Isotropic Quadratic Forms, Diophantine equations and Digital Signatures, DEFIv2

Martin Feussner and Igor Semaev

Selmer Center, University of Bergen, Bergen 5006, Norway
{martin.feussner,igor.semaev}@uib.no

Abstract. This work introduces DEFIv2 - 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.

Keywords: Digital signatures · Isotropic quadratic forms · Diophantine equations.

1 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 $a_1, \ldots, a_n$ and a it is to decide whether or not there exist x_i in $\{0, 1\}$ such that

$$x_1 a_1 + \ldots + x_n a_n = a.$$

This problem is known to be NP-complete [4]. Obviously, the subset sum problem is equivalent to solving (deciding) the system of multivariate quadratic Diophantine equations

$$x_1 a_1 + \ldots + x_n a_n = a,\ x_1^2 - x_1 = 0, \ldots,\ x_n^2 - x_n = 0.$$

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

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

where $x = (x_1, \ldots, x_n)$, and $f_i \in \mathbb{Z}[x]$, and $\deg f_i \leq 2$, is solvable in rational 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 DEFIv2 is presented. The security of the scheme is based on the hardness of computing isotropic vectors over commutative rings of characteristic 0 for quadratic forms with a trapdoor. Given a message, one may construct an isotropic vector for the form, where one of the entries is its digest and the rest of the entries serve as a signature.

The first version of the scheme was published on ePrint [1] and shared to the pqc-forum [2] where the scheme was broken by Henry Bambury and Phong Nguyen. The difference with the present version is in how vector Z is constructed when the signature is generated, see Section 3.7. The construction in the earlier version implies that a lattice similar to L in Section 5.5 contains a very short secret vector which may be recovered with BKZ algorithm. That leads to recovering the secret matrix B. The new version is immune to such lattice attacks as secret vectors are significantly larger than the vectors in L produced with BKZ, see Section 5.5 for details.

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 Section 4 parameters (NIST security level 1) forging a signature is equivalent to finding a relatively small solution to a nonhomogeneous quadratic Diophantine equation in 4 variables over $R = \mathbb{Z}[X]/(q)$, where (q) is the ideal in $\mathbb{Z}[X]$ generated by an irreducible polynomial $q = q(X)$ of degree $m = 28$. It is well known [8] that given one solution to a homogeneous quadratic equation over a ring it is possible to get all other solutions with parametrisation. However, this method is not applicable to nonhomogeneous equations. Also, there is a restriction on the solution size. Equivalently, one has to find a relatively small solution to a system of 28 multivariate quadratic Diophantine equations over $\mathbb{Z}$ in 84 variables to forge a signature for a given message.

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 [7] and Hidden Field Equations (HFE) [12] cryptosystems and their derivatives. Also, advances in solving common lattice problems as SVP (Shortest Vector Problem) and CVP (Closest Vector Problem) does not seem to undermine the new scheme, see Sections 5.5 and 4 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 [13]. Its variations were broken too, see [10] for a survey. Also, digital signature scheme [11] based on a quadratic congruence modulo a composite integer and its extensions were broken, see [3]. 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 [14] and [15] 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.

2 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 $\alpha_0, \alpha_1, \ldots, \alpha_{m-1}, \ldots$ 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 $|a| = \max_{0\leq i<m} |a_i|$ defines a norm on R. Also, for $y = (y_1, y_2, \ldots, y_n) \in R^n$ we set $|y| = \max_{1\leq i\leq n} |y_i|$. Let

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

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 [8]. 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 3, 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 5. Numerical parameters are proposed in Section 4, they provide 128-bit security of the scheme which corresponds to the NIST security category 1 according to [6].

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

$$J = \mathrm{Diag}\,(\pm 1, \ldots, \pm 1, \pm 1),$$

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

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

where $C = B^TJB \in R^{n\times n}$, is an isotropic quadratic form. For matrices B specified in Section 3 isotropic vectors are easy to calculate.

3 Signature Scheme

3.1 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 3.6 may be used.

3.2 Public Key

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

3.3 Signature Generation

Let M be a message and $h \in R^r$ encodes its hash value. One may take the entries of h of relatively 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 $Z^T JZ = 0$, where $Z = (Z'|Z'') \in R^n$. See Section 3.7, where the construction is specified for $r = 1, s = 3$.
3. Compute $y \in R^s$ by
$$y = B_{22}^{-1}\left(Z'' - B_{21}h\right).$$
4. The signature for M is y.

