New Digital Signature Algorithm EHTv2

Igor Semaev

Department of Informatics, University of Bergen, Norway
igor@ii.uib.no

Abstract. Every public-key encryption/decryption algorithm where the set of possible plain-texts is identical to the set of possible cipher-texts may be converted into a digital signature algorithm. That is quite different in the lattice (code)-based public-key cryptography. The decryption algorithm on a random input produces a valid plain-text, that is a signature, with a negligible probability. That explains why it is so difficult to construct a new secure and efficient lattice-based digital signature system. Though several solutions are known and taking part in the NIST Post Quantum Standardisation Process there is still a need to construct digital signature algorithms based on new principles. In this work, a new and efficient digital signature algorithm is suggested. Its design is simple and transparent. Its security is based on the hardness of an approximate closest vector problem in the maximum norm for some q-ary lattices. The signature is several times shorter than that provided by the NIST Selected Digital Signature Algorithms with comparable security level, while the public key size is larger.

Keywords: Linear inequalities · Matrices · Digital signatures· q-ary lattices.

1 Introduction

Digital signatures are an important area of applications for public-key cryptography. Every public-key encryption/decryption algorithm, where the set of possible plain-texts is identical to the set of possible cipher-texts, may be converted into a digital signature algorithm. The most notable examples are RSA and Rabin crypto-systems. That is quite different in lattice(code)-based and multivariate cryptography. The cipher-text is there larger than the plain-text as in NTRU and Regev's LWE based crypto-systems. The decryption algorithm on a random input produces a valid plain-text, that is a signature, with a negligible probability. That explains why it is so difficult to construct a new secure and efficient lattice-based digital signature system. Though several algorithms as GGH and some of NTRU-based were broken in [17, 5], yet another NTRU-based signature algorithm variation Falcon is among the finalists of the NIST Post Quantum Standardisation Process, [15]. Similarly, several variations of multivariate algorithms as HFE and TTM were broken [12, 10], and another multivariate signature algorithm Rainbow is among the finalists of the NIST competition. The history of the attacks and relevant countermeasures provides a better understanding

of the security of the cryptographic algorithms. However, the countermeasures make the resulting algorithms patchy and non-transparent, one may not feel certain about their security. So there is still a need to construct new digital signature algorithms. A new construction may improve the efficiency parameters compared with known solutions.

In the present work, a new and efficient digital signature primitive (hash-and-sign) is suggested. The design of the signature algorithm is simple and transparent. The security is based on the hardness of an approximate closest vector problem (CVP) for some specific q-ary lattices in the maximum norm. One proves that the signature is uniformly distributed if the hashing algorithm provides a uniform distribution on its outputs. The signature is several times shorter than that provided by the NIST Selected Digital Signature Algorithms with comparable security level, while the public key size is larger.

There are three approaches to the cryptanalysis of the new algorithm. First, find the private key given a public key only. Second, forge signatures without the knowledge of the private key. Third, find the private key or forge a new signature by analysing a number of valid signatures. We claim that it is hard to forge a valid signature for any given message as one will need to solve a hard CVC problem for some specific q-ary lattice. The cryptanalysis is presented in Section 5.

Published digital signature lattice-based constructions typically make use of short lattice bases as private keys and their random non-short perturbations as public keys. That is true for GGH [6], some its modifications as DRS, see [20], and NTRU-based signature algorithms as NTRUSign in [4]. Another approach based on the hardness of the SIS (Short Integer Solution) problem was implemented in [11],[16]. The present construction does not use neither short bases of relevant lattices, nor the hardness of the SIS problem.

The new digital signature algorithm does not have so far a so-called security proof, the proof that it stands all attacks by a reduction to an NP-hard problem or some hard computational problem in general lattices, etc. That is not uncommon in the field. The most notable example is the RSA crypto-system. We do not know if breaking the RSA results in fast integer factorisation. Another example is a multivariate signature algorithm Rainbow, a round 3 NIST candidate, which does not have a security proof. One of the NIST selected algorithms Falcon provides a reduction to the NTRU problem, which is the shortest vector problem for a very particular lattice. The NTRU problem was around for more than 25 years. Only recently a reduction-based evidence of its hardness was published in [19]. Similarly, a reduction-based security argument for the underlying problem of the new digital algorithm may require more time and effort.

