New Digital Signature Algorithm

No Author Given

No Institute Given

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 shorter than that provided by the NIST Selected Digital Signature Algorithms with a comparable security level, while the public key size is larger.

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 [15, 4], yet another NTRU-based signature algorithm variation Falcon is among the finalists of the NIST Post Quantum Standardisation Process, [13]. Similarly, several variations of multivariate algorithms as HFE and TTM were broken [10, 8], 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 algorithm (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 [5], some its modifications as DRS, see [18], and NTRU-based signature algorithms as NTRUSign in [3]. Another approach based on the hardness of the SIS (Short Integer Solution) problem was implemented in [9],[14]. 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 [17]. Similarly, a reduction-based security argument for the underlying problem of the new digital algorithm, the problem is specified in Section 9 of this work, may require more time and effort.

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 be positive integers. The sig-

nature for a message M is $x \in \mathbb{Z}_q^n$ such that $\mathrm{HASH}(M) = Ax + e$ for some public matrix $A \in \mathbb{Z}_q^{kn \times n}$ and vector $e \in \mathbb{Z}_q^{kn}$, where HASH denotes a public hash function. 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 $x \in \mathbb{Z}_q^n$ 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.

Assume that the message M was already signed with x. To forge another signature for M one has to solve a similar problem. Namely, one has to construct $x_1 \in \mathbb{Z}_q^n, e_1 \in \mathbb{Z}_q^{kn}$, where $x \neq x_1$, and the entries of e_1 are bounded, and such that $\mathrm{HASH}(M) = Ax_1 + e_1$. That is equivalent to finding a nonzero $y = x_1 - x \in \mathbb{Z}_q^n$ and an entry bounded vector $e_1 \in \mathbb{Z}_q^{kn}$ such that $e = Ay + e_1$. We have not found an efficient method to solve this problem too.

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.

In Section 5.4 we analyse the security of the signature algorithm if a number of valid signatures is available.

A reduced version is in Section 6 below and some explicit parameters are proposed in Section 7. The underlying hard problem for these parameters is described in Section 8.

2.1 Parameters

Let n, k, λ, c be positive integers, and q be an odd prime, $\lambda \geq 3$, and $2\lambda c + 1 < q$. Also, $h = \mathrm{HASH}(M)$ is a hash value of the message M, where h is encoded by a vector in $\mathbb{Z}_q^{kn}$.

2.2 Private Key

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

1. The matrix T is an integer $kn \times n$ matrix in a column echelon form

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

where entries $t_{1j}, t_{2j}, \dots, t_{kj}$ are called diagonal and they are not supposed to be secret. The entries of T below the diagonal are denoted by $*$, they may be randomly generated and are generally secret. In a variation introduced in Section 6 the entries below the diagonal are set to 0 for r right most columns of T, where r is a parameter. For the proposed parameters in Section 7 one sets $r = n/2$. That does not seem to affect the security, see Sections 8 and 10, and reduces the size of the public key as shown in Section 6.
Each tuple $[t_{1j}, t_{2j}, \dots, t_{kj}]$ has to satisfy the following properties. First, all entries are non-zero residues modulo q and at least one is coprime to q. Second, for any integer $b_1, b_2, \dots, b_k$ there is an integer u such that

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

For small 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.
2. The matrix $C = (C_{ij})$ is an integer $kn \times kn$-matrix, the 1-norm of the rows C_i of which satisfy $||C_i||_1 = \sum_{j=1}^{kn} |C_{ij}| \le \lambda$. To define C one may take $C = P_1 + P_2 + \dots + P_\lambda \mod q$, where P_i are permutation matrices of size $kn \times kn$. Experimentally, such C is invertible over rationals with high probability if $\lambda > 2$. We assume that C is invertible modulo q. For $\lambda = 2$ one can achieve $C = P_1 + P_2$ of full rank by Lemma 2, see Appendix (Section 11). However, we do not recommend $\lambda = 2$ due to a weakness found in Section 5.

