Isotropic Quadratic Forms, Diophantine Equations and Digital Signatures

Martin Feussner
Department of Informatics
University of Bergen
Bergen, Norway
martin.feussner@uib.no

Igor Semaev
Department of Informatics
University of Bergen
Bergen, Norway
igor.semaev@uib.no

***Abstract*—This work introduces DEFI - an efficient hash-and-sign digital signature scheme based on isotropic quadratic forms over a commutative ring of characteristic 0. The form is public, but the construction is a trapdoor that depends on the scheme's private key. For polynomial rings over integers and rings of integers of algebraic number fields, the cryptanalysis is reducible to solving a quadratic Diophantine equation over the ring or, equivalently, to solving a system of quadratic Diophantine equations over rational integers. It is still an open problem whether quantum computers will have any advantage in solving Diophantine problems.**

***Index Terms*—digital signatures, isotropic quadratic forms, Diophantine equations**

I. Introduction

Subset sum problem is usually treated as finding $0,1$-solutions to a linear Diophantine equation in n variables. More precisely, given positive integers $a_1,\dots,a_n,a$ it is to decide whether or not there exist x_i in $\{0,1\}$ such that

$$x_1a_1+\ldots+x_na_n=a.$$

This problem is known to be NP-complete [1]. Obviously, the subset sum problem is equivalent to solving (deciding) the system of multivariate quadratic Diophantine equations

$$x_1a_1+\ldots+x_na_n=a,\ x_1^2-x_1=0,\dots,\ x_n^2-x_n=0.$$

To decide whether or not a more general system of multivariate quadratic Diophantine equations

$$f_1(x)=0,\dots,f_m(x)=0, \tag{1}$$

where $x=(x_1,\dots,x_n)\in\mathbb{Z}^n$, and $f_i\in\mathbb{Z}[x]$, and $\deg f_i\le 2$, is solvable in integers is therefore at least NP-hard. Finding explicit integer solutions to (1) is generally difficult. In this work a new hash-and-sign digital signature scheme called DEFI is presented. The security of the scheme is based on the hardness of computing isotropic vectors for quadratic forms over commutative rings of characteristic 0. For polynomial rings R over $\mathbb{Z}$ and rings of integers of algebraic number fields the cryptanalysis is reducible to solving quadratic Diophantine equations over R or equivalently to solving systems of quadratic Diophantine equations over $\mathbb{Z}$ such as (1). For instance, for Section V parameters forging a signature is equivalent to solving a quadratic Diophantine equation over $R=\mathbb{Z}[X]/(X^{64}+1)$ in 3 variables or equivalently to solving a system of $m=64$ multivariate quadratic Diophantine equations over $\mathbb{Z}$ in $n=192$ variables. Also, there is a restriction on the solution size.

No modular transforms are used in the digital signature algorithm in this work, all calculations are performed in the ring of integers. Therefore, the security of the proposed algorithm does not rely on solving multivariate polynomial equations over finite fields as with Matsumoto-Imai [2] and Hidden Field Equations (HFE) [3] cryptosystems and their derivatives. Also, advances in solving common lattice problems such as Shortest Vector Problem (SVP) and Closest Vector Problem (CVP) does not seem to undermine the new scheme, see Sections IV-E and V below.

Several cryptographic schemes were claimed to be constructed upon the hardness of the subset sum problem and Diophantine equations. The most famous one is the Merkle-Hellman public key crypto-system, where a super-increasing vector, the scheme private key, was hidden with a modular linear transform to get the public key. The scheme was broken in [4]. Its variations were broken too, see [5] for a survey. Also, digital signature scheme [6] based on a quadratic congruence modulo a composite integer and its extensions were broken, see [7]. A number of key exchange protocols built on the difficulty of solving general Diophantine equations and finding equivalence for binary quadratic forms over rational integers were published in [8] and [9] respectively, see also the references in those publications. The cryptographic schemes above differ from the current proposal.

The idea of the new scheme and its cryptanalysis are due to Semaev, the implementation and all computer experiments are due to Feussner.

