A Method for Specifying Complete Signature Randomization and an Algebraic Algorithm Based on It

Alexandr Moldovyan †, Dmitriy Moldovyan †, Nikolay Moldovyan *† and Alyona Kurysheva †

St. Petersburg Federal Research Center of the Russian Academy of Sciences (SPC RAS), 39, 14th Line V.O., 199178 St. Petersburg, Russia; maa1305@yandex.ru (A.M.); mdn.spectr@mail.ru (D.M.); kurysheva.al@yandex.ru (A.K.)

* Correspondence: nmold@mail.ru

† These authors contributed equally to this work.

Abstract: To eliminate the limitations of signature randomization in known algebraic algorithms with a hidden group, the security of which is based on the computational complexity of solving large systems of power equations, a method for ensuring complete randomization is proposed. Based on this method, a new algorithm of the indicated type was developed, using a four-dimensional finite non-commutative associative algebra as an algebraic basis. We obtained estimates of the security of algorithms to direct attacks as well as from attacks based on known signatures, which confirm the effectiveness of the proposed signature randomization method. Due to the relatively small size and signature of the public and private keys, the developed algorithm is of interest as a potential practical post-quantum digital signature scheme.

Keywords: finite non-commutative algebra; associative algebra; computationally difficult problem; hidden commutative group; digital signature; post-quantum cryptography

MSC: 94A60; 94A62; 68P27

1. Introduction

Over the past few years and currently, the development of practical digital signature (DS) and public encryption algorithms suitable for the development of post-quantum cryptographic standards based on them has attracted special attention from the global cryptographic community [1,2], which has actively participated in the worldwide competition for the development of algorithms for post-quantum standards announced by the US National Institute of Standards and Technology (NIST) for the period 2017–2024 [3,4] in the following two categories:

1. post-quantum algorithms for public encryption and key encapsulation algorithms;
2. post-quantum algorithms of digital signature (DS).

Currently, work on the NIST post-quantum project and the associated NIST competition is in its final stages. Based on preliminary results, it can be concluded that post-quantum cryptographic algorithms (including algorithms selected to define post-quantum cryptographic standards) are significantly inferior to analog algorithms based on the factorization problem (FP) and discrete logarithm problem (DLP). Because of the significantly larger size of the signature and the public and secret keys, the known post-quantum public key algorithms are less practical. Apparently, this is an objective fact that needs to be accepted, since security considerations of the algorithms used take precedence over other application aspects (public-key algorithms based on FP and DLP are not resistant to attacks using a quantum computer [5,6]). The only question is to minimize the size of these parameters and the complexity of the hardware implementation while maintaining sufficiently high performance.
Thus, the post-quantum cryptography should be based on computationally complex problems other than DLP and FP. For example, post-quantum public-key cryptographic algorithms on groups [7], algebraic lattices [8], codes [9], and hash functions [10] have been proposed.

One attractive approach to developing post-quantum public-key algorithms involves exploiting the computational complexity of solving large systems of power equations [11,12]. Cryptographic algorithms based on the latter problem belong to the field of multivariate public-key cryptography (MPC) [13]. Quantum computers are ineffective for solving systems of power equations with many unknowns, so MPC algorithms are resistant to quantum attacks [14]. MPC algorithms have fairly high performance and a small digital signature size and are promising for practical use in the coming post-quantum era [15]. However, the known MPC algorithms have a significant drawback for practical use, which is the very large size of the public key.

For the development of practical post-quantum DS schemes, algebraic algorithms on finite non-commutative associative algebras (FNAA) with a hidden group, the security of which is based on the computational complexity of solving large systems of quadratic equations [16], are of great interest. A common feature of these algorithms is the repeated inclusion of a signature fitting element in the signature verification formula; for example, double [16], triple, and quadruple [17]. At the same time, four-dimensional and six-dimensional FNAA s are considered as an algebraic support of DS algorithms, suggesting the potential possibility of implementation on FNAA s with dimensions \( m > 6 \) [17].

This paper shows that the reappearance of a signature in the verification equation is due to its generation by some typical mathematical formula, causing limited randomization of the signature, which reduces the assessment of the security level compared to the assessment performed for the case of a direct attack (computing the secret key elements based on public-key elements). To eliminate this general drawback, a method for ensuring complete signature randomization in hidden group algorithms based on the complexity of solving large systems of power equations is proposed. A specific algorithm is developed that implements the introduced method.