The new digital signature algorithm has some similarity with the crypto-system EHT in [2]. It is impossible to use this crypto-system directly to generate signatures as the length of the cipher-text is larger than the length of the plain-text. Nevertheless, we call the new digital signature algorithm EHT too.

2 Signature Algorithm

In this section a basic version of the new signature algorithm is explained. The algorithm consists of private and public key generating algorithms, signature generating and verifying algorithms. They all are presented in this section along with the signature verification proof. Let q, n, k, d be positive integers. Suppose M is a message to sign and let $h = \mathrm{HASH}(M) \in \mathbb{Z}_q^{kn}$ denote a hash of M.

The signature for M is $x \in \mathbb{Z}_q^{n+d}$ such that $h = Ax + e$ for some public matrix $A \in \mathbb{Z}_q^{kn \times (n+d)}$ and a vector $e \in \mathbb{Z}_q^{kn}$. The entries of e represented as integers are bounded in absolute values. A detailed description of the algorithm is in this section below.

To forge a signature for a message M without knowledge of the private key one has to solve the following problem. Given $h \in \mathbb{Z}_q^{kn}$, where $h = \mathrm{HASH}(M)$, find $x \in \mathbb{Z}_q^{n+d}$ such that the entries of $h - Ax$ taken as integers are bounded in absolute values. We have not found an efficient method to solve the problem without inverting the hash function.

In Section 3, we prove that the signature x is uniformly distributed if the hash function provides a uniform distribution on $\mathbb{Z}_q^{kn}$. So in the random oracle model (the hash function is a random oracle) the signature algorithm itself is a random oracle.

2.1 Parameters

Let q, n, k, d, c be positive integers, where q, k, d, c are relatively small and n is up to several hundreds. Also, $h = \mathrm{HASH}(M)$ is a hash value of the message M, encoded by a vector in $\mathbb{Z}_q^{kn}$. The integer q defines the arithmetic of the scheme while n, d affect its security and efficiency.

2.2 Private Key

The private key consists of three matrices R, B, C.

1. The matrix R is an integer $(kn + d) \times (n + d)$ matrix as

$$R = \begin{pmatrix} T & 0 \\ R_1 & R_2 \end{pmatrix},$$

where

$$T = \begin{pmatrix} t_{11} & 0 & \dots & 0 \\ t_{21} & 0 & \dots & 0 \\ t_{k1} & 0 & \dots & 0 \\ * & t_{12} & \dots & 0 \\ * & t_{22} & \dots & 0 \\ * & t_{k2} & \dots & 0 \\ * & * & \dots & t_{1n} \\ * & * & \dots & t_{2n} \\ * & * & \dots & t_{kn} \end{pmatrix}, \tag{1}$$

is a matrix of size $kn \times n$, and the entries $t_{1j}, t_{2j}, \ldots, t_{kj}$ are called diagonal. The entries of T below and to the left of the diagonal are denoted by $*$, they are secret and may be chosen randomly.
Each tuple $[t_{1j}, t_{2j}, \ldots, t_{kj}]$ has to satisfy the following property. All entries are non-zero residues modulo q and at least one is coprime to q. For any integer $b_1, b_2, \ldots, b_k$ there is an integer u such that

$$\begin{aligned} |(b_1 - t_{1j}\, u) \bmod q| &\le c, \\ |(b_2 - t_{2j}\, u) \bmod q| &\le c, \\ &\ldots, \\ |(b_k - t_{kj}\, u) \bmod q| &\le c. \end{aligned} \tag{2}$$

For q and k used to construct signatures in this work all such tuples may be found by brute force. Let, for instance, $q = 61, k = 3, c = 8$. There is only one tuple $[t_1, t_2, t_3] = [1, 4, 15]$ modulo q up to a permutation of entries, multiplication the tuple by a residue coprime to q and changing the sign of the entries such that for any integers b_1, b_2, b_3 the system of inequalities $|(b_1 - t_1 u) \bmod 61| \le 8, |(b_2 - t_2 u) \bmod 61| \le 8, |(b_k - t_3 u) \bmod 61| \le 8$ has a solution u.
The matrix R_1, R_2 are arbitrary matrices of size $d \times n$ and of size $d \times d$ respectively. The matrix R_2 is invertible modulo q.