II. Isotropic Quadratic Forms

Suppose R is any commutative ring of characteristic 0 with unity and without zero divisors, a module over $\mathbb{Z}$ with finite or infinite basis $\alpha_0,\alpha_1,\dots,\alpha_{m-1},\dots$. For $a\in R$ where $a=a_0\alpha_0+a_1\alpha_1+\ldots+a_{m-1}\alpha_{m-1}, a_i\in\mathbb{Z}$ the function $|a|=\max_{0\le i<m}|a_i|$ defines a norm on R. Also, if

$y = (y_1, y_2, \ldots, y_n) \in R^n$, then we set $|y| = \max_{1\le i\le n} |y_i|$. Let

$$f(x_1, \ldots, x_n) = \sum_{1\le i\le n} c_{ii}x_i^2 + \sum_{1\le i<j\le n} 2c_{ij}x_ix_j \quad (2)$$

be a quadratic form over R. Denote $x = (x_1, \ldots, x_n)$, then $f(x) = x^TCx$, where $C \in R^{n\times n}$ is a symmetric $(n \times n)$-matrix with entries $c_{ij} \in R$. The quadratic form is called isotropic if it may represent 0. That is $f(z) = 0$ for a non-zero vector $z \in R^n$; the vector z is called isotropic. The security of the present digital signature scheme is based on the hardness of computing isotropic vectors $z \in R^n$ for the form $f(x)$. It is well known that given one solution to the homogeneous quadratic equation $f(x) = 0$, it is possible to calculate all other solutions over R by parametrisation [10]. However, in the proposed digital signature scheme some entries of the target isotropic $z \in R^n$ are prescribed by the hash value of a message. That makes the method inefficient for forgeries.

How to create an isotropic quadratic form $f(x)$ over R is shown in this section below. In Section III, we explain how to construct an isotropic vector z for $f(x)$. The vector z is a concatenation of the hash value h of the message and its signature y. To verify the signature, one checks that $f(z) = 0$ in R. When $R = \mathbb{Z}[X]/(q)$, where (q) is the ideal in $\mathbb{Z}[X]$ generated by a monic irreducible polynomial $q = q(X) \in \mathbb{Z}[X]$, the cryptanalysis of the scheme is presented in Section IV. Numerical parameters are proposed in Section V, they provide 128-bit security of the scheme which corresponds to the NIST security category 1 according to [11]. The performance of the scheme is there provided too.

Let r, s, n be positive integers such that $s \ge 2$ and $n = r+s$. Let J be a diagonal matrix of size $n\times n$ with diagonal entries ±1 as

$$J = \text{Diag}\,(\pm1, \ldots, \pm1, \pm1),$$

where both 1 and -1 may occur. Suppose $B \in R^{n\times n}$ is a matrix of size $n \times n$ over R and of rank n (the rows of B are linearly independent over R). It is easy to see that

$$f(x) = f(x_1, \ldots, x_n) = (Bx)^TJ(Bx) = x^TCx, \quad (3)$$

where $C = B^TJB \in R^{n\times n}$, is an isotropic quadratic form. For matrices B specified in Section III isotropic vectors are easy to calculate.

III. Signature Scheme

A. Private Key

Private key of the signature scheme is a matrix $B \in R^{n\times n}$, constructed with blocks as

sizes	r	s
r	B_{11}	0
s	B_{21}	B_{22}

where B_{ij} are matrices over R of sizes according to the definition above and the matrix B_{22} is invertible in $R^{s\times s}$. For efficiency reasons, the entries of $B_{11}, B_{21}, B_{22}, B_{22}^{-1}$ may be taken of relatively small norms. To construct B_{22}, formulae in Section III-F may be used.

B. Public Key

Public key of the signature scheme is the matrix $C = B^TJB \in R^{n\times n}$ which determines the quadratic form (3).

C. Signature Generation

Let M be a message and $h \in R^r$ encodes its hash value. One may take the entries of h of small norms.