2. The Used FNAA

Let us assume that in a finite \( m \)-dimensional vector space (with two standard operations: vector addition and scalar multiplication), given over a Galois finite field \( GF(p) \), the operation of vector multiplication (i.e., multiplication of a vector by a vector) is defined, which is closed and distributive on the left and right relative to the addition operation. Such algebraic structure is called a finite \( m \)-dimensional algebra. The multiplication of the vectors \( \sum_{i=0}^{m-1} a_i e_i \) and \( \sum_{i=0}^{m-1} b_i e_i \), where \( e_i \) are formal basis vectors, is usually given by the following formula:

\[
AB = \sum_{i=0}^{m-1} \sum_{j=0}^{m-1} a_i b_j (e_i e_j),
\]

where, instead of all possible products of pairs of basis vectors \( e_i e_j \), one-component vectors of the form \( \lambda e_k \), where \( \lambda \in GF(p^s) \) (in particular, the basis vectors \( e_k \)) are substituted in accordance with some basis vector multiplication table (BVMT); namely, the expression \( (e_i e_j) \) is replaced by a one-component vector \( \lambda e_k \) contained in the BVMT cell at the intersection of the \( i \)th row and the \( j \)th column.

For BVMTs of arbitrary-type, the vector multiplication operation defined in this way possesses the properties of closure and right and left distributivity; however, in algebraic signature schemes [16,17], the operation of exponentiation to a large-size degree (from 80 to 256 bits) is used, which can be carried out in an acceptable time only if it is possible to use the fast exponentiation algorithm that is based on sequential squaring. This possibility exists if the specified vector multiplication operation is associative. In this regard, FNAA s are considered as an algebraic support of algorithms with a hidden group. For estimation
of the security of DS algorithms with a hidden group, the knowledge of the structure of FNAA (as a decomposition into a set of commutative subalgebras) is very useful. The structure of FNAA is sufficiently well studied only for the four-dimensional case.

In connection with the latter, we will assume that a version of the algorithms [16,17] is being considered, which corresponds to the use of four-dimensional FNAA as their algebraic support. The results of the study of the structure of four-dimensional FNAA showed that they have a similar structure, regardless of the type of BVMT setting the multiplication operation. For example, the structure of the FNAA set by sparse BVMTs was studied in [18]. Due to the generality of the structure, the security estimates obtained below relate to different FNAA as used as an algebraic support. For certainty, we will assume the use of FNAA with vector multiplication defined by BVMT [18], presented in Table 1. Such certainty will be required to describe the set of all elements of a certain commutative subalgebra according to the coordinates of some representative of this subalgebra using a specific mathematical formula.

Below we use the following terms: group, hidden group, subalgebra, and scalar vector.

By the term “group” we mean a set of elements (with the vector multiplication operation) which contain a unity element relatively in which all elements are reversible.

By the term “hidden group” we mean a group contained in a considered algebra, the minimum generator system of which is used as a secret element (an element of the secret key).

By the term “subalgebra” we mean a subset of vectors of some considered algebra, which compose a ring.

By the term “scalar vector” we mean a vector \( L = \lambda E \), where \( E \) is the unit element (global two-sided unit) and \( \lambda \) is a scalar value, i.e., \( \lambda \in GF(p) \).

The general properties of the structure of four-dimensional FNAA are the following:

1. The four-dimensional FNAA contains \( \eta = p^2 + p + 1 \) various commutative subalgebras having order \( p^2 \) and intersecting strictly in the set of scalar vectors.

2. The commutative subalgebras relate to the following three types.

2.1. Subalgebras containing a multiplicative group \( \Gamma_1 \) having a cyclic structure and the order \( \Omega_1 = p^2 - 1 \). Since an element of order \( p^2 - 1 \) of this multiplicative group produces \( p^2 - 1 \) different non-zero vectors, the said group is therefore cyclic and the subalgebra is a Galois field of order \( p^2 \). Taking into account that all Galois fields of the same order are isomorphic, you can state that subalgebras of the considered type are isomorphic to the field \( GF(p^2) \). The number of such subalgebras is equal to \( \eta_1 = 2^{1}(p + 1) \).

