ 0 & 0 & 0.3898 & 0.2865 \\ 0 & 0 & 0 & 0.2430 \end{pmatrix}.$$

$$L = \begin{pmatrix} L = 1.0000 & 0 & 0 & 0 \\ 0.2042 & 1.0000 & 0 & 0 \\ 0.6654 & -0.7718 & 1.0000 & 0 \\ 0.3918 & 0.2934 & -0.7973 & 1.0000 \end{pmatrix}.$$

$U \equiv A$

In order to reduce the required storage capacity we save the matrix U in the upper triangular part of the initial matrix A, the matrix L (except the 1's of the main diagonal) to the lower triangular part of A, the row permutation matrix P as a vector $p = [2\ 1\ 4\ 3]$, and the column permutation matrix Q as a vector $q = [2\ 4\ 1\ 3]$ (matrices P and Q are the identity matrix with interchanged their rows and columns, respectively). Thus, $P \cdot A \cdot Q = L \cdot U$. The use of $A,\ p$, and q instead of $L,\ U,\ P,\ Q$ keeps the storage capacity to $O(n^2)$ which is the order of the storage capacity of the initial data. Even knowing either U or L it is an NP-hard problem to obtain the initial

data A. In order to restore the initial matrix A the following product $P^{-1} \cdot L \cdot U \cdot Q^{-1}$ must be computed. Due to the triangular form of L and U, only the required floating point operations have to be computed reducing the computational complexity of the multiplication. P and Q are permutation matrices, thus their inverses and their product do not increase the complexity.

7 Synopsis

In the work at hand we studied the problem of transforming cryptographic schemes using interpolation techniques.

We gave explicit forms for the discrete logarithm and Diffie–Hellman cryptographic functions. Also, we presented inverse interpolation methods, such as Aitken's and Neville's methods, for the well-known discrete logarithm problem as well as the Lucas logarithm problem.

Furthermore, we gave the representation of cryptographic functions through polynomials or algebraic functions and a special case of discrete logarithm problem. Finally, we analyzed a study of cryptographic functions using factorization of matrices.

References

1. Adelmann, C., Winterhof, A.: Interpolation of functions related to the integer factoring problem. Lect. Notes Comput. Sci. **3969**, 144–154 (2006)
2. Aly, H., Winterhof, A.: Polynomial representations of the Lucas logarithm. Finite Fields Appl. **12**(3), 413–424 (2006)
3. Burden, R.L., Faires, J.D.: Numerical Analysis, 6th edn. Brooks/Cole Publishing Company, Pacific Grove (1997)
4. Camenisch, J.L.: Group signature schemes and payment systems based on the discrete logarithm problem. Doctoral Dissertation, Zurich (1998)
5. Camenisch, J.L., Stadler, M.A.: Efficient group signature schemes for large groups. Lect. Notes Comput. Sci. **1294**, 410–424 (1997)
6. Choi, S.J., Youn, H.Y.: A novel data encryption and distribution approach for high security and availability using LU decomposition. Lect. Notes Comput. Sci. **3046**, 637–646 (2004)
7. Coppersmith, D., Shparlinski, I.: On polynomial approximation of the discrete logarithm and the Diffie-Hellman mapping. J. Cryptol. **13**(3), 339–360 (2000)
8. Datta, B.N.: Numerical Linear Algebra and Applications, 2nd edn. SIAM, Philadelphia (2010)
9. El Gamal, T.: A public-key cryptosystem and a signature scheme based on discrete logarithms. IEEE Trans. Inf. Theory **31**(4), 469–472 (1985)
10. El Mahassni, E., Shparlinski, I.E.: Polynomial representations of the Diffie-Hellman mapping. Bull. Aust. Math. Soc. **63**, 467–473 (2001)
11. Laskari, E.C., Meletiou, G.C., Tasoulis, D.K., Vrahatis, M.N.: Transformations of two cryptographic problems in terms of matrices. ACM SIGSAM Bull. **39**(4), 127–130 (2005)
12. Laskari, E.C., Meletiou, G.C., Vrahatis, M.N.: Aitken and Neville inverse interpolation methods over finite fields. Appl. Numer. Anal. Comput. Math. **2**(1), 100–107 (2005)

13. Laskari, E.C., Meletiou, G.C., Vrahatis, M.N.: Aitken and Neville inverse interpolation methods for the Lucas logarithm problem. Appl. Math. Comput. **209**, 52–56 (2009)
14. Lysyanskaya, A., Ramzan, Z.: Group blind digital signatures: a scalable solution to electronic cash. Lect. Notes Comput. Sci. **1465**, 184–197 (1998)
15. Meidl, W., Winterhof, A.: A polynomial representation of the Diffie-Hellman mapping. Appl. Algebra Eng. Commun. Comput. **13**, 313–318 (2002)
16. Meletiou, G.C.: Explicit form for the discrete logarithm over the field GF(p, k). Arch. Math. (Brno) **29**, 25–28 (1993)
17. Meletiou, G.C., Mullen, G.L.: A note on discrete logarithms in finite fields. Appl. Algebra Eng. Commun. Comput. **3**(1), 75–78 (1992)
18. Meletiou, G.C., Laskari, E.C., Tasoulis, D.K., Vrahatis, M.N.: Matrix representations of cryptographic functions. J. Appl. Math. Bioinformatics **3**(1), 205–213 (2013)
19. Mullen, G.L., White, D.: A polynomial representation for logarithms in GF(q). Acta Arith. **47**(3), 255–261 (1986)
20. Niederreiter, H.: A short proof for explicit formulas for discrete logarithms in finite fields. Appl. Algebra Eng. Commun. Comput. **1**(1), 55–57 (1990)
21. Shparlinski, I.E.: Cryptographic Applications of Analytic Number Theory: Complexity Lower Bounds and Pseudorandomness. Progress in Computer Science and Applied Logic, vol. 22. Birkhauser Verlag, Basel (2003)
22. Stadler, M.: Publicly verifiable secret sharing, advances in cryptology. Lect. Notes Comput. Sci. **1070**, 190–199 (1996)
23. Triantafyllou, D.: Numerical linear algebra methods in data encoding and decoding. J. Appl. Math. Bioinformatics **3**(1), 193–203 (2013)
24. Wells, A.L., Jr.: A polynomial form for logarithms modulo a prime. IEEE Trans. Inf. Theory **IT-30**, 845–846 (1984)
25. Wilkinson, J.H.: The Algebraic Eigenvalue Problem. Clarendon Press, Oxford (1965)