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

Theorem 1. *For every integer vector $a = (a_1, a_2, \ldots, a_{kn})$ there exist an integer vector $y = (y_1, y_2, \ldots, y_n)$ and an integer vector $z = (z_1, z_2, \ldots, z_{kn})$, where $|z_i| \le c$ for every $i = 1, \ldots, kn$, such that $a \equiv Ty + z \mod q$.*

Proof. We show how to compute iteratively the entries 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

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

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 $jk+1, jk+2, \ldots, jk+k$ and columns $1, \ldots, j$, 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

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

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

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

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

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

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

2.3 Public Key

The public key is an integer $kn \times n$ matrix $A \equiv CTB \mod q$.

2.4 Signature Generation

To sign the message M one computes $h = \mathrm{HASH}(M)$. Let $a = (a_1, a_2, \ldots, a_{kn})$ such that $a \equiv C^{-1}h \mod q$. The vectors

$$y = (y_1, y_2, \ldots, y_n), \quad z = (z_1, z_2, \ldots, z_{kn}),$$

such that $a \equiv Ty + z \mod q$ and $|z_i| \leq c$ are then computed according to Theorem 1, where each y_j is taken uniformly from the set of solutions to (2) for appropriate $b_1, \ldots, b_k$. The signature is $x \equiv B^{-1}y \mod q$, where $x \in \mathbb{Z}_q^n$. We call $e = Cz$ the error vector for M, x. Given h, the vector y is generally not unique. There may exist messages M which admit several valid signatures.

2.5 Signature Verification

To verify the signature x for M one computes $h = \mathrm{HASH}(M)$ and Ax. Let $e \equiv h - Ax \mod q$, 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 $|e_i| \leq \lambda c$ for every $1 \leq i \leq kn$.

2.6 Verification Proof

According to the signature generating algorithm $C^{-1}h \equiv Ty + z \mod q$, where $z = (z_1, z_2, \ldots, z_{kn})$ and $|z_j| \leq c$ and $x \equiv B^{-1}y$. Then

$$C^{-1}h \equiv Ty + z \equiv TBx + z \mod q, \quad \text{and} \quad h \equiv Ax + Cz \mod q, \quad (3)$$

where the entries of $e = Cz$ are bounded by λc in absolute value. So the signature is accepted.

3 Signature distribution

In this section we prove that if $h = \mathrm{HASH}(M)$ is distributed uniformly on $\mathbb{Z}_q^{kn}$, then the signature x is uniformly distributed on $\mathbb{Z}_q^n$. 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 $x = B^{-1}y$ is uniformly distributed it is enough to prove that y is uniformly distributed. According to Theorem 1, it is enough to prove that if $b_1, \ldots, b_k$ are generated independently and uniformly

at random on residues modulo q and the solution u to (2) is taken uniformly from $A(b_1, \ldots, b_k)$, then u is uniformly distributed on residues modulo q. The probability of u is equal to

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

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

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

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

$$|i_1| \le c, \ldots, |i_k| \le c \qquad a_2 \equiv i_2 - t_2 i_1 \bmod q, \ldots, a_k \equiv i_k - t_k i_1 \bmod q. \tag{4}$$

Then

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

where $a_2 \equiv b_2 - t_2 b_1, \ldots, a_k \equiv b_k - t_k b_1$. Moreover, $u \in A(b_1, \ldots, b_k)$ if and only if $b_1 = u + i_1$, where $i_1, \ldots, i_k$ is a solution to (4). Since $a_2, \ldots, a_k$ may take any values, we get

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

where the last sum is over all the solutions $i_1, \ldots, i_k$ to (4).

4 Complexity

One may solve the linear system $Ca = h \mod q$ for a with Wiedemann's algorithm [19] in at most $\lambda(kn)^2$ additions and $(kn)^2$ multiplications modulo q. Another option is to keep the precomputed matrix C^{-1} modulo q and compute $a \equiv C^{-1} h \mod q$.

In signature generating the vector y may be computed in around $2ckn + kn^2/2$ multiplications modulo q and the complexity of computing $x \equiv B^{-1} y$ is n^2 multiplications. The signature size is $\lceil n \log_2 q \rceil$ 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, see Table 1. 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^{-1} (or a minimal polynomial for C to apply the Wiedemann algorithm) and the matrices B^{-1}, T. Instead, one may keep a seed and generate B^{-1} and T with this seed if necessary. That can

be done easily as all the entries of T except zeros and diagonal are random and the entries of B are random, where B is invertible with probability close to 1. So the size of the private key may be made negligible.