2.2. Subalgebras containing the multiplicative group \( \Gamma_2 \) having a two-dimensional cyclic structure and order \( \Omega_2 = (p - 1)^2 \), i.e., \( \Gamma_2 \) is generated by two generators, each of which has order \( p - 1 \). Every one of the subalgebras of this type contains \( 2p - 1 \) irreversible vectors, and the number of subalgebras of this type is \( \eta_2 = 2^{1}(p + 1) \).

2.3. Subalgebras containing the multiplicative group \( \Gamma_3 \) having a cyclic structure and order \( \Omega_3 = p(p - 1) \). The number of such subalgebras is \( \eta_3 = p + 1 \).

3. All elements of a given commutative subalgebra can be described as a set of vectors \( V \), whose coordinates are given by the coordinates of some vector \( C \) (that is not a scalar vector) contained in the subalgebra, and a pair of scalar variables \( h, u \in GF(p) \). In particular, for FNAA set by Table 1, all elements of a subalgebra containing a multiplicative group of the \( \Gamma_2 \) type are described by the coordinates of the vector \( C = (c_0, c_1, c_2, c_3) \) by the following formula [18]:

\[
C = (c_0, c_1, c_2, c_3) = \left( h, \frac{h c_1}{c_0}, u, u + h \frac{c_3 - c_2}{c_0} \right). \tag{2}
\]

In the algorithms from the articles [16,17], FNAA are given over the field \( GF(p) \) with the characteristic \( p = 2q + 1 \), where \( q \) is a prime number of a sufficiently large size.
Table 1. Setting a four-dimensional FNAA \((\lambda \neq 0)\) with a global two-sided unit of the form \((0,0,1,1)\) [18].

<table>
<thead>
<tr>
<th>( \times )</th>
<th>( e_0 )</th>
<th>( e_1 )</th>
<th>( e_2 )</th>
<th>( e_3 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( e_0 )</td>
<td>0</td>
<td>( \lambda e_3 )</td>
<td>( e_0 )</td>
<td>0</td>
</tr>
<tr>
<td>( e_1 )</td>
<td>( \lambda e_2 )</td>
<td>0</td>
<td>0</td>
<td>( e_1 )</td>
</tr>
<tr>
<td>( e_2 )</td>
<td>0</td>
<td>( e_1 )</td>
<td>( e_2 )</td>
<td>0</td>
</tr>
<tr>
<td>( e_3 )</td>
<td>( e_0 )</td>
<td>0</td>
<td>0</td>
<td>( e_3 )</td>
</tr>
</tbody>
</table>

3. Limited Randomization and Attack Based on Known Signatures

In algebraic algorithms [16,17], the DS includes a randomizing part in the form of one or more natural numbers and a fitting part represented by the fitting vector \(S\). Since concatenation of several natural numbers can be represented as a single number of larger size, for all DS algorithms the signature can be represented as a pair of values \((e, S)\), where \(e\) is a natural number that plays the role of a randomizing parameter. The use of verification equations with multiple occurrences of the fitting element determines the use of a standard formula for calculating the signature in the algorithms [16,17], which has the form

\[
S = BC^n H^d A^{-1},
\]

in which the natural powers \(n\) and \(d\) are calculated beforehand (depending on the value of \(e\) and some other hidden randomization parameters), and the vectors \(A, B, G,\) and \(H\) are fixed secret vectors that are part of the secret key associated with the public key; the vectors \(G\) and \(H\) have the same order \(q\) and compose a minimum system of generators \(<G, H>\) of the hidden commutative group (that is a subgroup of order \(q^2\) of a multiplicative group of the \(\Gamma_2\) type of some commutative subalgebra, contained in the four-dimensional FNAA used as an algebraic support of the DS algorithm).

Note that the vector \(G^n H^d\) takes on only values from the hidden group. Due to the uniqueness of the natural values \(n\) and \(d\) for each signature and their dependence on the signature randomization parameters, the \(G^n H^d\) value is generally unique for each signature; however, the number of such vector values does not exceed the value \(q^2\) (the order of the hidden group). Due to the left and right multipliers (which are non-permutable, i.e., \(AB \neq BA\)), the vector \(S\) takes random values from the entire FNAA, but no more than \(q^2\) different values, whereas FNAA includes \(q^4\) different four-dimensional vectors.

Thus, the fitting element of the signature \(S\) takes on only a very small fraction of the values from the FNAA, i.e., the randomization of the DS in the algorithms [16,17] is incomplete and even significantly limited. Incompleteness of signature randomization may cause potential vulnerability to attacks using known signatures, which can be implemented as follows.