2. The matrix C is a $kn \times (kn + d)$ -matrix with integer entries as

$$C = \begin{pmatrix} 1 & 0 \ldots 0 * \ldots * \\ 0 & 1 \ldots 0 * \ldots * \\ \ldots & \\ 0 & 0 \ldots 1 * \ldots * \end{pmatrix}. \tag{3}$$

The first kn columns of C contain the unity matrix of size $kn \times kn$. The entries of the right d columns are secret and may be randomly generated.

3. The matrix B is an arbitrary integer $(n + d) \times (n + d)$ -matrix invertible modulo q.

Theorem 1. *For every integer vector $\bar{a} = (a_1, a_2, \ldots, a_{kn})$ there exist an integer vector $\bar{y} = (y_1, y_2, \ldots, y_n)$ and an integer vector $\bar{z} = (z_1, z_2, \ldots, z_{kn})$, where*

$$|z_i| \le c, \quad i = 1, \ldots, kn,$$

such that $\bar{a} \equiv T\bar{y} + \bar{z} \mod q$.

Proof. First, we show how to compute iteratively y_j and $z_{(j-1)k+1}, \ldots, z_{jk}$ for $j = 1, \ldots, n$. For $j = 1$ we set

$$\begin{pmatrix} b_1 \\ b_2 \\ \ldots \\ b_k \end{pmatrix} = \begin{pmatrix} a_1 \\ a_2 \\ \ldots \\ a_k \end{pmatrix}.$$

and $y_1 = u$, where u is a solution to the system of inequalities (2). Then

$$\begin{pmatrix} z_1 \\ z_2 \\ \dots \\ z_k \end{pmatrix} \equiv \begin{pmatrix} b_1 \\ b_2 \\ \dots \\ b_k \end{pmatrix} - \begin{pmatrix} t_{11} \\ t_{21} \\ \dots \\ t_{k1} \end{pmatrix} y_1 \mod q.$$

The entries of the left hand side vector are bounded by c in absolute value by (2). Let T_j be a sub-matrix of T of size $k \times j$ in the rows $jk+1, jk+2, \dots, jk+k$ and columns $1, \dots, j$, where $1 \le j \le n-1$. The entries of T_j are denoted by $*$ in the definition of T. For $j > 1$ we set

$$\begin{pmatrix} b_1 \\ b_2 \\ \dots \\ b_k \end{pmatrix} \equiv \begin{pmatrix} a_{(j-1)k+1} \\ a_{(j-1)k+2} \\ \dots \\ a_{jk} \end{pmatrix} - T_{j-1} \begin{pmatrix} y_1 \\ \dots \\ y_{j-1} \end{pmatrix} \mod q.$$

Then $y_j = u$, where u is a solution to the system of inequalities (2). So

$$\begin{pmatrix} z_{(j-1)k+1} \\ z_{(j-1)k+2} \\ \dots \\ z_{jk} \end{pmatrix} \equiv \begin{pmatrix} b_1 \\ b_2 \\ \dots \\ b_k \end{pmatrix} - \begin{pmatrix} t_{1j} \\ t_{2j} \\ \dots \\ t_{kj} \end{pmatrix} y_j \mod q$$

and the entries of the left hand side vector are bounded by c in absolute value. Therefore for every $1 \le j \le n$,

$$\begin{pmatrix} a_{(j-1)k+1} \\ a_{(j-1)k+2} \\ \dots \\ a_{jk} \end{pmatrix} \equiv \begin{pmatrix} * \dots * t_{1j} \\ * \dots * t_{2j} \\ * \dots * \dots \\ * \dots * t_{kj} \end{pmatrix} \begin{pmatrix} y_1 \\ \dots \\ y_{j-1} \\ y_j \end{pmatrix} + \begin{pmatrix} z_{(j-1)k+1} \\ z_{(j-1)k+2} \\ \dots \\ z_{jk} \end{pmatrix} \mod q.$$

So $\bar{a} \equiv T\bar{y} + \bar{z} \mod q$, where $\bar{z} = (z_1, z_2, \dots, z_{kn})$ and $|z_i| \le c$. The statement is proved.