The complexity parameters are significantly lower for the reduced version of the signature algorithm in Section 6.

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, T, B from $A = CTB$ besides searching over C or B according to their definitions. However, if λ and k are small, one may recover around $n/\lambda - n/k\lambda$ rightmost columns of CT and B^{-1} relatively fast.

Really, let b be the rightmost column of the matrix B^{-1}. Then Ab is the rightmost column of CT. As T is in a column echelon form, the rightmost column of CT has at most $k\lambda$ nonzero entries. Let $A(m)$ be a sub-matrix of A in $m > n$ randomly chosen rows. With probability at least $\binom{kn-k\lambda}{m}/\binom{kn}{m}$ we have $A(m)b \equiv 0 \mod q$ and b is recovered by solving a system of linear equations. One thus recovers the rightmost column of CT with the number of trials at most $\binom{kn}{m}/\binom{kn-k\lambda}{m}$. One now eliminates $k\lambda$ rows from A, where the rightmost column of CT has non-zero entries. Then the second rightmost column of CT is similarly recovered, etc. The l-th right most column of CT is found after $\binom{kn-(l-1)k\lambda}{m}/\binom{kn-lk\lambda}{m}$ trials for $l \le n/\lambda - n/k\lambda$. For $l > n/\lambda - n/k\lambda$ the complexity of recovering the columns of CT grows very fast because the system of linear equations $A(m)b = 0$ will have rank $m = kn - lk\lambda < n$ and the number of solutions is of order $q^{n-kn+lk\lambda}$ before the system gets trivial, that is with q^n solutions, which happens when $l = n/\lambda$. Therefore, approximately $n/\lambda - n/k\lambda < n/\lambda$ right most columns of CT and of B^{-1} may be recovered. To forge signatures one has to know the whole matrices C, T, B. According to the definitions in Section 7 the left most $s = n - n/\lambda$ columns of T still contain $nks - ks(s+1)/2 \approx kn^2(1/\lambda - 1/\lambda^2)$ unknown non-diagonal entries and the left most s columns of B still contain $ns = n^2(1 - 1/\lambda)$ unknown entries, residues modulo q. Therefore, the method is not efficient enough to forge signatures.

5.2 Existential Forgery by Guessing

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

around $(\frac{q}{2\lambda c+1})^{(k-1)n}$ linear systems of n equations in n variables modulo q in order forge a signature for 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 λc 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} 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 $\leq \lambda c\sqrt{kn}$ from h. By Gaussian heuristic, see [16], the minimum distance between any h and L is $O(\sqrt{kn}\, q^{1-1/k})$ for average h. Therefore, to forge signatures one has to solve a CVP-instance for L with a small approximation factor $O(\frac{\lambda c}{q^{1-1/k}})$. The approximate CVP is hard for general lattices of large rank if the approximation factor is small [12]. 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 [6]. It is claimed in [6] that the CVP may be solved in heuristic time $2^{0.292\, d+o(d)}$ by a lattice sieving algorithm with the same amount of memory, where d is the rank of the lattice. That is not efficient for $d = kn$.

5.4 Key Recovery under Known Message Attack

Let $m \geq kn$ messages M_i, $i = 1, \ldots, m$ be signed with the same private key and s_i be their signatures respectively. Then

$$h_i - As_i \equiv Cf_i \mod q,\ i = 1, \ldots, m, \tag{5}$$

where Cf_i is the error vector for M_i, s_i and $h_i = \mathrm{HASH}(M_i)$. The entries of f_i are bounded by c. In these equations h_i, A, s_i are public while f_i, C are secret. Let

$$U = [h_1 - As_1, \ldots, h_m - As_m] \equiv [Cf_1, \ldots, Cf_m] \mod q$$

be a matrix of size $kn \times m$ whose columns are left hand side columns in (5). Let b be a row in $C^{-1} \mod q$ and $v \equiv bU \mod q$. The entries of v are at most c in absolute value as they are some entries of f_i. The vector v belongs to the lattice L generated by the rows of U modulo q. The lattice L is of rank m and of volume $\mathrm{Vol} = q^{m-kn}$. The Euclidean norm of v is at most $c\sqrt{m}$ and L contains at least nk such vectors. Thus BKZ reduction or a sieving algorithm may be tried to recover such v and the rows b of $C^{-1} \mod q$. Recovering the matrices T, B is then easy.