1. Given M, compute $h \in R^r$.
2. Set $Z' = B_{11}h \in R^r$. Generate randomly $Z'' \in R^s$ such that $Z^TJZ = 0$, where $Z = (Z'|Z'') \in R^n$. See Section III-G, where the construction is specified for $r = 1, s = 3$.
3. Compute $y \in R^s$ by
$$y = B_{22}^{-1}\,(Z'' - B_{21}h)\,.$$
4. The signature for M is y.

In the variation of the scheme presented in Section V, an additional parameter δ_y is used. The generated signature y is correct if $|y| < \delta_y$.

D. Signature Verification

Let M, y be a signed message.

1. If $y \notin R^s$, then reject. Otherwise, compute $h \in R^r$.
2. Set $z = (h|y) \in R^n$. If
$$f(z) = z^TCz = 0,$$
then accept the signature, otherwise reject.

In Section V, the signature is rejected if $|y| \ge \delta_y$.

E. Verification Proof

Let M, y be a correctly generated signature. For $z = (h|y)$ we have

$$B_{21}\,h + B_{22}\,y = Z''$$

and

$$Bz = \begin{pmatrix} B_{11} & 0 \\ B_{21} & B_{22}\end{pmatrix}\begin{pmatrix} h \\ y\end{pmatrix} = \begin{pmatrix} Z' \\ Z''\end{pmatrix} = Z.$$

So,

$$f(z) = z^TCz = [Bz]^TJ[Bz] = Z^TJZ = 0.$$

F. How to Generate B_{22}

Set

$$B_{22} = P_1E_1P_2E_2\ldots P_kE_kF \quad (4)$$

for randomly generated elementary and permutation matrices E_i and P_i respectively and a unimodular matrix $F \in R^{s\times s}$ which is easy to invert and hard to guess. The number k is a parameter, see explicit constructions in Section V. Then

$$B_{22}^{-1} = F^{-1}E_k^{-1}P_k^{-1}E_{k-1}^{-1}P_{k-1}^{-1}\ldots E_1^{-1}P_1^{-1}.$$

A matrix $E \in R^{s\times s}$ is called elementary if $E = \text{Diag}(1, \ldots, 1) + V_{ij}, 1 \le i, j \le s, i \ne j$, where $V_{ij} \in R^{s\times s}$ is such that

$$V_{ij}[u, v] = \begin{cases} b \ne 0 & \text{if } (u, v) = (i, j), \\ 0 & \text{if } (u, v) \ne (i, j). \end{cases}$$

Then $E^{-1} = \text{Diag}(1, \ldots, 1) + V'_{ij}$, where

$$V'_{ij}[u, v] = \begin{cases} -b & \text{if } (u, v) = (i, j) \\ 0 & \text{if } (u, v) \neq (i, j). \end{cases}$$

G. How to Generate Z

Set $r = 1, s = 3, n = 4$. The construction may be easily extended to larger parameters. Fix $B_{11} = 1$ in the definition of B and $J = \text{Diag}(1, 1, -1, -1)$. Then

1) set $Z_1 = Z' = h$. To construct $Z'' = (Z_2, Z_3, Z_4) \in R^3$ do the following.
2) Take randomly $u_1, u_2 \in R$ such that $u_2 - u_1^2 u_2 = 2v$ for some $v \in R$. For instance, one may take $u_1 = 1 + 2v_1$ for some $v_1 \in R$, then $v = -2u_2(v_1 + v_1^2)$. Or one may take $u_2 = 2v_2$ for some $v_2 \in R$. Then $v = v_2(1 - u_1^2)$. For efficiency, one may take u_1, u_2 of relatively small norms.
3) Set

$$\begin{aligned} Z_2 &= v + u_2 u_1^2 - Z_1 u_1, \\ Z_3 &= v + Z_1 u_1, \\ Z_4 &= u_1 u_2 - Z_1, \end{aligned}$$

and $Z = (Z_1, Z_2, Z_3, Z_4)$.