In the variation of the scheme presented in Section 4, an extra parameter γ_y is used. The generated signature y is correct if additionally $|y| < \gamma_y$.

3.4 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
$$f(z) = z^T Cz = 0,$$
then accept the signature, otherwise reject.

In the variation in Section 4, the signature is rejected if $|y| \geq \gamma_y$ as well.

3.5 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,

$$f(z) = z^T C z = [Bz]^T J [Bz] = Z^T J Z = 0.$$

3.6 How to Generate B_{22}

One may set

$$B_{22} = (\prod_{i=1}^{k} P_i E_i)\, F \tag{4}$$

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 4. Then

$$B_{22}^{-1} = F^{-1}(\prod_{i=1}^{k} E_{k-i+1}^{-1} P_{k-i+1}^{-1}).$$

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

3.7 How to Generate Z

In this section we 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 $J = \mathrm{Diag}(1, 1, -1, -1)$. Then

1. Let $(v_1, v_2, v_3, v_4) = \mathrm{HASH}(M) \in R^4$. Compute $h = v_1 v_4 - v_2 v_3 \in R$.
2. Generate randomly $(a_1, a_2, a_3, a_4), (d_1, d_2, d_3, d_4) \in R^4$ such that

$$a_1 a_4 - a_2 a_3 = \det \begin{pmatrix} a_1 & a_2 \\ a_3 & a_4 \end{pmatrix} = 1,$$

$$d_1 d_4 - d_2 d_3 = \det \begin{pmatrix} d_1 & d_2 \\ d_3 & d_4 \end{pmatrix} = 1.$$

   Similar to (4), the matrices $A = \begin{pmatrix} a_1 & a_2 \\ a_3 & a_4 \end{pmatrix}$ and $D = \begin{pmatrix} d_1 & d_2 \\ d_3 & d_4 \end{pmatrix}$ may be formed as products of randomly generated elementary and permutation matrices.

3. Compute
$$\begin{pmatrix} V_1 & V_2 \\ V_3 & V_4 \end{pmatrix} = \begin{pmatrix} d_1 & d_2 \\ d_3 & d_4 \end{pmatrix} \begin{pmatrix} v_1 & v_2 \\ v_3 & v_4 \end{pmatrix} \begin{pmatrix} a_1 & a_2 \\ a_3 & a_4 \end{pmatrix}.$$
4. Set
$$\begin{aligned} Z_1 = h &= V_1V_4 - V_2V_3, \\ Z_2 &= V_1V_2 + V_3V_4, \\ Z_3 &= V_1V_2 - V_3V_4, \\ Z_3 &= V_1V_4 + V_2V_3, \end{aligned}$$
and $Z = (Z_1, Z_2, Z_3, Z_4)$. Therefore,
$$\begin{aligned} Z^TJZ &= Z_1^2 + Z_2^2 - Z_3^2 - Z_4^2 \\ &= (V_1V_4 - V_2V_3)^2 + (V_1V_2 + V_3V_4)^2 - (V_1V_2 - V_3V_4)^2 - (V_1V_4 + V_2V_3)^2 \\ &= 0. \end{aligned}$$

4 Proposed Parameters and Performance

In this section, we propose parameters for the scheme that correspond to the NIST security category 1 level [6]. We refer to this scheme as DEFIv2-1. Let $r = 1, s = 3, n = 4$ and $m = 28$. We set $q = q(X) = X^m + X + 1$ which is an irreducible polynomial in $\mathbb{Z}[X]$. That defines the ring $R = \mathbb{Z}[X]/(q)$, which corresponds to a ring of integers in the algebraic number field $K = \mathbb{Q}(\alpha)$, where α is a root of $q(X)$.

The matrix $B_{22} \in R^{3\times3}$ is constructed by (4). That is as a product of k_B random elementary (with only 1 non-zero off-diagonal entry containing only 1 non-zero coefficient ±1) and random permutation matrices and a matrix F; the latter itself may be decomposed into a product of elementary matrices and is defined as
$$F = \begin{pmatrix} 1 & -y & 0 \\ x & 1 & y \\ 0 & x & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ x & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & x & 1 \end{pmatrix} \begin{pmatrix} 1 & -y & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$
The variables x, y have coefficients taken randomly from $[-\delta_F, \delta_F] \setminus \{0\}$ and the entries of $B_{21} \in R^{3\times1}$ have coefficients taken randomly from $[-\delta_{B_{21}}, \delta_{B_{21}}] \setminus \{0\}$. The construction of A and D follows similarly as a product of only k_{AD} random elementary and random permutation matrices and no F.