Let there be some set including \(z\) known authentic signatures. The fitting element \(S_i\) of each \(i\)th signature was calculated using Formula (3). At the same time, the vector fixator \(R_i\) can be calculated using the \(i\)th signature and the DS verification equation, which was calculated during the signature generation procedure using random natural values \(k < q\) and \(t < q\) according to the formula

\[
R_i = FG^k H^t F^{-1},
\]

where the vector \(F\) is an element of the secret key (in a particular case, \(F\) is equal to the secret element \(A\) from Formula (3)). Thus, two vector equations of the form (2) and (3) are associated with each signature, in which the unknowns are the vectors \(A, B, F\) (each one sets four scalar unknowns, which are its coordinates), \((G^n H^d)\), and \((G^k H^t)\). In this case, the values of the latter always belong to the hidden group and can be calculated using Formula (2) for a given representative of the hidden group. Taking \(C\) as an unknown common to all signatures, the unknown values \((G^n H^d)\) and \((G^k H^t)\) for each signature can be set by two different pairs of scalar unknowns \(h, u \in GF(p)\).
Thus, each signature (of the z known genuine signatures) sets according to Formulas
(3) and (4) two vector quadratic equations

\[ S_i A = B(G^n H^d); \quad R_i F = F(G^k H^t), \]

which, using Formula (2), are reduced to 8 cubic scalar equations with 16 fixed scalar
unknowns (i.e., common to all known signatures and being the coordinates of secret vectors
A, B, F, and C) and 4 unique scalar unknowns. For z known authentic signatures, we
have a system of 4z power equations with 16 + 2z scalar unknowns. From the condition
of equality of the number of equations and the number of unknowns, we obtain the equation

\[ 8z = 16 + 4z, \]

from which we determine a presumably sufficient number of known signatures to calculate
the secret values of A, B, and F: z = 4.

The last value sets the number of equations and unknowns in the solved system of
cubic equations equal to 32. With this number of power equations in the system, the
computational complexity of its solution is quite high. However, finding the unknown F
can be carried out by solving only the equations obtained by Formula (4), i.e., the vector
equations of the form

\[ R_i F = F(G^k H^t). \]

For z known signatures, we obtain 4z scalar equations with 8 fixed scalar unknowns
(coordinates of vectors F and C) and 2z unique scalar unknowns. From the equation

\[ 4z = 8 + 2z \]

we find the value z = 4, which sets the number of known signatures for the case of equality
of the number of unknowns and the number of equations in the system. The last value
is 4z = 16. The computational complexity of solving systems of power equations with
such a number of equations and unknowns is quite low, and the attack using the known
signatures can be considered effectively implemented in practice.

Finding the unknowns A and B can be carried out by solving only the equations
obtained by Formula (3), i.e., the vector equations of the form

\[ S_i A = B(G^n H^d). \]

In this case, for z known signatures, we obtain 4z scalar equations with 12 fixed scalar
unknowns (coordinates of vectors A, B, and C) and 2z unique scalar unknowns. From the
equation

\[ 4z = 12 + 2z \]

we find the value z = 6, which sets the number of known signatures for the case of equality
of the number of unknowns and the number of equations in the system. The last value
is 4z = 24. This variant of the attack based on known signatures is also potentially feasible
in practice.

The last two attacks significantly reduce the expected security level of algorithms [16,17],
obtained from estimates of the computational complexity of solving systems of equations
connecting elements of the secret key with elements of the public key (in the latter system, there
are 12 more scalar unknowns). The considered variants of attack based on known signatures
show the possibility of fairly effective use of the limitations of signature randomization in the
known algebraic DS algorithms with a hidden group, based on the computational difficulty
of solving systems of power equations, and the relevance of developing a method for setting
complete randomization in algorithms of this type.
4. A Way to Ensure Complete Randomization of the Signature

By the completeness of the signature randomization, we will understand the potential possibility to obtain any reversible vector as the value of the fitting element $S$. To achieve this, it seems reasonable to include in Formula (3) a multiplier in the form of a uniformly random reversible vector $V$, selected from the set of all reversible values of FNAA, used as an algebraic support of the DS scheme. It is proposed to set a formula for calculating the fitting element $S$ in the following form:

$$S = DG^nH^dV,$$

where $D$ is a secret reversible vector (element of the secret key).