The application is successful if $c\sqrt{m}$ is around the first minimum of L. Otherwise, the lattice may contain too many vectors whose norm is close to the norm of the target vectors v. The first minimum is bounded by $\sqrt{\gamma_m}\, \mathrm{Vol}^{1/m}$,

where γ_m is the Hermit constant for rank-m lattices. For large m we have $\gamma_m \le (1.745/2\pi e)m$, see [16]. We take the smallest m such that

$$|v| \le c\sqrt{m} \le \sqrt{(1.745/2\pi e)m}\; q^{(m-kn)/m}.$$

So $m \ge \left\lceil \ln q / \ln\left(\frac{q}{c}\sqrt{\frac{1.745}{2\pi e}}\right)\right\rceil kn$. The complexity of sieving is $2^{0.292\,m+o(m)}$ operations according to [11] with the memory of the same order. As m is very large, this method is inefficient. Block BKZ reduction does not provide any advantage over the plain sieving algorithm in this setting. For instance, for the parameters in Section 7 the size of the optimal block computed according to [2] is again m.

Let $\lambda = 2$ and let $C = P_1 + P_2$ be a sum of two permutation matrices P_1, P_2 such that the permutation $Q = P_1 P_2^{-1}$ has two cycles of odd length. Then $C^{-1} = (1/2)\,K$ over rationals, where the entries of K are $0, \pm 1$ by Lemma 2 in the Appendix. So $K(h_i - As_i) = 2f_i \mod q$ and one may recover K and therefore C faster than in the general case. That is why $\lambda = 2$ is not recommended. For $\lambda > 2$ the matrix C^{-1} does not have such representation. Therefore the method does not work.

6 Reduced Public Key

According to the cryptanalysis in Section 5.1, around $n/\lambda - n/k\lambda$ rightmost columns of CT and of B^{-1} may be recovered relatively fast given A. Therefore these columns may not be considered secret. One can use that to reduce the size of the public key. Let $1 \le s \le n$ be an integer parameter and $r = n - s$. We set

$$C = \begin{pmatrix} C_1 & C_4 \\ C_2 & C_3 \end{pmatrix}, \quad T = \begin{pmatrix} T_1 & 0 \\ T_2 & T_3 \end{pmatrix}, \quad B = \begin{pmatrix} B_1 & 0 \\ B_2 & I \end{pmatrix}.$$

The sub-matrices C_1, C_3 are square and of size $ks \times ks$ and $kr \times kr$ respectively. That defines the size of the sub-matrices C_2, C_4. The sub-matrices T_1, T_3 are of size $ks \times s$ and $kr \times r$ respectively, where T_3 is defined by (8) below. That is the matrix T_3 is of type (1), where only diagonal entries are non-zero. The sub-matrix B_1 is square and of size $s \times s$, invertible modulo q, and I is an identity matrix of size $r \times r$. We assume that C_3, C_4, T_3 are public while $C_1, C_2, T_1, T_2, B_1, B_2$ are secret. Then

$$A \equiv CTB \equiv \begin{pmatrix} C_1T_1B_1 + C_4(T_2B_1 + T_3B_2) & C_4T_3 \\ C_2T_1B_1 + C_3(T_2B_1 + T_3B_2) & C_3T_3 \end{pmatrix} \equiv \begin{pmatrix} A_1 & A_4 \\ A_2 & A_3 \end{pmatrix}.$$

Suppose C_3 is invertible modulo q. Then

$$A \equiv \begin{pmatrix} A' + C_4C_3^{-1}A_2 & A_4 \\ A_2 & A_3 \end{pmatrix} \mod q, \tag{6}$$

where $A' = C'T_1B_1$ and $C' \equiv C_1 - C_4C_3^{-1}C_2$. Let, for instance, $\lambda = 4$. One chooses the rows of C_1, C_2 to be of 1-norm equal to 1 and the rows of C_3, C_4 to

be of 1-norm equal to 3. Experimentally, with high probability, C_3 is invertible modulo q and C_3^{-1} is quite dense. So the secret matrix C' is not sparse in that case and it hides the structure of $T_1 B_1$ in the definition of A'. As A is public, one may recover the matrix A'. However, one can not recover any columns of $C'T_1$ and B_1^{-1} from A' as in Section 5.1.