To generate a valid B_{22}, we ensure that each entry meets a minimum guessing complexity, 2^{Ω_B}, before accepting it. If an entry fails this criterion, B_{22} is regenerated. By construction, the polynomial coefficients of B_{22}-entries are in $[-\gamma_{B_{22}} + 1, \gamma_{B_{22}} - 1]$. We use a metric to estimate the guessing complexity by computing the inverse probability of each polynomial entry occurrence. That is a product of the precomputed inverse probabilities (likelihoods) of its coefficient. The likelihood values are stored as rounded down base-2 integers in the implementation for efficiency — for example, a probability of $\frac{1}{123}$ is stored as 6. In total, 2^{25} of B_{22} were generated at random to build and validate these tables.

We now introduce bound parameters: $\gamma_{C_1}, \gamma_{C_2}, \gamma_{C_3}, \gamma_{B_{22}}, \gamma_{B_{22}^{-1}}, \gamma_y$. These are strict bounds on the absolute value of the polynomial coefficients in the blocks of the public key matrix C, the secret matrices B_{22} and B_{22}^{-1}, and the signature y respectively. The blocks of $C \in R^{n\times n}$ to which these bounds apply are:

sizes	r	s
r	C_1	C_2
s	C_2	C_3

Any of $C, B_{22}, B_{22}^{-1}, y$ are regenerated if coefficients exceed the bound. This also imposes a signature rejection condition if y does not satisfy the bound.

We also provide the parameter d, which represents the output size of FIPS202-SHAKE256 in bits that the scheme operate with. The choice of d allows for each polynomial coefficient in $(v_1, v_2, v_3, v_4) = \mathrm{HASH}(M) \in R^4$ to be represented by a whole number $d/4m$ of bits. Also, that takes into account increased collisions further explained in Section 5. Each coefficient value is initially in the range $[0, 2^{d/4m} - 1]$. To center the values around zero, we take them in the range $[-2^{d/4m-1}, 2^{d/4m-1} - 1]$. The final hash representation $h = v_1v_4 - v_2v_3 \in R$ is then computed from this.

The values for the parameters were chosen to result in an efficient implementation with minimized public key and signature size. The parameters are provided Table 1.

Table 1. DEFIv2-1 parameters

m	n	s	r	k_B	k_{AD}	δ_F	$\delta_{B_{21}}$	Ω_B	γ_{C_1}	γ_{C_2}	γ_{C_3}	$\gamma_{B_{22}}$	$\gamma_{B_{22}^{-1}}$	γ_y	d
28	4	3	1	13	10	4	8	112	2^{11}	2^{12}	2^{15}	2^7	2^{11}	2^{45}	336

The performance with the provided parameters is summarized in Table 2. The secret key is a seed for the random number generator to generate the secret key matrix B_{21} and byte packing of the secret key matrix B_{22}^{-1}. The average time estimates (in milliseconds) are based on 10^4 iterations, compiled with the `-O3` optimization flag, 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 DEFIv2-1 follows the submission guidelines in [6] and is available at [17]. Although an optimized implementation has not yet been developed, the performance metrics of the reference implementation (with the `-O3` optimization flag) in Table 2 are comparable to those of optimized implementations of some of the fastest secure digital signature schemes currently available or proposed [16].

Table 2. Performance of DEFIv2-1

Public key	515 bytes
Private key	426 bytes
Signature	483 bytes
Public key + signature	998 bytes
Key generation	0.902 ms
Signature generation	0.126 ms
Signature verification	0.054 ms
Average trials for valid B	2.708
Average trials for valid C	2.095
Average trials for valid signature	1.003
Expected maximum trials for valid signature	3

5 Cryptanalysis

The security of the scheme depends on the basis ring R not counting the parameters r, s, n. In what follows we set $R = \mathbb{Z}[X]/(q)$, where (q) is the ideal in $\mathbb{Z}[X]$ generated by a monic irreducible polynomial $q = q(X)$ of degree m with integer coefficients. Let $|a|, a \in R$ be the maximum in absolute values of the coefficients of a polynomial of degree $< m$ which represents a modulo $q(X)$. We 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 $q(X)$.