However, the presence of a random multiplier $V$ makes it impossible to use verification equations with multiple occurrences of $S$ and thus necessitates the search for other methods to prevent the use of the value of $S$ as a fitting parameter for forging DS. As such, we propose using the doubling of the signature verification equation. By “doubling the verification equation” we mean using two similar verification equations; in each equation, the fitting element of the signature is included once.

The use of the doubled DS verification equation implies the formation of two random vector fixators (according to their values and the document, the randomizing element of the signature is calculated), the values of which will be recovered during the signature verification procedure performed by using the correctly calculated fitting element $S$. The calculation of the required value of $S$ can be provided, using formulas for generating vector fixators, which are consistent with Formula (6) and with verification equations. The following two formulas have been developed as such:

$$R_1 = AG^{k_1}H^{t_1}V^{j^h||h}; \quad R_2 = FG^{k_2}H^{t_2}V^{j^h||h},$$

in which the pre-generation of random natural numbers $k_1, t_1, k_2, t_2$, and a random reversible vector $V$ is assumed, as well as the pre-calculation of the value of the hash function $H^h(M)$ (where $||$ denotes the concatenation operation) from the signed document $M$, the hash function being represented as a concatenation of two bit strings of the same size (treated as natural numbers). The vector $J$ has the order $p^2-1$ (i.e., it is a generator of a multiplicative group of the $\Gamma_1$ type) and is a common parameter for all users of the DS algorithm. The purpose of the use of various multipliers $j^h$ and $j^h||h$ in the first and second Formula (6) is to prevent the possibility of using the element $S$ as a fitting parameter in the forging signature attack (calculating a signature from verification equations without knowing the secret key).

Using a pair of fixed Formulas (5) and (6) allows us to develop various versions of verification equations specifying various variants of algebraic DS schemes based on the computational difficulty of solving large systems of power equations. However, for all such algorithms, the complexity of the attack based on known signatures will be the same for a given value of the characteristic of the $GF(p)$ field, over which a four-dimensional FNAA is set as an algebraic support of the DS algorithm. This is predetermined by fixing Formulas (5) and (6). Let us estimate the computational complexity of this attack as the complexity of solving the emerging system of power equations, the latter being estimated by the number of equations and unknowns in the system.

Given that the smallest number of equations arises when calculating fixed unknown parameters separately (it is obvious that the method of separating variables is not applicable to unique unknowns), first consider the option of constructing a system of equations for $z$ different genuine signatures using a pair of Equation (6). It is easy to establish that in this version of the attack, we have $8z$ scalar equations, $12$ fixed scalar unknowns (associated with unknown vectors $A$, $F$, and $C$, where $C$ is a representative of the hidden group) and $8z$ unique scalar unknowns (uniqueness is understood in the sense that they relate to only one pair of vector Equation (6)). For any value of $z$, we have a number of unknowns that exceeds the number of equations by 12. The underdeterminacy of the emerging system
shows that a small number of solutions, if possible, can be obtained only for very large values of $z$. A similar situation occurs with other attempts to construct a system of equations using one or two formulas from Formulas (5) and (6).

Thus, the most effective attack option based on known signatures seems to be the construction of a system based on the triple of Equations (5) and (6) and its subsequent solution. In this case, for $z$ known signatures, we have $12z$ scalar equations, 16 fixed scalar unknowns (associated with unknown vectors $A$, $F$, $D$, and $C$) and $10z$ unique scalar unknowns (four associated with a random vector $V$ and six associated with three random vectors $(G^kH^1)$, $(G^kH^t)$, and $(G^kL^2H^2)$ from the hidden group defined by the vector $C$). Equating the number of equations and the number of unknowns, we obtain the equation

$$12z = 16 + 10z$$

from which we find the value $z = 8$. The latter sets the number of equations equal to 96. For such a large number of power equations, the computational complexity of finding a solution to the system exceeds the level of $2^{256}$ bit operations (see Table 1 in [19]), i.e., an attack based on known signatures is computationally impossible. The latter means that the proposed mechanism for ensuring complete randomization is sufficient. Let us consider the implementation of a specific DS algorithm based on it.

5. The DS Algorithm Implementing the Proposed Method of Signature Randomization

We will assume that the four-dimensional FNAA used as the algebraic support of the DS algorithm is set by Table 1 over the field $GF(p)$, where prime $p = 2q + 1$ with 128-bit prime $q$. A collision-resistant 256-bit hash function $H(\cdot)$, the said FNAA, and some vector $J$ of order $p^2 - 1$ are common to all users of the digital signature algorithm.