Theorem 2. *For every integer vector $a = (a_1, a_2, \dots, a_{kn+d})$ there exist an integer vector $y = (y_1, y_2, \dots, y_{n+d})$ and an integer vector $z = (z_1, z_2, \dots, z_{kn+d})$, where*

$$|z_i| \le c, \quad i = 1, \dots, kn, \tag{4}$$
$$z_i = 0, \quad i = kn+1, \dots, kn+d, \tag{5}$$

such that $a \equiv Ry + z \mod q$.

Proof. Let $\bar{a} = (a_1, a_2, \dots, a_{kn})$ and $a' = (a_{kn+1}, \dots, a_{kn+d})$. By Theorem 1, we have $\bar{a} \equiv T\bar{y} + \bar{z} \mod q$, where $\bar{y} = (y_1, y_2, \dots, y_n)$, and $\bar{z} = (z_1, z_2, \dots, z_{kn})$, and (4) holds.

Let $y' = (y_{kn+1}, \dots, y_{kn+d})$ satisfy the system of linear equations

$$a' \equiv R_2\, \bar{y} + R_3\, y' \mod q.$$

That implies $a \equiv Ry + z \mod q$, such that both (4) and (5) hold.

2.3 Public Key

The public key is an integer $kn \times (n+d)$ -matrix $A \equiv CRB^{-1} \mod q$.

2.4 Signature Generation

To sign the message M one computes $h = \text{HASH}(M) \in \mathbb{Z}_q^{kn}$. One takes secret $a_{kn+1}, \ldots, a_{kn+d}$ from $\mathbb{Z}_q$ and computes $a_1, \ldots, a_{kn}$ such that $Ca \equiv h \mod q$, where $a = (a_1, \ldots, a_{kn}, a_{kn+1} \ldots, a_{kn+d}) \in \mathbb{Z}_q^{kn+d}$. The vectors

$$y = (y_1, y_2, \ldots, y_{n+d}), \quad z = (z_1, z_2, \ldots, z_{kn+d}),$$

such that $a \equiv Ry + z \mod q$, and (4) and (5) hold, are computed according to Theorem 2. The signature for M is $x \equiv By \mod q$, so $x \in \mathbb{Z}_q^{n+d}$. Given h, the vector y is generally not unique. There may exist messages M which admit several valid signatures. We call $e = (z_1, z_2, \ldots, z_{kn})$ of size kn the error vector for M, x.

2.5 Signature Verification

To verify the signature x for M one computes $h = \text{HASH}(M)$ and Ax. One computes $e \equiv h - Ax \mod q$, where $e \in \mathbb{Z}_q^{kn}$ and such that the entries of $e = (e_1, e_2, \ldots, e_{kn})$ are at most $(q-1)/2$ in absolute value. The signature is accepted if $|e_i| \leq c$ for every $1 \leq i \leq kn$.

2.6 Verification Proof

We have

$$Ca \equiv h, \qquad a \equiv Ry + z \mod q,$$

where $z = (z_1, z_2, \ldots, z_{kn}, 0, \ldots, 0)$, $|z_j| \leq c$, and $x \equiv By$. Then

$$a \equiv Ry + z \equiv RB^{-1}x + z \mod q, \quad \text{so} \quad h \equiv Ax + Cz \mod q, \tag{6}$$

where the entries of $e = Cz = (z_1, z_2, \ldots, z_{kn})$ are bounded by c in absolute value. So the signature is accepted.

