UDC 003.26
doi:10.31799/1684-8853-2023-3-59-69
EDN: GXPTKZ

Articles

Post-quantum signature algorithms with a hidden group and doubled verification equation

A. A. Moldovyan[a], *Dr. Sc., Tech., Professor, Chief Researcher, orcid.org/0000-0001-5480-6016*
N. A. Moldovyan[a], *Dr. Sc., Tech., Professor, Chief Researcher, orcid.org/0000-0002-4483-5048, nmold@mai.ru*
[a]*St. Petersburg Federal Research Center of the RAS, 39, 14th Line, 199178, Saint-Petersburg, Russian Federation*

Introduction: *One of the current topical problems of cryptography is the development of post-quantum digital signature algorithms with relatively small sizes of the public key and signature.* ***Purpose:*** *To develop a new method for designing post-quantum algebraic signature algorithms with a hidden group, based on the computational complexity of solving large systems of quadratic multivariate equations, which allows to reduce the size of the public key and signature as compared to the known analogues.* ***Results:*** *We propose a new method for designing digital signature algorithms with a signature of the form (e, S), where e is a natural number (randomization parameter) and S is a vector (fitting parameter. The method makes it possible to reduce the dimension of finite non-commutative associative algebras used as an algebraic support. The method is distinguished by the use of the technique of doubling the verification equation for fixing the hidden group, which allows one to set the formation of the vector S depending on the random reversible vector and thereby eliminates the influence of the number of signed documents on the security, which is typical of the known analogous algorithms. The method has been tested by the development of a specific post-quantum signature algorithm, various modifications of which use algebras of different dimensions. A preliminary security assessment of the proposed algorithm has been performed.* ***Practical relevance:*** *Due to comparatively small sizes of signature and public key, the introduced signature algorithm represents significant practical interest as a prototype of a post-quantum signature standard.*

Keywords *— post-quantum crypto schemes, computer security, digital signature, discrete logarithm problem, cryptography, finite non-commutative algebras, associative algebras, commutative groups.*

For citation: Moldovyan A. A., Moldovyan N. A. Post-quantum signature algorithms with a hidden group and doubled verification equation. *Informatsionno-upravliaiushchie sistemy* [Information and Control Systems], 2023, no. 3, pp. 59–69. doi:10.31799/1684-8853-2023-3-59-69, EDN: GXPTKZ

Introduction

Development of post-quantum standards on public-key cryptographic algorithms is one of current challenges faced by the global cryptographic community [1, 2]. This challenge is due to the fact that modern standards for public key cryptographic algorithms are based on the computational difficulty of the discrete logarithm problem and of the factoring problem, each of which can be solved on a quantum computer in polynomial time [3, 4].

Post-quantum public-key cryptalgorithms are to be based on computationally difficult problems different from the said two ones. One can mention post-quantum algorithms on codes [5, 6], and on hash functions [7]. Regarding post-quantum digital signature algorithms, their main disadvantage is the large total size of the public key and signature. To overcome this shortcoming, signature schemes with a hidden group were proposed in [8–11], using computational complexity of so called hidden discrete logarithm problem (HLP). However, some of such signature schemes (for example, introduced in [8, 9]) are vulnerable against algebraic attacks [12].

A new paradigm for the development of algorithms with a hidden group has been proposed in [13, 14]. That paradigm exploits the computational difficulty of finding a solution to a large system of quadratic multivariate equations with many unknowns. The latter problem is considered as an attractive post-quantum primitive [15, 16]. It had been put into the base of security of multivariate cryptographic algorithms, like EFLASH [17], MQDSS [18], Rainbow [19, 20], GeSMM [21], and UOV [22]. However, the multivariate cryptographic algorithms have a very significant drawback, which consists in the extremely large size of the public key (up to several hundred kilobytes (several megabytes) for the case of 128-bit (256-bit) security [20, 21]).