Therefore,

$$\begin{aligned} Z^T J Z &= Z_1^2 + Z_2^2 - Z_3^2 - Z_4^2 \\ &= Z_1^2 - (u_1 u_2 - Z_1)^2 + (v + u_2 u_1^2 - Z_1 u_1)^2 \\ &\quad -(v + Z_1 u_1)^2 \\ &= (2Z_1 - u_1 u_2) u_1 u_2 \\ &\quad -(2v + u_2 u_1^2)(2Z_1 u_1 - u_2 u_1^2) \\ &= (2Z_1 - u_1 u_2) u_1 u_2 - (2Z_1 - u_1 u_2) u_1 u_2 \\ &= 0. \end{aligned}$$

IV. Cryptanalysis

The security of the scheme depends on the basis ring R not counting the parameters r, s, n. Let $R = \mathbb{Z}[X]/(q)$, where (q) is the ideal in $\mathbb{Z}[X]$ generated by a monic irreducible polynomial $q = q(X)$ of degree m with integer coefficients. We set $|a|, a \in R$ to be the maximum in absolute values of the coefficients of a polynomial of degree $< m$ which represents a modulo $q(X)$ and call that a max-norm. To simplify some arguments below, we may assume that R is the ring of integers of the algebraic number field $K = \mathbb{Q}(\alpha)$, where α is a root of $q(X)$.

A. Private Key Recovery

Given public matrix C recover a matrix $B \in R^{n \times n}$ such that $C = B^T J B$. That equation may be written as a system of $(n^2 + n)/2$ quadratic Diophantine equations in $n(n - r)$ variables, the entries of B_{21}, B_{22}, over R and is generally hard to solve. If the entries of B are represented by very sparse polynomials a guessing strategy may work to recover them. That is to be avoided when choosing the parameters, see Section V.

B. Forgery Attack over $\mathbb{Z}$

One may write the form (3) as

$$f(x) = x^T C x = f_0(\bar{x}) + f_1(\bar{x})\alpha + \ldots + f_{m-1}(\bar{x})\alpha^{m-1}, \quad (5)$$

where $f_i(\bar{x})$ are quadratic forms over $\mathbb{Z}$ the variables of which are the coefficients of the polynomials $x_i = x_{i0} + x_{i1}\alpha + \ldots + x_{im-1}\alpha^{m-1}$ and

$$\bar{x} = (x_{10}, x_{11}, \ldots, x_{nm-1}).$$

Forging the signature for a message M with the hash $h = (x_1, \ldots, x_r)$ is thus equivalent to solving the system of quadratic Diophantine equations

$$f_0(\bar{x}) = 0, \ldots, f_{m-1}(\bar{x}) = 0,$$

where the variables

$$x_{ij},\ 1 \leq i \leq r,\ 0 \leq j < m$$

are fixed by the entries of h. That is a system of m Diophantine equations in $(n - r)m$ variables. Such equations are generally hard to solve as discussed in Section I.

C. Forgery Attack over R

In order to forge the signature for a message M, one may compute its hash $h \in R^r$ and set $(x_1, \ldots, x_r) = h$. One then randomly chooses $x_{r+1}, \ldots, x_{n-1}$ from R with bounded max-norms. One may try to calculate $z \in R$ such that $f(x) = 0$, where $x = (x_1, \ldots, x_r, x_{r+1}, \ldots, x_{n-1}, z)$, as

$$\begin{aligned} f(x) = c_{nn} z^2 + 2(c_{n1} x_1 + c_{n2} x_2 + \ldots + c_{nn-1} x_{n-1}) z \\ + g(x_1, \ldots, x_{n-1}) = 0. \end{aligned}$$