As the matrices A_3, A_4 are sparse and may be made constants, one essentially stores only the entries of A_1, A_2 for the public key. That makes kns residues modulo q.

7 Proposed parameters

In this section we propose parameter sets. They are chosen to approximately fit two security levels 2^{120} and 2^{240} bit operations to break the system. The parameters are optimised to balance the complexity of the three so far best attacks. They are a guessing algorithm in Section 5.2, another guessing algorithm in Section 10 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. In order to reduce the public key the method in Section 6 was applied with $r = s = n/2$ and $k = 2$. In every parameter set below we take $\lambda = 4$ and $c = 2$.

Parameter sets are defined for prime $q = 23$ and composite $q = 24, 25$. There is only one tuple $[t_1, t_2] = [1, 5]$ modulo both $q = 23$ and 24 up to a permutation, multiplication of the tuple modulo q by residues coprime to q and changing sign of the tuple entries such that the system of inequalities (2) for $k = 2, c = 2$ has a solution u for every integer b_1, b_2. For $q = 25$ there are two such tuples $[1, 5]$ and $[1, 10]$. One takes $[t_1, t_2]$ independently to define the diagonal entries of T.

7.1 Security level 2^{120}

We set $(n, k, q, \lambda, c) = (230, 2, 23, 4, 2)$. The signature size is 1040 bits (130 bytes) and the public key size is 30.3 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 λc in absolute value is $\left(\frac{2\lambda c+1}{q}\right)^{(k-1)n} \approx 2^{-100.3}$. Therefore, one has to solve $2^{100.3}$ linear systems with $n = 230$ equations and $n = 230$ variables modulo $q = 23$ to forge the signature with this method on the average. This is the best attack so far in this setting. The complexity matches the complexity of the guessing algorithms in Section 10. The algorithm in [1] solves an instance of CVP in Section 5.3 and therefore finds e in $2^{134.3}$ operations with memory size $2^{134.3}$, these figures may in fact be larger due to hidden factors. This choice fits the first security level 2^{120}. The new algorithm with this parameters was implemented on a common computer. The signature verification took less than 10^{-3} of a second per signature and the signature generation was around 4 times longer.

Using $q = 24$ significantly reduces the size of the signature and the public key. For $(n, k, q, \lambda, c) = (200, 2, 24, 4, 2)$ the signature size is 917 bits (115 bytes)

and the public key size is 23.3 Kbytes. These parameters fit the first security level 2^{120} by the argument above as $\left(\frac{2\lambda c+1}{q}\right)^{(k-1)n} \approx 2^{-99.5}$, see Section 5.2, and the algorithm in [1] solves an instance of CVP in Section 5.3 and therefore finds e and forges x in 2^{117} operations with memory size 2^{117}, these figures may be larger due to hidden factors.

Using $q = 25$ for this security level does not significantly reduce the size of the signature and the public key.

7.2 Security level 2^{240}

For $(n, k, q, \lambda, c) = (500, 2, 23, 4, 2)$ the signature size is 2262 bits (283 bytes) and the public key size is 142.2 Kbytes. These parameters fit the second security level 2^{240} as $\left(\frac{2\lambda c+1}{q}\right)^{(k-1)n} \approx 2^{-218.0}$, see Section 5.2, and the algorithm in [1] solves an instance of CVP in Section 5.3 and therefore finds e and forges x in $2^{292.5}$ operations with memory size $2^{292.5}$.

For $(n, k, q, \lambda, c) = (430, 2, 24, 4, 2)$ the signature size is 1972 bits (247 bytes) and the public key size is 106.7 Kbytes. These parameters fit the second security level 2^{240} as $\left(\frac{2\lambda c+1}{q}\right)^{(k-1)n} \approx 2^{-213.9}$, see Section 5.2, and the algorithm in [1] solves an instance of CVP in Section 5.3 and therefore finds e and forges x in $2^{251.5}$ operations with memory size $2^{251.5}$.