The algebraic algorithms with a hidden group exploiting computational complexity of solving a large system of quadratic multivariate equations provides possibility to develop signature algorithms with small size of both the public key and the signature. In algorithms of this type, a digital signature is a pair of values (e, $\mathbf{S}$), the number e and the vector $\mathbf{S}$. For example, the signature algorithm [13] uses a collision-resistant hash function $f(\cdot)$ and the next verification equation with two entries of the vector $\mathbf{S}$:

$$\mathbf{R}^* = \left(\mathbf{Y}_1\mathbf{S}\mathbf{Z}_1\right)^e \left(\mathbf{Y}_2\mathbf{S}\mathbf{Z}_2\right)^{e^2}.$$

In the latter equation the vectors $\mathbf{Y}_1$, $\mathbf{Z}_1$, $\mathbf{Y}_2$, $\mathbf{Z}_2$ are elements of the public key, which are calculated as follows: $\mathbf{Y}_1 = \mathbf{A}\mathbf{G}^u\mathbf{B}$; $\mathbf{Z}_1 = \mathbf{C}\mathbf{J}\mathbf{A}^{-1}$; $\mathbf{Y}_2 = \mathbf{A}\mathbf{J}^w\mathbf{B}$; $\mathbf{Z}_2 = \mathbf{C}\mathbf{G}\mathbf{A}^{-1}$ (when performing a direct attack against the algorithm, these four formulas define four vector quadratic equations with the unknown vectors $\mathbf{G}^u$, $\mathbf{J}$, $\mathbf{J}^w$, and $\mathbf{G}$), where u, w are private integers; $\mathbf{G}$, $\mathbf{J}$, $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$ are private reversible vectors such that $\mathbf{GJ} = \mathbf{JG}$ and $\mathbf{G}$, $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$ are pairwise non-permutable vectors. The signature $(e, \mathbf{S})$ is valid, if $f(M||\mathbf{R}^*) = e$, where M is a signed document and || denotes the concatenation operation. Note the verification equation connects the signature with the public key and the latter equality connects both the signature and the public key with the document M. The use of verification equations with two [13] and multiple [14, 23] entries of the vector $\mathbf{S}$ in the verification equation is focused on increasing resistance to attacks using $\mathbf{S}$ as a fitting parameter.

Unfortunately, for a fixed value of the public key, the number of different vectors that can potentially be a signature element is limited by the order of the hidden commutative group, since the vector $\mathbf{S}$ is computed by the formula $\mathbf{S} = \mathbf{B}^{-1}\mathbf{G}^n\mathbf{J}^r\mathbf{C}^{-1}\mathbf{X}$ [13], where integers n and r are computed depending on the document to be signed; vectors $\mathbf{G}$ and $\mathbf{J}$ compose a minimum generator system of the commutative hidden group; vectors $\mathbf{B}$ and $\mathbf{C}$ are elements of the private key. It can be shown that, given five (or more) different signatures, it is possible to compose a system of five vector quadratic equations, which can be solved independently of other secret values. The latter provides a significant reduction in the complexity of solving a system of equations that connects the elements of the public key with the elements of the secret key.

Therefore, to ensure the required security level of the algorithm from [13], it is required to use finite non-commutative associative algebras (FNAAs) of a sufficiently large dimension m ($m \geq 10$ for 128-bit security and $m \geq 20$ for 256-bit security), which limits the possibility of reducing the size of the public key and signature and increasing performance.

This article proposes a new method for developing algebraic signature algorithms with a hidden group, which, for a given security level, provides possibility to use FNAAs of comparatively small dimension as algebraic support. A new post-quantum signature algorithm is introduced as implementation of the method. The signature has the form $(e, \mathbf{S})$ and the indicated restriction is eliminated due to the fact that the signature element $\mathbf{S}$ can take any value in the FNAA used as an algebraic support. To insure such possibility, the technique of doubling the verification equation is used for setting the hidden group. Previously [24], this technique was used as a way to define a hidden group in signature algorithms on finite commutative algebras, which are based on computational difficulty of the HLP. Earlier, such technique was used by authors to implement the post-quantum resistance criterion, when designing the signature algorithms on FNAAs, based on the HLP [9, 24]. The introduced method is illustrated by a signature algorithm in which two different hash functions are used as an auxiliary technique for providing security against attacks using the signature element $\mathbf{S}$ as a fitting element.