Denote $a = 2(c_{n1} x_1 + c_{n2} x_2 + \ldots + c_{nn-1} x_{n-1})$ and $b = g(x_1, \ldots, x_{n-1})$. If $c_{nn} \neq 0$, then z satisfies the quadratic equation

$$c_{nn} z^2 + a z + b = 0 \quad (6)$$

with roots $(-a \pm \sqrt{a^2 - 4bc_{nn}})/2c_{nn}$. One of the roots is in R if and only if

$$v = a^2 - 4bc_{nn} = u^2 \quad (7)$$

for some $u \in R$, and

$$2c_{nn} | a - u \quad \text{or} \quad 2c_{nn} | a + u. \quad (8)$$

We calculate the probability of the conditions with an heuristic argument. Let $D = \max |a^2 - 4bc_{nn}|$, where the maximum is taken over all possible values of $x_1, \ldots, x_{n-1}$ with bounded max-norms as above. Condition (7) implies that $\text{Norm}_{K/\mathbb{Q}}(v) \in \mathbb{Z}$ is a square of magnitude D^m. The probability that an integer of such magnitude is a square is $D^{-m/2}$. That is very small for the proposed parameters in Section V. The probability of (8) is around $2\,|\text{Norm}_{K/\mathbb{Q}}(2c_{nn})|^{-1}$, that is of magnitude $|2c_{nn}|^{-m}$. We conclude that this forgery is not efficient for $c_{nn} \neq 0$. If $c_{nn} = 0$, then (6) has a root in R if and only if a divides b in R which happens with exponentially small probability too.

More generally, for a parameter l such that $1 \le l \le n-r-1$ one randomly chooses $x_{r+1}, \dots, x_{n-l}$ from R with bounded max-norms. One then tries to calculate $z_1, \dots, z_l \in R$ such that $f(x) = 0$, where $x = (x_1, x_2, \dots, x_{n-l}, z_1, \dots, z_l)$. The unknowns $z_1, \dots, z_l$ must satisfy

$$g(z_1, \dots, z_l) = 0 \tag{9}$$

for a quadratic polynomial $g(z_1, \dots, z_l)$ in l variables with coefficients from R. Since the problem is Diophantine, it is difficult to decide whether (9) is solvable or not and calculate the solutions. Even for $R = \mathbb{Z}$ an efficient algorithm to solve a general binary quadratic Diophantine equation may not exist as the minimal solution size in bits may depend exponentially in the size of input as with negative Pell equation, see [12].

D. Adapting Attack

Given signed message M, y, one may try to construct another signature y' for M. Let $x = (h|y) = (x_1, \dots, x_{n-1}, x_n)$. Therefore $z = x_n$ is a root in R of the quadratic equation (6). If another root

$$x'_n = -a/c_{nn} - x_n \in R,$$

then one constructs another signature M, y' as $f(x_1, \dots, x_{n-1}, x'_n) = 0$. However, $x'_n \in R$ if and only if c_{nn} divides a in R. For random a this happens with probability $|\mathrm{Norm}_{K/\mathbb{Q}}(c_{nn})|^{-1}$. This probability is of order $|c_{nn}|^{-m}$, and is very small even for moderate m. One may try to modify at least one of x_i, $r+1 \le i \le n$ in a similar way. The success probability is

$$1 - \prod_{i=r+1}^{n} (1 - |\mathrm{Norm}_{K/\mathbb{Q}}(c_{ii})|^{-1}). \tag{10}$$

It is easy to compute $\mathrm{Norm}_{K/\mathbb{Q}}$ numerically given the roots of the polynomial $q(X)$. The probability (10) is therefore easy to compute. It is very low for the parameters in Section V as $< 2^{-347}$ for 10^4 public keys C.

The adapting attack may be extended to modifying several entries of the signature. One has to solve a Diophantine equation in $l \ge 2$ variables similar to (9), where one solution is given. The parametrisation produces solutions from the field K and generally does not work for the ring R.