For DEFIv2-1 we need to provide a 128-bit security for the message M hash value representation, $h = v_1v_4 - v_2v_3 \in R$ where $(v_1, v_2, v_3, v_4) = \mathrm{HASH}(M) \in R^4$. That is achieved by ensuring that v_1, v_2, v_3, v_4 are represented as polynomials of degree < 28 with coefficients in $[-4, 3]$. The number of such polynomial tuples is $8^{4\cdot 28} = 2^{336}$. This hash value representation introduces an additional layer of collisions: while a standard collision would occur when two different messages M_1 and M_2 produce the same (v_1, v_2, v_3, v_4), the form $h = v_1v_4 - v_2v_3$ allows for distinct digests to result in the same h. For instance, changing signs and permuting some of (v_1, v_2, v_3, v_4) results in the same h. The distribution of the coefficients of the polynomial h, observed experimentally, implies that the number of different h is over 2^{256}.

5.1 Private Key Recovery

Given public matrix C recover a matrix $B \in R^{n\times n}$ such that $C = B^T JB$. That equation may be written as a system of $(n^2 + n)/2$ quadratic Diophantine equations in $n(n - r)$ (we assume that B_{11} is public) variables, the entries of B_{21} and B_{22}, over R. That is generally hard to solve. If the entries of B are represented by very sparse polynomials a guessing strategy may work to recover them. For the proposed parameters, we avoid that by ensuring that each entry meets some minimum guessing complexity. For B_{21} this is straightforward, there are 2^{112} possibilities for each entry. For B_{22}, we expect there to be more than

2^{112} possibilities for each entry based on the chosen metric. An adversary, after correctly guessing one entry from a column of B (say b_{22}) may then attempt to recover b_{32} and b_{42} from $c = b_{22}^2 - c_{22} = b_{32}^2 + b_{42}^2$ by solving an instance of SVP in a lattice of rank $2m$ and of volume $V = \mathrm{Norm}_{K/\mathbb{Q}}(c)$. The last calculation may be conservatively estimated by $(2m)^3 \log_2^2 V$ binary operation. From experiments using over 2^{20} such c generated with our proposed parameters, we get that $V > 2^{132}$. Recovering b_{22}, b_{32}, b_{42} should thus takes $> 2^{143}$ binary operations.

5.2 Forgery Attack over $\mathbb{Z}$

One may write the form (3) as

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

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

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

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 1.

5.3 Forgery Attack over $\boldsymbol{R}$

Let M be a message with the hash $h \in R^r$. In order to forge a signature one sets $(x_1, \ldots, x_r) = h$, and 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 $x = (x_1, \ldots, x_r, x_{r+1}, \ldots, x_{n-1}, z)$. That is

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

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 $c_{nn} \ne 0$, then z satisfies the quadratic equation

$$c_{nn}z^2 + az + b = 0 \tag{6}$$

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

$$v = a^2 - 4bc_{nn} = u^2 \tag{7}$$

for some $u \in R$, and

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

We will estimate 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 $\mathrm{Norm}_{K/\mathbb{Q}}(v)$ is a square. The maximum of that norm is of magnitude D^m. The probability that an integer of such magnitude is a square is $D^{-m/2}$. For the proposed parameters in Section 4, we ran experiments to estimate D. We randomly generated 2^{10} public keys, and for each, 2^{10} random $h, x_1, \ldots, x_{n-1}$ with $x_1, \ldots, x_{n-1}$ having coefficients in $[-1, 1]$. The minimum entry obtained from all $|b^2 - 4dc_{nn}|$ was $2^{15.96}$. So we conservatively estimate $D^{-m/2} \ll 2^{-223.44}$ which is very small.

The probability of (8) is around $2\,|\mathrm{Norm}_{K/\mathbb{Q}}(2c_{nn})|^{-1}$, that is of magnitude $|2c_{nn}|^{-m}$. For the proposed parameters, the value of $2\,|\mathrm{Norm}_{K/\mathbb{Q}}(2c_{nn})|^{-1}$ from 2^{16} randomly generated public keys was $\leq 2^{-201.20}$. 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 $a|b$ in R which happens with exponentially small probability too. Similar holds for other c_{ii}.