For $(n, k, q, \lambda, c) = (400, 2, 25, 4, 2)$ the signature size is 1858 bits (232 bytes) and the public key size is 93.5 Kbytes. These parameters fit the second security level 2^{240} as $\left(\frac{2\lambda c+1}{q}\right)^{(k-1)n} \approx 2^{-222.5}$ and the algorithm in [1] solves an instance of CVP in Section 5.3 and therefore finds e and forges x in $2^{234.0}$ operations with memory size $2^{234.0}$, these figures may be larger due to hidden factors. It is easy to see that for $q = 25$ and $c = 2$ the signature algorithm generates a unique signature for any given hash value h.

7.3 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 [13]. In Table 1 bits, bytes and kilobytes are abbreviated by b, B and kB respectively. One sees that 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.

8 Underlying Problem

Let $n, k, s, r = n - s$ be positive integers, q be an odd prime, and let $\delta < q$ be a positive real. Let R denote an integer matrix of size $kn \times s$, whose entries modulo

Table 1. Comparison with NIST 3-rd round candidates

algorithm	security	public key	arithm. q	sign.
$(200, 2, 24, 4, 2)$	119 b	23.2 kB	24	115 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

q were generated uniformly at random. Also, let K be an integer matrix of size $kn \times r$ and of rank r modulo q. Given an integer vector h of size kn, one asks to find integer vectors x_1, x_2 of size s and r respectively and an integer vector e of size kn such that every entry of e is at most δ in absolute value and

$$Rx_1 + Kx_2 + e \equiv h \bmod q. \tag{7}$$

Let's denote the concatenations $A = R|K$ and $x = x_1|x_2$. Then (7) is equivalent to $Ax + e \equiv h \bmod q$. That is an instance of the CVP in the maximum norm for the lattice generated by the columns of A modulo q. Heuristically, the problem has a solution for every h if $q^n(2\delta + 1)^{kn} > q^{kn}$.

The security of the new signature algorithm is based on the hardness of solving (7) for some matrices R and K. For the parameters $k = 2, s = r = n/2, \lambda = 4$ and matrices specified in Sections 6, the matrix R may be considered as generated uniformly according to Section 9 below. To construct the matrix K, let

$$T_3 = \begin{pmatrix} t_{11} & 0 & \dots & 0 \\ t_{21} & 0 & \dots & 0 \\ \dots & & & \\ t_{k1} & 0 & \dots & 0 \\ 0 & t_{12} & \dots & 0 \\ 0 & t_{22} & \dots & 0 \\ \dots & & & \\ 0 & t_{k2} & \dots & 0 \\ \dots & & & \\ 0 & 0 & \dots & t_{1r} \\ 0 & 0 & \dots & t_{2r} \\ \dots & & & \\ 0 & 0 & \dots & t_{kr} \end{pmatrix} \tag{8}$$

be a matrix of size $kr \times r$ for non-zero diagonal entries t_{ij} such that each tuple $[t_{1j}, t_{2j}, \dots, t_{kj}]$ has to satisfy (2) in Section 2.2.

Also, let C_3 and C_4 be matrices of size $kr \times kr$ and $ks \times kr$ respectively and whose rows have 1-norm (the sum of the absolute values of the entries) equal to 3. The matrices C_3, C_4, T_3 are public. Then

$$K = \begin{pmatrix} C_4 T_3 \\ C_3 T_3 \end{pmatrix} \tag{9}$$

is a matrix of size $kn \times r$ and of rank r. See Section 9 below for details.

8.1 Sub-problem.

Let a matrix K of size $kn \times s$ be defined by (9). Given an integer vector h of length kn find an integer vector z of length s and an integer vector e of length kn such that $Kz + e \equiv h \bmod q$ and every entry of e is bounded by δ in absolute value. Heuristically, the problem has a solution if $q^s(2\delta+1)^{kn} > q^{kn}$. An efficient algorithm to solve this equation implies an efficient algorithm to solve the equation (7).

9 Reduction to the Underlying Problem

When constructing the public matrix A in Section 6, the matrix T_2 of size $kr \times s$ may be taken uniformly at random. So the matrix A_2 of size $kr \times s$ in (6) is uniformly distributed. The matrix $A' = C'T_1B_1$ of size $ks \times s$ is generated independently of A_2. It depends on $\frac{ks(s-1)}{2}$ randomly chosen entries of T_1 and on s^2 randomly chosen entries of the invertible B_1, besides randomly chosen C_1, C_2. So the matrix $R = \begin{pmatrix} A_1 \\ A_2 \end{pmatrix}$ of size $kn \times s$ depends on $\frac{ks(s-1)}{2} + krs + s^2$ independent residues modulo q and on C_1, C_2.