E. Lattice Attack

Suppose $r = 1, s = 3$ and Z is constructed by Section III-G formulae. Then $Z_2 + Z_3 = u_2$, where $u_2 \in R$ is taken randomly each time the signature is generated. Let b_{ij} denote the entries of the matrix B and let $y = (y_2, y_3, y_4) \in R^3$ be the signature computed for a hash $h \in R$. We get

$$\begin{aligned} Z_2 + Z_3 - u_2 = b_{21}h + b_{22}y_2 + b_{23}y_3 + b_{24}y_4 + b_{31}h \\ + b_{32}y_2 + b_{33}y_3 + b_{34}y_4 - u_2 = 0. \end{aligned}$$

Denote $a = (h, y_2, y_3, y_4, h, y_2, y_3, y_4, -1) \in R^9$ and $b = (b_{21}, b_{22}, b_{23}, b_{24}, b_{31}, b_{32}, b_{33}, b_{34}, u_2) \in R^9$. Then $ab = 0$ in R, where b is an unknown vector. Let $x \in \mathbb{Z}^{9m}$ be the vector of coefficients of the entries of b and $A \in \mathbb{Z}^{m \times 9m}$ is such that $Ax = 0$ is equivalent to $ab = 0$.

The vector x belongs to the kernel lattice L of rank $d = 8m$ and of volume $V = \sqrt{\det AA^T}$. For parameters in Section V, the average Euclidean norm of x significantly exceeds the expected norm $\sqrt{d/2\pi e}\, V^{1/d}$ of the shortest non-zero vector in L, see [13]. For instance, for the parameters in range of those in Section V we got that $d = 512, V = 2,44 \cdot 10^{269}$, where the volume was averaged over 100 samples of a. The expected norm of the shortest non-zero vector in L was 18.38, but the norm of the sought vector x was ≥ 29.59 for 10^4 random choices of b. Therefore, solving SVP in L does not seem applicable to recover x and therefore b. Similar holds for another equation $Z_4 + Z_1 = u_1 u_2$, where $u_1, u_2 \in R$ are taken randomly each time the signature is generated.

V. Proposed parameters

In this section we propose parameters for DEFI-128 which is of 128-bit security level (NIST security category 1, [11]). Let $n = 4, s = 3, r = 1, m = 64$. We set $q = q(X) = X^m + 1$ which is an irreducible polynomial in $\mathbb{Z}[X]$. That defines the ring $R = \mathbb{Z}[X]/(q)$. That is the ring of integers of the cyclotomic algebraic number field $K = \mathbb{Q}(\alpha)$, where α is a root of $q(X)$.

The entries of $B_{21} \in R^{3\times 1}, B_{22} \in R^{3\times 3}$, and $u_1, u_2 \in R$ are taken randomly according to Section III. We now define a number of parameters which affect both the security and the efficiency of the scheme. The entries of B_{21} are represented by polynomials modulo $q(X)$ with coefficients $0, \pm 1, \pm 2$, where the number of non-zero coefficients is $\lambda_{B_{21}}$. The matrix B_{22} is constructed by (4). That is as a product of k random elementary matrices E_i and random permutation matrices P_i and a matrix F defined below; the latter itself may be decomposed into a product of elementary matrices.

$$\begin{aligned} F &= \begin{pmatrix} 1 & -y & 0 \\ x & 1 & y \\ 0 & x & 1 \end{pmatrix} \\ &= \begin{pmatrix} 1 & 0 & 0 \\ x & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & x & 1 \end{pmatrix} \begin{pmatrix} 1 & -y & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}. \end{aligned} \tag{11}$$

The non-zero entries of elementary matrices E_i and of the matrix F are represented by polynomials modulo $q(X)$ with coefficients $0, \pm 1$. Let λ_E and λ_F denote the number of terms with non-zero coefficients ± 1 in those representations. Lastly, u_1, u_2 (see Section 3.7, where we choose $u_2 = 2v_2$) are represented by polynomials modulo $q(X)$ with coefficients $0, \pm 1, \pm 2$, where the number of non-zero entries is λ_{u_1,u_2}. We summarise the values of the above parameters in Table I. By construction, guessing each of the entries of B_{21} takes

TABLE I
DEFI-128 parameters

n	s	r	m	k	$\lambda_{B_{21}}$	λ_E	λ_F	λ_{u_1,u_2}
4	3	1	64	14	24	1	17	35

around 2^{106} trials. Guessing F in the construction of B_{22} takes $2^{134.59}$ trials and guessing each of u_1 or u_2 takes $2^{130.27}$ trials.