3 Signature distribution

In this section we prove that if $h = \text{HASH}(M)$ is distributed uniformly on $\mathbb{Z}_q^{kn}$, then the signature x is uniformly distributed on $\mathbb{Z}_q^{n+d}$. Recall that (2) has a solution for every $b_1, \ldots, b_k$. We can there put $t_1 = t_{1j} = 1, t_2 = t_{2j} \ldots, t_k = t_{kj}$ to simplify the notation below. So (2) is equivalent to the following statement. For every tuple of residues $b_1, \ldots, b_k$ modulo q there exist u and $i_1, \ldots, i_k$, where $|i_1| \leq c, \ldots, |i_k| \leq c$, and u is a residue modulo q, such that $b_1 \equiv u + i_1, b_2 \equiv ut_2 + i_2, \ldots, b_k \equiv ut_k + i_k$. Let $A(b_1, \ldots, b_k)$ denote the set of such u.

In order to prove that the signature $x = By$ is uniformly distributed it is enough to prove that y is uniformly distributed. According to Theorem 1 and Theorem 2, it is enough to prove that if $b_1, \dots, b_k$ are generated independently and uniformly at random on residues modulo q and the solution u to (2) is taken uniformly from $A(b_1, \dots, b_k)$, then u is uniformly distributed on residues modulo q. The probability of u is equal to

$$\frac{1}{q^k} \sum_{u \in A(b_1, \dots, b_k)} \frac{1}{|A(b_1, \dots, b_k)|},$$

where the sum runs over all $b_1, \dots, b_k$ such that $u \in A(b_1, \dots, b_k)$. The following lemma implies that this probability is $1/q$.

Lemma 1. $\sum_{u \in A(b_1, \dots, b_k)} \frac{1}{|A(b_1, \dots, b_k)|} = q^{k-1}$.

Proof. The inequalities (2) are equivalent to $b_1 - u \equiv i_1, b_2 - t_2 b_1 \equiv i_2 - t_2 i_1, \dots, b_k - t_k b_1 \equiv i_k - t_k i_1 \mod q$, where $|i_1| \le c, \dots, |i_k| \le c$. Let $s = s(a_2, \dots, a_k)$ be the number of solutions $i_1, \dots, i_k$ to

$$|i_1| \le c, \dots, |i_k| \le c, \qquad a_2 \equiv i_2 - t_2 i_1 \bmod q, \dots, a_k \equiv i_k - t_k i_1 \bmod q. \quad (7)$$

Then

$$|A(b_1, \dots, b_k)| = s(a_2, \dots, a_k),$$

where $a_2 \equiv b_2 - t_2 b_1, \dots, a_k \equiv b_k - t_k b_1$. Moreover, $u \in A(b_1, \dots, b_k)$ if and only if $b_1 = u + i_1$, where $i_1, \dots, i_k$ is a solution to (7). Since $a_2, \dots, a_k$ may take any values, we get

$$\sum_{u \in A(b_1, \dots, b_k)} \frac{1}{|A(b_1, \dots, b_k)|} = \sum_{a_2, \dots, a_k} \sum_{i_1, \dots, i_k} \frac{1}{s(a_2, \dots, a_k)} = \sum_{a_2, \dots, a_k} 1 = q^{k-1},$$

where the last sum is over all the solutions $i_1, \dots, i_k$ to (7).

4 Complexity

The linear system $Ca = h \mod q$ is easy to solve for any chosen $a_{kn+1}, \dots, a_{kn+d}$.

In signature generating the vector y may be computed in around $kn^2/2$ multiplications modulo q for small d. The complexity of computing $x \equiv By$ is n^2 multiplications. The signature size is $\lceil (n + d) \log_2 q \rceil$ bits. The complexity of verification is essentially kn^2 multiplications modulo q to compute Ax. Remark, that q may be taken relatively small compared with digital signature algorithms from the NIST competition. So the computation is very fast in that case.

For the public key one has to keep the matrix A, that is kn^2 residues modulo q. For the private key one keeps the matrix C and the matrices B, T. Instead, one may keep a seed and generate C, B and R with this seed if necessary. That can be done easily as all the entries of T except zeros and diagonal are random. The entries of B are random providing B is invertible modulo q which holds with high probability. So, the size of the private key may be made negligible.

5 Cryptanalysis

There are three approaches to the cryptanalysis: find private key given public key only, find private key by analysing a number of valid signatures, and forge signatures without the knowledge of the private key.