Let $k = 2$ and $s = r = n/2$, and $\lambda = 4$. We set the public C_3, C_4 to be a sum of three permutation matrices of size $s \times s$ each. The secret C_1, C_2 are permutation matrices of size $s \times s$.

Then the matrix R of size $2n \times n/2$, that is with n^2 entries depends on $n^2 - n/2$ randomly and independently chosen residues modulo q and besides on randomly chosen C_1, C_2. As $q^{n^2-n/2}(\frac{n}{2}!)^2 > q^{n^2}$ for relatively small q, the number of independent parameters for the entries of R is larger than the number of their entries. So, heuristically, the matrix R is uniformly distributed. By construction, the matrix $K = \begin{pmatrix} A_4 \\ A_3 \end{pmatrix} = \begin{pmatrix} C_4T_3 \\ C_3T_3 \end{pmatrix}$ is of full rank $r = n/2$.

The signature x for a hash value h satisfies $Ax + e = h$, where the entries of e are at most λc in absolute value. Let $x = x_1|x_2$, where x_1, x_2 are of size s. Then $Ax + e = h$ implies $Rx_1 + Kx_2 + e = h$. To forge a signature one must solve an instance of the problem in Section 8 with parameters $k = 2, r = s = n/2$ and $\delta = \lambda c = 4c$.

The size of the public key is essentially n^2 residues modulo q. Verification cost is essentially n^2 multiplications modulo q.

10 Solving the Underlying Problem by Guessing

There are two guessing type algorithms to find a solution to (7). First, one may guess n small (bounded by δ) entries of e, find $x = x_1|x_2$ by solving a system of linear equations modulo q, then check other $kn - n$ entries of e. If they all are small (bounded by δ), then the solution is found. The probability of success is $(\frac{2\delta+1}{q})^{kn-n}$.

Second, let x_1 be a random vector of our choice and $h - Rx_1 = h_1|h_2$, where the size of h_1 is ks and the size of h_2 is kr. Also, let $e = e_1|e_2$, where the size of e_1 is ks and the size of e_2 is kr. Assume that C_3 is invertible modulo q. By using Theorem 1, one finds vectors x_2 and f such that the entries of f are small (bounded by $\lfloor \delta/3 \rfloor$) and

$$T_3 x_2 + f = C_3^{-1} h_2.$$

Then $C_3 T_3 x_2 + e_2 = h_2$, where the entries of $e_2 = C_3 f$ are small (bounded by δ) as the 1-norm of the rows of C_3 is equal to 3. The probability that the vector $e_1 = h_1 - C_4 T_3 x_2$ has small entries is $(\frac{2\delta+1}{q})^{ks}$. So (7) is satisfied for such $x = x_1|x_2$ and e. The success probabilities of the both algorithms are equal for $ks = kn - n$. Therefore, one may set $s = n - n/k$. We conjecture that the equation (7) is hard for K defined by (9), such s and for large n.