We check if each entry of B_{22} has a guessing complexity of at least 2^{106} to be considered a valid B_{22}. If this condition is not met, then B_{22} is regenerated. The metric used to evaluate the guessing complexity is very conservative as it assumes an adversary has knowledge of the coefficients and their frequencies in the polynomial (which obviously is not the case). If the unique coefficients are listed as $c_1, c_2, ..., c_t$ and f_i refers to the frequency of c_i, then the guessing complexity is expressed as $\frac{m!}{f_1! \times ... \times f_k!}$. After guessing one entry of B (say b_{22}) the adversary may try to recover b_{32}, b_{42} from $c = c_{22} - b_{22}^2 = -b_{32}^2 - b_{42}^2$ by solving an instance of SVP in a lattice of rank $2m$ and of volume $V = \text{Norm}_{K/\mathbb{Q}}(c)$. The last calculation may be conservatively estimated by $m^3 \log_2^2 V$ binary operation. Recovering b_{22}, b_{32}, b_{42} thus takes $> 2^{143}$ binary operations.

We now introduce bound parameters: $\delta_{C_1}, \delta_{C_2}, \delta_{C_3}$ and δ_y. These are the bounds on the norms of the entries in the blocks of the public key matrix C and the signature y, the entities are to be strictly smaller than those parameters. In the construction of C and y, if entries exceed their bounds then they are regenerated. In the verification, the signature is rejected if $|y| \geq \delta_y$. The blocks of $C \in R^{(r+s)\times(r+s)}$ to which these bounds apply are:

sizes	r	s
r	C_1	C_2
s	C_2	C_3

The parameters were chosen as such to result in an efficient implementation with minimized public key and signature size, they are provided in Table II.

TABLE II
BOUND PARAMETERS FOR DEFI-128

δ_{C_1}	δ_{C_2}	δ_{C_3}	δ_y
2^6	2^8	2^{10}	2^{17}

With the given parameters, the performance is summarized in Table III. The secret key is simply a seed for the random number generator to generate the secret key matrices B_{21} and B_{22}. The 128-bit security for $h = \text{HASH}(M) \in R$ is achieved by ensuring that the digest is represented as a polynomial of degree < 64 with coefficients in $[-8, \ldots, 7]$. The number of such polynomials is $16^{64} = 2^{256}$. Thus the digest is 256 bits. The average time estimates (in milliseconds) are based on 10^5 iterations on a laptop with Windows 10 64-bit operating system and x64-based processor: 12th Gen Intel(R) Core(TM) i7-12800H@2.40 GHz with 16.0 GB Ram. The reference implementation for DEFI-128 adhering to the submission guidelines of [11] is available at [14]. Although an optimized implementation has not yet been developed, the timings based on the reference implementation are comparable to those from optimized implementations of some of the fastest secure digital signature schemes currently available [15].

TABLE III
PERFORMANCE OF DEFI-128

Public Key	800 bytes
Private Key	48 bytes
Signature	432 bytes
Public Key + Signature	1232 bytes
Key Generation	0.431 ms
Signature Generation	0.177 ms
Signature Verification	0.082 ms
Trials for a valid public key	1.458
Trials for a valid B_{22}	1.322
Trials for a valid signature	1.044

VI. 64-BIT CHALLENGE

We also provide a 64-bit DEFI challenge. The parameters for DEFI-64 are in Table IV and Table V. In particular, we set $R = \mathbb{Z}[X]/(X^{32} + 1)$.