Preliminaries

The technique of doubling the verification equation (see, for example, [9]) consists in specifying two different equations defining computation of the next two vectors $\mathbf{R}_1$ and $\mathbf{R}_2$, depending on the signature $(\mathbf{S}, e)$ and the public key $(\mathbf{P}_{11}, \mathbf{P}_{12}, \ldots, \mathbf{P}_{1b}, \mathbf{P}_{21}, \mathbf{P}_{22}, \ldots, \mathbf{P}_{2d})$:

$$\mathbf{R}_1 = f_1(\mathbf{P}_{11}, \mathbf{P}_{12}, \ldots, \mathbf{P}_{1b}, \mathbf{S}, e);$$

$$\mathbf{R}_2 = f_1(\mathbf{P}_{21}, \mathbf{P}_{22}, \ldots, \mathbf{P}_{2d}, \mathbf{S}, e),$$

where b and d are some small natural numbers; the vectors $\mathbf{P}_{11}, \mathbf{P}_{12}, \ldots, \mathbf{P}_{1b}, \mathbf{P}_{21}, \mathbf{P}_{22}, \ldots, \mathbf{P}_{2d}$ are elements of the public key.

And then, using a collision-resistant hash function $f(\cdot)$, connection of the signature and public key with an electronic document M is confirmed by checking the validity of the equality $f(M||\mathbf{R}_1||\mathbf{R}_2) = e$.

In the introduced algorithm with a hidden group, FNAAs are used as algebraic support. An m-dimensional FNAAs is defined as a finite vector space (defined over a field $GF(p)$) with the non-commutative associative multiplication operation of the vectors $\mathbf{A} = a_0\mathbf{e}_0 + a_1\mathbf{e}_1 + \ldots + a_{m-1}\mathbf{e}_{m-1}$ and $\mathbf{B} = b_0\mathbf{e}_0 + b_1\mathbf{e}_1 + \ldots + b_{m-1}\mathbf{e}_{m-1}$ (where $\mathbf{e}_i$, $i = 0, 1, \ldots, m-1$, are basis vectors), defined by the next formula:

$$\mathbf{AB} = \sum_{i=0}^{m-1}\sum_{j=0}^{m-1} a_i b_j \mathbf{e}_i \mathbf{e}_j, \qquad (1)$$

in which the values a_i and b_i are multiplied as the field elements. Using a basis vector multiplication table (BVMT), every product $\mathbf{e}_i\mathbf{e}_j$ is replaced by a one-component vector (see more details in [13, 24]).

Table 1 sets a 4-dimensional FNAA with the two-sided global unit $\mathbf{E} = (1, 1, 0, 0)$, for which the decomposition into commutative subalgebras is studied well in [10]:

1) the said algebra contains $p^2 + p + 1$ commutative subalgebras having order p^2; every non-scalar vector is contained in a unique subalgebra; every scalar vector is included in all subalgebras, i. e., the latter intersect exactly in the set of scalar vectors $\{\mathbf{L}: \mathbf{L} = h\mathbf{E}, h = 0, 1, \ldots, p-1\}$;

2) the multiplicative group Γ of the FNAA has order equal to

$$\Omega = p(p^2 - 1)(p - 1); \qquad (2)$$

3) the group Γ includes $p(p+1)/2$ commutative subgroups Γ_1 possessing 2-dimensional cyclicity (i. e., a minimum generator system of the subgroup Γ_1 contains two vectors of the same order) and having order equal to

$$\Omega_1 = (p-1)^2; \quad (3)$$

4) the group Γ includes $p(p-1)/2$ commutative cyclic subgroups Γ_2 of the order

$$\Omega_2 = p^2 - 1 = (p-1)(p+1); \quad (4)$$

5) the group Γ includes $p+1$ commutative cyclic subgroups Γ_3 of the order

$$\Omega_3 = p(p-1). \quad (5)$$

The vector $\mathbf{G} = g_0\mathbf{e}_0 + g_1\mathbf{e}_1 + g_2\mathbf{e}_2 + g_3\mathbf{e}_3$, such that $g_2 \neq 0$, and $g_3 \neq 0$, determines the commutative subalgebra described as the next set of vectors $\mathbf{X}$ [10], which includes $\mathbf{G}$:

$$\mathbf{X} = (x_0, x_2, x_3, x_4) = \left(d,\ d + k\frac{g_1 - g_0}{g_2},\ k,\ k\frac{g_3}{g_2}\right), \quad (6)$$

where $d, k = 0, 1, \ldots, p-1$. Type of the commutative group including all reversible vectors of the set (6) depends on the value of

$$\Delta = (g_0 - g_1)^2 + 4\lambda g_2 g_3.$$

If $\Delta \neq 0$ is a quadratic residue in $GF(p)$, then the multiplicative group of the latter subalgebra relates to the Γ_1-type groups and has order $(p-1)^2$ [10]. Note that the probability of a vector $\mathbf{G}$ selected at random from the set (6) is equal to ≈ 0.5 (i. e., to the probability that the value $\Delta \neq 0$ is a quadratic residue).