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

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

for a quadratic polynomial $g(z_1, \ldots, z_l)$ 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 [5].

5.4 Adapting Attack

Given signed message M, y, one may try to construct another signature y' for M. Let $x = (h|y) = (x_1, \ldots, x_{n-1}, x_n)$. 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 $f(x_1, \ldots, x_{n-1}, x_n') = 0$. 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, $r+1 \leq i \leq n$ 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 and the maximum probability obtained using 2^{16} randomly generated C was $2^{-165.48}$ for the parameters in Section 4.

The adapting attack may be extended to modifying several entries of the signature. One has to solve a Diophantine equation in $l \geq 2$ 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.

5.5 Lattice Attack

Suppose $r = 1, s = 3$ and $Z'' \in R^3$ is constructed by Section 3.7 formulae. Every signature $y \in R^3$ results in one equation

$$\big(B_{21}|B_{22}\big)\begin{pmatrix} h \\ y \end{pmatrix} = Z'',$$

where h is constructed from the hash of the message and $\big(B_{21}|B_{22}\big) \in R^{3\times 4}$ is the scheme secret key. Given N signatures $M_i, y_i, i = 1, \ldots, N$, one may form a matrix

$$H = \begin{pmatrix} h_1 \; h_2 \; \ldots \; h_N \\ y_1 \; y_2 \; \ldots \; y_N \end{pmatrix} \in R^{4\times N},$$

where h_i are constructed from the hash of M_i. Let here

$$Z = \big(Z''_1 \; \ldots \; Z''_N\big) \in R^{3\times N}.$$

Then $\big(B_{21}|B_{22}\big)\,H = Z$ and so $\big(B_{21}|B_{22}|Z\big) = \big(B_{21}|B_{22}\big)\,\big(I_4|H\big)$, where I_4 is a unity (4×4)-matrix. The rows of $\big(B_{21}|B_{22}|Z\big)$ belong to a module generated over R by the rows of $\big(I_4|H\big)$. One may construct an integer $(4m \times (4+N)\,m)$-matrix the rows of which represent over $\mathbb{Z}$ the rows of $\big(I_4|H\big)$. Let L be a lattice of rank $4m$ generated by the rows of that matrix. Since the rows of $\big(B_{21}|B_{22}|Z\big)$, after transforming into a $(3m \times (4+N)\,m)$-matrix over $\mathbb{Z}$, have relatively small entries compared with the rows of $\big(I_4|H\big)$ and they belong to L, one may try to apply a lattice reduction algorithm to recover some or all of them.

However, experimentally, with 2^9 randomly generated secret keys used to sign $1 \leq N \leq 2^5$ random messages, with the parameters in Section 4, the largest vector v_l in a LLL reduced basis of L was significantly shorter than the shortest row-vector v_s in the target matrix $\big(B_{21}|B_{22}|Z\big)$. More precisely, $\frac{\|v_s\|}{\|v_l\|} > 3.34$, where $\|\cdot\|$ denotes the Euclid norm of a vector. So, the rows of $\big(B_{21}|B_{22}|Z\big)$ should be impossible to recover directly from the reduced basis. Using BKZ generally makes the reduced basis even smaller and therefore won't help to recover the secret.

6 DEFI Challenge

Here we provide details to our publicly available 90-bit challenge for DEFIv2 which we shall call DEFIv2-c. The parameters for this challenge are listed in Table 3. In particular, we set $R = \mathbb{Z}[X]/(X^{16} + X + 1)$.

Table 3. DEFIv2-c parameters

m	n	s	r	k_B	k_{AD}	δ_F	$\delta_{B_{21}}$	Ω_B	γ_{C_1}	γ_{C_2}	γ_{C_3}	$\gamma_{B_{22}}$	$\gamma_{B_{22}^{-1}}$	γ_y	d
16	4	3	1	9	9	4	8	63	2^{10}	2^{11}	2^{12}	2^6	2^9	2^{42}	256

The challenge is to find an attack on the scheme that requires less than 2^{90} binary operations to deduce any of the secret entries of matrix B or to forge a signature for the hash of a message. The challenge files contain data collected from signing 2^{14} randomly generated messages using one key-pair and is available at [18]. It contains the following files which are formatted as JSON arrays:

- `C.txt` - contains a public key matrix $C \in R^{4\times 4}$ in its uncompressed form.
- `v.txt` - contains the hash of a message as $(v_1, v_2, v_3, v_4) \in R^4$.
- `h.txt` - contains the hash value representation $h = v_1v_4 - v_2v_3 \in R$.
- `y.txt` - contains the signature $y \in R^3$ in its uncompressed form.
- `z.txt` - contains $z = (h|y) \in R^4$ from the signature verification step.