5.1 Private Key Recovery

We have not found any efficient method to recover the matrices C, R, B from $A \equiv CRB^{-1}$ besides searching over C or B according to their definitions which is not efficient.

5.2 Existential Forgery by Guessing

Given h, one may try small values ($\le c$ in absolute value) of some $n+d$ entries of $e \equiv h - Ax \bmod q$, compute x by solving a system of linear equations and check if all other $(k-1)n-d$ entries of e are at most c in absolute value. The success probability is $(\frac{2c+1}{q})^{(k-1)n-d}$. So, on the average, one needs to solve around $(\frac{q}{2c+1})^{(k-1)n-d}$ linear systems of n equations in n variables modulo q in order forge a signature for the hash value h.

5.3 Existential Forgery by Solving CVP

To forge the signature for a hash value h one is to find a vector e whose entries are bounded by c in absolute value and $h \equiv Ax + e$ for some vector x. This problem always has a solution for the parameters defining the signature algorithm. Let L be a lattice of rank kn and of volume q^{kn-n-d} generated by the columns of A modulo q. Thus it is enough to solve an approximate CVP-instance for L in the maximum norm.

The solution of this problem implies a vector in L at the Euclidean distance $\le c\sqrt{kn}$ from h. By Gaussian heuristic, see [18], the minimum distance between any h and L is $O(\sqrt{kn}\, q^{1-1/k})$ for average h and small d. Therefore, to forge signatures one has to solve a CVP-instance for L with a small approximation factor $O(\frac{c}{q^{1-1/k}})$. The approximate CVP is hard for general lattices of large rank if the approximation factor is small [14]. It is an open question how to use the structure of A to accelerate the solution.

One may also apply an exact CVP algorithm as in [1] or [7]. It is claimed in [7] that the CVP may be solved in heuristic time $2^{0.292\, r+o(r)}$ by a lattice sieving algorithm with the same amount of memory, where r is the rank of the lattice. That is not efficient for $r = kn$.

5.4 Key Recovery under Known Message Attack

Let r messages M_i, $i = 1, \ldots, r$ be signed with the same private key and x_i be theirs signatures respectively. Let $h_i = \mathrm{HASH}(M_i)$ and so $e_i \equiv h_i - Ax_i \mod q$

is an error vector for M_i. As $Ca_i \equiv h_i$, one may write

$$a_i = RB^{-1}x_i + e_i.$$

This equation does not seem to leak significant information about the matrix RB^{-1} as a_i, and therefore x_i, depend on d secret randomly chosen residues modulo q and d secret right most columns of C.

5.5 Adaptive forgery

A message M with a hash value h may have several valid signatures $x_1, x_2, \ldots$ So, the equations

$$h \equiv Ax_1 + e_1, h \equiv Ax_2 + e_2, \ldots$$

are available. Let a signature x_0 for another message M_0 with a hash value h_0 be available and so $h_0 \equiv Ax_0 + e_0$. One may try to modify x_0 to yet another valid signature $x_0 + x_1 - x_2$ for M_0 and get

$$h_0 \equiv A(x_0 + x_1 - x_2) + e_0 + e_1 - e_2.$$

However, the probability to accept $x_0 + x_1 - x_2$, that is the probability that every entry of $e_0 + e_1 - e_2$ is bounded by c in absolute value, is very low. For instance, if $c = 2$ and $q \geq 9$ this probability is

$$(1 - 40/125)^{kn}.$$

So this method is not efficient.

5.6 Matrix A

In this section we show, with an heuristic argument, that if d is large enough, then the public key matrix $A \equiv CRB^{-1}$ is uniformly distributed, where the matrices C, R, B are chosen according to the definitions in Section 2.2.

Let R, B be defined by Section 2.2. We neglect that the matrices R_3, B have to be invertible modulo q, which holds with high probability. Then the matrix R depends on $kn(n-1)/2 + d(n+d)$ free variables, and the matrix B depends on $(n+d)^2$ free variables, residues modulo q.