The formation of a public key in the form of a set of seven vectors $Y_1, T_1, U_1, Y_2, T_2, U_2$, and $W$ (with a total size of $\approx 448$ bytes) is carried out according to the following algorithm:

1. Generate two random vectors $G$ and $H$ of the order $q$ that form the basis $<G,H>$ of a hidden group possessing two-dimensional cyclicity and the order $q^2$.
2. Generate a random vector $P$ of the order $q$, and random invertible vectors $A, B, D,$ and $F$ that satisfy the inequalities $AB \neq BA$, $AD \neq DA$, $AF \neq FA$, $AP \neq PA$, $AG \neq GA$, $BD \neq DB$, $BF \neq FB$, $BP \neq PB$, $BG \neq GB$, $DF \neq FD$, $DP \neq PD$, $DG \neq GD$, $FP \neq PF$, $FG \neq GF$, and $PG \neq GP$.
3. Generate random natural numbers $x < q$ and $w < q$ and calculate vectors

$$Y_1 = AG^x A^{-1}; \quad T_1 = AH^{w-x} P^w D^{-1}; \quad U_1 = DP D^{-1};$$

$$Y_2 = FGH F^{-1}; \quad T_2 = FP^x B^{-1}; \quad U_1 = BP B^{-1}; \quad W = B P^{w-x} D^{-1}.$$  \hspace{1cm} (7)

Numbers $x$ and $w$ and the vectors $A, B, D, F, G, H,$ and $P$ compose a secret key with a total size of $\approx 480$ bytes.

The generation of the basis $<G,H>$ of the hidden group can be performed as follows:

1. Generate a random vector $G'$.
2. If $G'$ is a non-scalar vector and $G'^q = E$ (probability of this event is 0.25), then $G \leftarrow G'$; otherwise, go to step 1.
3. Generate a random non-scalar vector $H'$.
4. If $H'$ is a non-scalar vector and $H'^q = E$ (probability of this event is 0.25), then $H \leftarrow H'$; otherwise, go to step 1.

The algorithm for generating DS includes the following steps:

1. Calculate the value of the hash function from the signed electronic document $M$: $h^1|h = H(M)$, where the 256-bit hash value is represented as a concatenation of 128-bit natural numbers $h'$ and $h$.
2. Generate a random reversible vector $V$ and random natural numbers $k_1, t_1, k_2,$ and $t_2 (< q)$. Next, using Formula (6), calculate the values of the vector fixators $R_1$ and $R_2$.
3. Calculate the hash value from the document $M$ to which the vectors $R_1$ and $R_2$ are attached: $e'|e = H(M, R_1, R_2)$, where the 256-bit hash value is represented as a concatenation of two 128-bit natural numbers $e'$ and $e$.

4. Calculate the value of the degree $d$: $d = (t_1 + x - w) \mod q$.

5. Calculate the value of the first auxiliary fitting element of the DS using the formula $s = e'^{-1}(t_2 - t_1) \mod q$.

6. Calculate the degree $n$ by formula $n = (k_2 - e's) \mod q$ (or by formula $n = (k_2 - t_2 + t_1) \mod q$).

7. Calculate the value of the second auxiliary fitting element of the DS by formula $\sigma$: $\sigma = x^{-1}e'^{-1}(k_1 - k_2 + e's) \mod q$ (or by formula $\sigma = x^{-1}e'^{-1}(k_1 - k_2 + t_2 - t_1) \mod q$).

8. Calculate the degree $b$: $b = q - (e + w) \mod q$.

9. Calculate the fitting signature element $S$ using the following formula:

$$S = DP^bG^nH^dV.$$  \hspace{1cm} (9)

A digital signature to an electronic document $M$ is a set of values $(e', \epsilon, s, \sigma, S)$ with a total size of 128 bytes. Given that $H^d = H^{k_2}$, the computational complexity of the signature generation algorithm can be roughly estimated as 6 exponentiation operations to a 128-bit degree and 2 exponentiation operations to a 256-bit degree (calculation of the vectors $j^{b,h}$ and $j^{e,h}$), i.e., estimates as $\approx 15,400$ multiplication operations in the field $GF(p)$.