Acknowledgments. The authors have no acknowledgments to make.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Feussner, M., Semaev, I.: Isotropic Quadratic Forms, Diophantine Equations and Digital Signatures. Cryptology ePrint Archive, Paper 2024/679. Available at: https://eprint.iacr.org/2024/679 (2024)
2. Feussner, M., Semaev, I.: pqc-forum: New Digital Signature Scheme - DEFI. Available at: https://groups.google.com/a/list.nist.gov/g/pqc-forum/c/x7-nf3NuYTs/m/dGvflCePAQAJ (2024)
3. Estes, D., Adleman, L.M., Kompella, K., McCurley, K.S., Miller, G.L.: Breaking the Ong-Schnorr-Shamir signature scheme for quadratic number fields. In: Advances in Cryptology – Crypto'85, LNCS, vol. 218, pp. 3–13. Springer, Heidelberg (1986)
4. Garey, M.R., Johnson, D.S.: Computers and Intractability: A Guide to the Theory of NP-Completeness. Series of Books in the Mathematical Sciences, W. H. Freeman and Company, New York (1979)
5. Lagarias, J.C.: On the computational complexity of determining the solvability or unsolvability of the equation $X^2 - DY^2 = -1$. Trans. Amer. Math. Soc. 260, 485–508 (1980)

6. National Institute of Standards and Technology (NIST): Post-Quantum Cryptography Standardization. Available at: https://csrc.nist.gov/projects/post-quantum-cryptography/post-quantum-cryptography-standardization (accessed 2024)
7. Matsumoto, T., Imai, H.: Public Quadratic Polynomial-Tuples for Efficient Signature-Verification and Message-Encryption. In: Advances in Cryptology – Eurocrypt'88, LNCS, vol. 330, pp. 419–453. Springer, Heidelberg (1988)
8. Mordell, L.J.: Diophantine equations. Academic Press, London and New York (1969)
9. Nguyen, P.Q.: Hermite's constant and Lattice Reduction. In: The LLL Algorithm, Survey and Applications, pp. 145–178. Springer-Verlag, Heidelberg (2010)
10. Odlyzko, A.M.: The Rise and Fall of Knapsack Cryptosystems. AT&T Bell Laboratories, Murray Hill, New Jersey (1984)
11. Ong, H., Schnorr, C.P., Shamir, A.: An efficient signature scheme based on quadratic equations. In: Proceedings of the 16th ACM Symposium on Theory of Computing (STOC'84), pp. 208–216 (1984)
12. Patarin, J.: Hidden Field Equations (HFE) and Isomorphisms of Polynomials (IP): two new Families of Asymmetric Algorithms. In: Advances in Cryptology – Eurocrypt '96, LNCS, vol. 1070, pp. 33–48. Springer, Heidelberg (1996)
13. Shamir, A.: A polynomial-time algorithm for breaking the basic Merkle-Hellman cryptosystem. IEEE Trans. on Information Theory 30, 699–704 (1984)
14. Yosh, H.: The key exchange cryptosystem used with higher order Diophantine equations. IJNSA 3(2), 43–50 (2011)
15. Prasamsa, K.V., Kameswari, P.A., Raju, K.N., Surendra, T., Devi, D.M.: A key exchange algorithm with binary quadratic forms to design complex security framework. Advances in Mathematics: Scientific Journal 10(1), 589–595 (2021)
16. PQShield: NIST Signature Zoo. Available at: https://pqshield.github.io/nist-sigs-zoo/ (accessed 2024)
17. Feussner, M: DEFIv2-1 [GitHub repository]. Available at: https://github.com/martinfeussner/DEFIv2/tree/main/DEFIv2-1 (accessed 2024)
18. Feussner, M: DEFIv2-c [GitHub repository]. Available at: https://github.com/martinfeussner/DEFIv2/tree/main/DEFIv2-c (accessed 2024)