It is easy to prove that the values of $n(n-1)/2 + d(n+d)$ variables in R determine pairs of matrices R_1, B_1, also as in Section 2.2, such that $RB^{-1} = R_1 B_1^{-1}$ and the diagonal entries in R_1 are as in R.

The matrix C depends on knd free variables. We take into account the equivalence above and get that the matrix $A \equiv CRB^{-1}$ depends on

$$\begin{aligned} knd + kn(n-1)/2 + d(n+d) + (n+d)^2 - n(n-1)/2 - d(n+d) \\ = knd + (k-1)n(n-1)/2 + (n+d)^2 \end{aligned}$$

free variables. Since the matrix A is of size $kn \times (n+d)$, the inequality

$$knd + (k-1)n(n-1)/2 + (n+d)^2 \geq kn(n+d)$$

would heuristically imply that A is thus generated uniformly. This inequality holds for $d = n(\sqrt{(k+1)/2} - 1) + o(n)$ when n is growing. In particular, for $k = 2$ we have $d = (\sqrt{3/2} - 1)n + o(n)$. Similarly, for $k = 3$ we have $d = (\sqrt{2} - 1)n + o(n)$.

6 Proposed parameters

In this section we propose parameter sets. They are chosen to approximately security level 2^{120} bit operations to break the system. The parameters are optimised to balance the complexity of so far best attacks. They are a guessing algorithm in Section 5.2 and a lattice sieving algorithm to solve a relevant instance of CVP in Section 5.3. Also, the parameters are chosen to minimise the size of the signature and the size of the public key: $k = 3, c = 2$ and $q = 9$. We take the parameter $d = (\sqrt{2} - 1)n + o(n)$ thus providing the public matrix A of size $kn \times (n+d)$ depends on more than $kn(n+d)$ free variables, residues modulo q. The entries of $A \equiv CRB^{-1}$ are complicated cubic polynomials in the variables of C, R, B^{-1}. This, at least heuristically, implies that A is taken uniformly.

For $q = 9, k = 3, c = 2$ there is only one tuple $[1, 2, 4]$, up to the equivalence, which satisfy the condition in Section 2.2. For $q = 7, k = 3, c = 2$ there are several such tuples, for instance $[1, 1, 1]$.

6.1 Security level 2^{120}

We set $(n, k, d, q, c) = (135, 3, 57, 9, 2)$. The signature size is 609 bits (77 bytes) and the public key size is 30.81 Kbytes. In order to forge a signature x given a hash value h one may apply the attack in Section 5.2, where the probability to find x such that every entry of $e \equiv h - Ax \bmod q$ is bounded by c in absolute value is $\left(\frac{2c+1}{q}\right)^{kn-n-d} \approx 2^{-180.6}$. Therefore, one has to solve $2^{180.6}$ linear systems modulo $q = 9$ to forge the signature with this method on the average. The algorithm in [1] solves an instance of CVP in Section 5.3 and therefore forges signature in $2^{118.3+o(118.3)}$ operations with memory size of the same magnitude. This choice fits security level 2^{120}.

Alternatively, a similar security level is provided by taking $(n, k, d, q, c) = (140, 3, 59, 7, 2)$. The signature size is 70 bytes and the public key size is 33.7 Kbytes.

6.2 New Algorithm versus NIST Selected Digital Signature Algorithms

We summarise the security and some complexity parameters of the new algorithm in the first line of Table 1 and put them against those of the NIST Selected Digital Signature Algorithms with approximately matching security 2^{120}, see [15]. In Table 1 bits, bytes and kilobytes are abbreviated by b, B and kB respectively. The signatures generated with the new algorithm are several times shorter than those of the NIST algorithms though the public key size is significantly larger. A similar holds for higher security levels.

References

1. A. Becker, N. Gama, A. Joux, *Solving shortest and closest vector problems: The decomposition approach*, IACR Cryptology ePrint Archive, 2013/685.