References

1. A. Becker, N. Gama, A. Joux, *Solving shortest and closest vector problems: The decomposition approach*, IACR Cryptology ePrint Archive, 2013/685.
2. G. Hanrot, X. Pujol, D. Stehlé, *Analyzing blockwise lattice algorithms using dynamical systems,* in CRYPTO 2011, LNCS, vol. 7073, pp. 1–20, Springer 2011.
3. 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.
4. C. Gentry and M. Szydlo, *Cryptanalysis of the Revised NTRU Signature Scheme* in EUROCRYPT 2002, LNCS, vol. 2332, pp. 299–320, Springer 2002.
5. O. Goldreich, S. Goldwasser, S. Halevi, *Public-key cryptosystems from lattice reductions problems*, in CRYPTO 1997. LNCS, vol. 1294, pp. 112–131, Springer 1997.
6. T. Laarrhoven, *Sieving for closest lattice vectors(with preprocessing)*, arXiv: 1607.04789v1, 16 Jul 2016.
7. X. Nie, X. Jiang, L. Hu, and J. Ding, *Cryptanalysis of Two New Instances of TTM Cryptosystem*, Cryptology ePring Archive.
8. L. Goubin, N. Courtois, *Cryptanalysis of the TTM Cryptosystem*, in ASIACRYPT 2000, LNCS, vol 1976, pp. 44–57, Springer 2000.
9. C. Gentry, C. Peikert, V. Vaikuntanathan, *Trapdoors for hard lattices and new cryptographic constructions*, in STOC 2008, pp. 197–206, ACM 2008.
10. 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.
11. T. Laarhoven, *Sieving for shortest vectors in lattices using angular locality-sensitive hashing,* in CRYPTO 2015, LNCS vol. 9215, pp. 3–22, Springer 2015.
12. 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.
13. https://csrc.nist.gov/projects/post-quantum-cryptography/post-quantum-cryptography-standardization.
14. D. Micciano, C. Peikert, *Trapdoors for Lattices: Simpler, Tighter, Faster, Smaller*, in EUROCRYPT 2012, LNCS, vol 7237, pp. 700-718, Springer 2012.
15. Ph. Q. Nguyen and O. Regev, *Learning a Parallelepiped: Cryptanalysis of GGH and NTRU Signatures*, J, Cryptol. vol. 22 (2009), pp. 139–160.

16. Ph. Q. Nguyen, B. Vallée, editors, *The LLL Algorithm, Survey and Applications,* Springer,2010.
17. A. Pellet-Mary, D. Stehle, *On the hardness of the NTRU problem,* in ASIACRYPT'21, LNCS 13090, pp.3–35, 2021.
18. T. Plantard, W. Susilo, and K. T. Win, *A Digital Signature Scheme Based on CVP_∞*, in PKC 2008, LNCS 4939, pp. 288–307, 2008.
19. D. Wiedemann, *Solving sparse linear equations over finite fields*, IEEE Trans. on Inf. Theory, vol. 32 (1986), pp. 54–62.

11 Appendix

Lemma 2. *Let P_1, P_2 be permutation matrices of size $m \times m$. Then $C = P_1 + P_2$ is of rank m if and only if the permutation $Q = P_2 P_1^{-1}$ has only odd cycles. In this case,* $\det C = \pm 2^s$*, where s is the number of cycles in Q.*

Proof. The matrix C is of full rank if and only if the system of linear equations $x(P_1 + P_2) = 0$ has only zero solution. The system is equivalent to $x = -xQ$. The latter has a non-zero solution if and only if Q has at least one cycle of even length. That proves the first part of the lemma. To prove the rest, let $r_1, \ldots, r_s$ be the lengths of the cycles in Q and $R = CP_1^{-1} = I_m + Q$, where I_m is an identity permutation.

Let r be an odd number and $P = [2, 3, \ldots, r, 1]$ be a permutation with exactly one cycle of length r. We consider P as a matrix and get

$$R_r = I_r + P = \begin{pmatrix} 1 & 1\,0 \ldots 0\,0 \\ 0 & 1\,1 \ldots 0\,0 \\ \ldots & \\ 0 & 0\,0 \ldots 1\,1 \\ 1 & 0\,0 \ldots 0\,1 \end{pmatrix}. \tag{10}$$

So $\det(R_r) = 2$. There is a permutation U such that

$$U^{-1} C P_1^{-1} U = U^{-1} R U = I_m + U^{-1} Q U = \begin{pmatrix} R_{r_1} & 0 & \ldots & 0 \\ 0 & R_{r_2} & \ldots & 0 \\ \ldots & & & \\ 0 & 0 & \ldots & R_{r_s} \end{pmatrix}, \tag{11}$$

where R_{r_i} is defined by (10). Therefore, $\det(R) = 2^s$.

For $\lambda = 2$ one may choose a random secret permutation Q of $[1, 2, \ldots, kn]$ with two cycles of close odd lengths and a random secret permutation P_1, and set $P_2 = QP_1$. Then $C = P_1 + P_2$. The inversion of R_r is a Toeplitz matrix of size $r \times r$ whose first row is $[1, -1, 1, -1, \ldots, 1]$ multiplied by $1/2$. By (11) one can easily compute the inversion of C.