TABLE IV
DEFI-64 PARAMETERS

n	s	r	m	k	$\lambda_{B_{21}}$	λ_E	λ_F	λ_{u_1,u_2}
4	3	1	32	9	9	1	7	15

TABLE V
BOUND PARAMETERS FOR DEFI-64

δ_{C_1}	δ_{C_2}	δ_{C_3}	δ_y
2^5	2^6	2^8	2^{14}

The challenge is to find an attack on the scheme that requires less than 2^{72} binary operations on a single core to deduce any of the secret entries of matrix B or to forge a signature for the hash of a message. The challenge is available at [16] and contains the following files:

- $C.txt$ - contains a public key in its uncompressed form as a JSON array. This is matrix $C \in R^{4\times4}$ with polynomial entries in each cell, stored sequentially following row-major traversal.
- $h.txt$ - contains the hash value $h \in R^1$ of 10,000 random messages with polynomial coefficients in $[-8, ..., 7]$. Each hash value is stored on a separate line, formatted as a JSON array.
- $y.txt$ - contains the signature $y \in R^3$ generated for the corresponding hash value. Each signature is stored on a separate line following row-major traversal, formatted as a JSON array.

REFERENCES

[1] G. Eason, B. Noble, and I. N. Sneddon, "Computers and Intractability: A Guide to the Theory of NP-Completeness," Series of Books in the Mathematical Sciences (1st ed.), New York: W. H. Freeman and Company, 1979.

[2] T. Matsumoto and H. Imai, "Public Quadratic Polynomial-Tuples for Efficient Signature-Verification and Message-Encryption," in Eurocrypt'88, LNCS vol. 330, pp. 419–453, Springer, 1988.

[3] J. Patarin, "Hidden Field Equations (HFE) and Isomorphisms of Polynomials (IP): two new Families of Asymmetric Algorithms," in Eurocrypt '96, LNCS vol. 1070, pp. 33–48, Springer, 1996.

[4] A. Shamir, “A polynomial-time algorithm for breaking the basic Merkle - Hellman cryptosystem,” IEEE Trans. on Information Theory. vol. 30 (1984), pp. 699–704.
[5] A. M. Odlyzko, “The Rise and Fall of Knapsack Cryptosystems,” AT&T Bell Laboratories Murray Hill, New Jersey 07974.
[6] H. Ong, C. P. Schnorr, and A. Shamir, “An efficient signature scheme based on quadratic equations,” Proc. 16th ACM Symp. Theor. Comput. (STOC’84), pp. 208–216.
[7] D. Estes, L. M. Adleman, K. Kompella, K. S. McCurley, and G. L. Miller, “Breaking the Ong-Schnorr-Shamir signature scheme for quadratic number fields,” in Crypto’85, LNCS 218, pp. 3–13, Springer, 1986.
[8] H. Yosh, “The key exchange cryptosystem used with higher order Diophantine equations,” IJNSA, vol. 3 (2011), no. 2, pp. 43–50.
[9] K. V. Prasamsa1, P. A. Kameswari, K. N. Raju, T. Surendra, and D. M. Devi, “A key exchange algorithm with binary quadratic forms to design complex security framework,” Advances in Mathematics: Scientific Journal, vol. 10 (2021), no. 1, pp. 589–595.
[10] L. J. Mordell, “Diophantine equations,” Academic Press, London and New York, 1969.
[11] National Institute of Standards and Technology, “Post-Quantum Cryptography Standardization,” NIST. [Online]. Available: https://csrc.nist.gov/projects/post-quantum-cryptography/post-quantum-cryptography-standardization. [Accessed: May 3, 2024].
[12] J. C. Lagarias, “On the computational complexity of determining the solvability or unsolvability of the equation $X^2 - DY^2 = -1$,” Trans. Amer. Math. Soc. 260 (1980), pp. 485–508.
[13] P. Q. Nguyen, “Hermite’s constant and Lattice Reduction,” The LLL Algorithm, Survey and Applications, pp. 145–178, Springer, 2010.
[14] M. Feussner, ”DEFI128,” GitHub repository, [Online]. Available: https://github.com/martinfeussner/DEFI/tree/dd038b3/DEFI128.
[15] PQShield, “NIST Signatures Zoo,” PQShield. [Online]. Available: https://pqshield.github.io/nist-sigs-zoo/. [Accessed: May 3, 2024].
[16] M. Feussner, ”DEFI64 Challenge,” GitHub repository, [Online]. Available: https://github.com/martinfeussner/DEFI/tree/dd038b3/DEFI64 Challenge.