Table 1. Comparison with NIST 3-rd round candidates

algorithm	security	public key	arithm. q	sign.
$(135, 3, 57, 9, 2)$	118 b	30.81 kB	9	77 B
Dilithium level 2	121 b	1.31 kB	8380417	2420 B
Falcon level 2	120 b	0.897 kB	12289	666 B
SPHINCS$^+$ level 1	133 b	0.032 kB	-	7856 B

2. A. Budroni, I. Semaev, *New Public-Key Crypto-System EHT*, IACR Cryptology ePrint Archive, 2021/234.
3. G. Hanrot, X. Pujol, D. Stehlé, *Analyzing blockwise lattice algorithms using dynamical systems,* in CRYPTO 2011, LNCS, vol. 7073, pp. 1–20, Springer 2011.
4. J. Hoffstein, N. Howgrave-Graham, J. Pipher, J.H. Silverman, W. Whyte *NTRUSign: Digital signatures using the NTRU lattice* in CT-RSA 2003. LNCS, vol. 2612, pp. 122–140, Springer 2003.
5. C. Gentry and M. Szydlo, *Cryptanalysis of the Revised NTRU Signature Scheme* in EUROCRYPT 2002, LNCS, vol. 2332, pp. 299–320, Springer 2002.
6. O. Goldreich, S. Goldwasser, S. Halevi, *Public-key cryptosystems from lattice reductions problems*, in CRYPTO 1997. LNCS, vol. 1294, pp. 112–131, Springer 1997.
7. T. Laarrhoven, *Sieving for closest lattice vectors(with preprocessing)*, arXiv: 1607.04789v1, 16 Jul 2016.
8. X. Nie, X. Jiang, L. Hu, and J. Ding, *Cryptanalysis of Two New Instances of TTM Cryptosystem*, Cryptology ePring Archive.
9. P. Q. Nguyen, O. Regev, *Learning a Parallelepiped: Cryptanalysis of GGH and NTRU Signatures,* Journal of Cryptology, vol. 22(2), pp.139–160.
10. L. Goubin, N. Courtois, *Cryptanalysis of the TTM Cryptosystem*, in ASIACRYPT 2000, LNCS, vol 1976, pp. 44–57, Springer 2000.
11. C. Gentry, C. Peikert, V. Vaikuntanathan, *Trapdoors for hard lattices and new cryptographic constructions*, in STOC 2008, pp. 197–206, ACM 2008.
12. J.C. Faugére, A. Joux, *Algebraic Cryptanalysis of Hidden Field Equation (HFE) Cryptosystems Using Gröbner Bases*, in CRYPTO 2003, LNCS, vol 2729, pp. 44–60, Springer, 2003.
13. T. Laarhoven, *Sieving for shortest vectors in lattices using angular locality-sensitive hashing,* in CRYPTO 2015, LNCS vol. 9215, pp. 3–22, Springer 2015.
14. D. Micciancio, and S. Goldwasser , *Complexity of Lattice Problems: A Cryptographic Perspective*, The Kluwer International Series in Engineering and Computer Science, vol. 671, Kluwer Academic Publishers, Boston, MA, 2002.
15. https://csrc.nist.gov/projects/post-quantum-cryptography/post-quantum-cryptography-standardization.
16. D. Micciano, C. Peikert, *Trapdoors for Lattices: Simpler, Tighter, Faster, Smaller*, in EUROCRYPT 2012, LNCS, vol 7237, pp. 700-718, Springer 2012.
17. Ph. Q. Nguyen and O. Regev, *Learning a Parallelepiped: Cryptanalysis of GGH and NTRU Signatures*, J, Cryptol. vol. 22 (2009), pp. 139–160.
18. Ph. Q. Nguyen, B. Vallée, editors, *The LLL Algorithm, Survey and Applications,* Springer,2010.
19. A. Pellet-Mary, D. Stehle, *On the hardness of the NTRU problem,* in ASIACRYPT'21, LNCS 13090, pp.3–35, 2021.
20. T. Plantard, W. Susilo, and K. T. Win, *A Digital Signature Scheme Based on CVP_∞*, in PKC 2008, LNCS 4939, pp. 288–307, 2008.

21. D. Wiedemann, *Solving sparse linear equations over finite fields*, IEEE Trans. on Inf. Theory, vol. 32 (1986), pp. 54–62.