The algorithm for verifying the signature $(e', \epsilon, s, \sigma, S)$ to the document $M$ is performed using the public key $(Y_1, T_1, U_1, Y_2, T_2, U_2, W)$ and has the following form:

1. Calculate the hash value of the document $M$: $h||h = H(M)$.

2. Calculate the values of the vectors $R_1'$ and $R_2'$ by the following two formulas:

$$R_1' = Y_1^{e'^rT_1U_1'S}h^{b,h}; \quad R_2' = Y_2^{e'^sT_2U_2'WS}h^{b,h}. $$

3. Calculate the hash value from the document $M$ to which the vectors $R_1'$ and $R_2'$ are concatenated: $e'||e = H(M, R_1', R_2')$, where the 256-bit hash value is represented as a concatenation of two 128-bit integers $e'$ and $e$.

4. If the equalities $e' = e'$ and $e = e$ hold true simultaneously, then the DS is accepted as genuine; otherwise, the DS is rejected as false.

The computational complexity of the DS verification algorithm can be roughly evaluated as 4 exponentiations of four-dimensional vectors to a 128-bit degree and 2 exponentiations to a 256-bit degree, i.e., as $\approx 12,300$ multiplication operations in the field $GF(p)$.

You can easily show the correctness of the proposed DS algorithm, namely, that a correctly calculated signature $(e', \epsilon, s, \sigma, S)$ to the document $M$ passes the verification procedure as a genuine signature. To prove the correctness of the proposed DS algorithm, you can calculate the vectors $R_1'$ and $R_2'$:

$$R_1' = Y_1^{e'^rT_1U_1'S}h^{b,h} =$$

$$= (AG^sA^{-1})^{e'} e^{A^{-1}DP^{w-x}P^{w-x}D^{-1}(DP^{-1})}^{e'} DP^bG^nH^dV h^{b,h} =$$

$$= AG^{xe'^r}H^{w-x}P^{w-x}P^{w-x}D^{-1}(DP^{-1})^{e'} DP^bG^nH^dV h^{b,h} =$$

$$= AG^{k_1-k_2+e's+k_2-e's}H^{w-x+t_1+t_1+x-x}V h^{b,h} =$$

$$= AG^{k_1}H^{t_1}V h^{b,h} = R_1' =$$

$$R_2' = Y_2^{e'^sT_2U_2'WS}h^{b,h} =$$

$$= (FGHF^{-1})^{e'} e^{F^{-1}P^{x}B^{-1}}(BP^{-1})^{e'} BP^{w-x}D^{-1}(DP^{-1})^{e'} DP^bG^nH^dV h^{b,h} =$$

$$= FG^{e'^s}H^{e'^s}P^{x+w}P^{x+w}D^{-1}(DP^{-1})^{e'} DP^bG^nH^dV h^{b,h} =$$

$$= FG^{k_2-k_1+k_2-k_1}H^{t_1}V h^{b,h} =$$

$$= FG^{k_2}H^{t_1}V h^{b,h}.$$
The computation of the hash value $e' = H(M, R_1', R_2')$ and the comparison of $e' | e$ with the value of $e' = H(M, R_1, R_2)$ gives:

$$\{ R'_1 = R_1; R'_2 = R_2 \} \Rightarrow H(M, R'_1, R'_2) = H(M, R_1, R_2) \Rightarrow \{ e' = e'; e = e \}.$$

6. Discussion

A quantum computer is not efficient for solving large systems of power equations. Therefore, the proposed algorithm, whose security is based on the computational complexity of the latter problem, is secure to quantum attacks if it is secure to attacks with using ordinary computers.

A detailed examination of possible structural attacks (attacks exploiting the specifics of algorithm construction) using conventional computers is an independent research task. Let us consider the security of the proposed algorithm to a direct attack related to solving a basic computationally difficult problem.

The developed DS algorithm implements the proposed randomization method corresponding to Formula (5); furthermore, additional randomization is introduced by introducing a unique multiplier $P^b$ for each signature in Formula (9) used in the developed algorithm to calculate the fitting element of the signature $S$. This leads to the fact that the computational complexity of the attack based on known signatures exceeds the estimate of $2^{256}$ obtained for the case of using Formula (5). This required the use of an additional exponentiation operation in the signature generation procedure, resulting in a decrease in its performance by about 12%.

The implementation of algorithms in strict accordance with Formula (5) also practically prevents attacks based on known signatures; however, using Formula (9) allows you to include multipliers equal to various degrees of the secret vector $P$ in formulas for calculating elements of the public key, the presence of which is aimed at ensuring resistance to attacks using the search for potential equivalent keys. At the same time, the security level to a direct attack (calculation of secret vectors based on public-key elements) increases due to the appearance of additional square vector equations in the system of equations written in accordance with Formulas (7) and (8). The system of square vector equations considered in a direct attack has the following form:

$$\begin{align*}
Y_1 A &= AQ_0; \quad T_1 D = AQ_2 P_1; \quad U_1 D = DP; \\
Y_2 F &= FQ_1; \quad T_2 B = FP_2; \quad U_2 B = BP; \quad WD = BP_3,
\end{align*}$$

(10)

where $Q_0 = G^x$, $Q_1 = GH$, $Q_2 = H^{u-x}$, $P_1 = P^u$, $P_2 = P^x$, $P_3 = P^{u-x}$.

The solution of the system of power vector Equation (10) is reduced to solving a system of 28 power (square and cubic) scalar equations with 34 scalar unknowns. The coordinates of the 6 unknown vectors $A, B, D, F, P,$ and $Q_0$ define 24 scalar unknowns, and the coordinates of each of the unknown vectors $Q_1, Q_2 (P_1, P_2, P_3)$ are expressed (according to Formula (2)) through the coordinates of the vector $Q_0 (P)$ and 2 scalar unknowns $h, u \in GF(p)$, specifying 10 additional scalar unknowns. Thus, a direct attack is associated with the solution of a system of 28 power equations with 34 unknowns, given in the $GF(p)$ field of a prime 129-bit order. Considering that the computational complexity $\Psi$ of solving a system of 26 quadratic equations with 26 unknowns in a finite field of 8-bit order has a level of $2^{80}$ (see Table 1 in [19]) and using estimates [20] that take into account the dependence of $\Psi$ on the value of the field order, we obtain the following preliminary estimate of the security level of the proposed algorithm to a direct attack: $\Psi > 2^{128}$. Clarifying this assessment is an independent task. Additionally, further research on security evaluation should be conducted in the direction of developing structural attacks that use the algorithm design features and of evaluating their computational complexity.
It should be noted that the number of scalar unknowns exceeds the number of equations, indicating the potential existence of a set of equivalent keys, but this does not change the obtained preliminary assessment of the security to a direct attack.

7. Conclusions

To eliminate the demonstrated potential vulnerability of the algebraic DS algorithms [16, 17] to attacks based on known signatures due to limited randomization of the signature, a method is proposed to ensure complete randomization of the signature. Based on the latter, a new DS algorithm based on a four-dimensional FNAA has been developed, which is of interest as a practical post-quantum public-key signature algorithm. An assessment of its resistance to direct attack associated with the solution of a large system of power equations is given. As a direction for further research, it is of interest to consider implementation of similar algorithms on FNAAAs with dimensions $m \geq 6$, which provides a potential opportunity to increase the security level by increasing the number of power equations in the system, a problem which arises when considering direct attacks.

Author Contributions: Conceptualization, A.M.; methodology, A.M. and A.K.; validation, D.M.; formal analysis, N.M. and D.M.; investigation, A.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Russian Science Foundation, grant number 24-21-00225, https://rscf.ru/project/24-21-00225/ (accessed on 25 June 2024).

Data Availability Statement: Data are contained within the article.

Acknowledgments: When editing the article, the authors used online services https://translate.yandex.ru/ (accessed on 25 June 2024), https://languagetool.org/ru (accessed on 25 June 2024), and https://gpt-chartbot.ru (accessed on 12 June 2024) only to check the English grammar in some sentences from Sections 1, 3, 4, and 7.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NIST</td>
<td>National Institute of Standards and Technologies of the USA</td>
</tr>
<tr>
<td>DS</td>
<td>Digital signature</td>
</tr>
<tr>
<td>FNAA</td>
<td>Finite non-commutative associative algebra</td>
</tr>
<tr>
<td>BVMT</td>
<td>Basis vector multiplication table</td>
</tr>
<tr>
<td>DLP</td>
<td>Discrete logarithm problem</td>
</tr>
<tr>
<td>FP</td>
<td>Factorization problem</td>
</tr>
</tbody>
</table>

References


Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.