It is assumed that, depending on the required level of security, FNAAs of different dimensions m will be used as algebraic support of the developed algorithm. For the case $m \geq 6$, the FNAAs are set with using the next formula for generating the BVMTs for arbitrary even dimensions [25]:

$$\mathbf{e}_i\mathbf{e}_j = \begin{cases} \mathbf{e}_{i+j \bmod m}, & \text{if } i \bmod 2 = 0; \\ \mathbf{e}_{i-j \bmod m}, & \text{if } i \bmod 2 = 1,\ j \bmod 2 = 0; \\ \lambda\mathbf{e}_{i-j \bmod m}, & \text{if } i \bmod 2 = 1,\ j \bmod 2 = 1, \end{cases}$$

where $i, j \in \{0, 1, \ldots, m-1\}$. The latter formula allows to construct BVMTs setting FNAAs with global two-sided unit of the form $\mathbf{E} = (1, 0, 0, \ldots, 0)$. Tables 2 and 3 shows the BVMTs for the cases $m = 6$ and $m = 8$. In framework of this article, we consider the FNAAs set over the ground finite field $GF(p)$ with odd characteristic p (such that $p = 2q + 1$, where p and q are prime numbers). However, the method by [25] can be also used for setting the FNAAs over the finite fields of characteristic two.

The condition for the vector $\mathbf{A} = (a_0, a_1, a_2, a_3, a_4, a_5)$ to be reversible is the following non-equality [25]:

$$\frac{1}{4}\left((a_0 + a_2 + a_4)^2 - \lambda(a_1 + a_3 + a_5)^2\right) \times$$
$$\times \left((a_0 - a_2)^2 + (a_0 - a_4)^2 + (a_2 - a_4)^2 - \right.$$
$$\left. -\lambda(a_1 - a_3)^2 - \lambda(a_1 - a_5)^2 - \lambda(a_3 - a_5)^2\right)^2 \neq 0. \quad (7)$$

Finding a reversible 6-dimensional vector $\mathbf{A}$ can be done by generating at random six of its coordinates and checking the validity of inequality (7). For the cases $m \geq 8$ there are no formulas for the condition of vector reversibility and the following method for generation of reversible vectors can be applied: generate random vectors $\mathbf{B}$ until $\mathbf{B}^{p^2-1} = \mathbf{E}$ or $\mathbf{B}^{(p-1)^2} = \mathbf{E}$.

In the proposed signature algorithm, we use the primes p having the size $|p| = 80$ and $|p| = 128$ bits (the corresponding values of $|q|$ are equal to 79 and 127 bits). We also use two collision-resistant hash functions, the usual type $f(\cdot)$ and the vector type $\mathbf{H}(\cdot)$. The hash function of the latter type take on the values in the FNAA used as algebraic support.

■ ***Table 1.*** Defining non-commutative associative multiplication of 4-dimensional vectors ($\lambda \neq 0$) [10]

$\circ$	$\mathbf{e}_0$	$\mathbf{e}_1$	$\mathbf{e}_2$	$\mathbf{e}_3$
$\mathbf{e}_0$	$\mathbf{e}_0$	0	0	$\mathbf{e}_3$
$\mathbf{e}_1$	0	$\mathbf{e}_1$	$\mathbf{e}_2$	0
$\mathbf{e}_2$	$\mathbf{e}_2$	0	0	$\lambda\mathbf{e}_1$
$\mathbf{e}_3$	0	$\mathbf{e}_3$	$\lambda\mathbf{e}_0$	0

■ ***Table 2.*** Defining non-commutative associative multiplication of the 6-dimensional vectors ($\lambda \neq 0$) [25]

$\circ$	$\mathbf{e}_0$	$\mathbf{e}_1$	$\mathbf{e}_2$	$\mathbf{e}_3$	$\mathbf{e}_4$	$\mathbf{e}_5$
$\mathbf{e}_0$	$\mathbf{e}_0$	$\mathbf{e}_1$	$\mathbf{e}_2$	$\mathbf{e}_3$	$\mathbf{e}_4$	$\mathbf{e}_5$
$\mathbf{e}_1$	$\mathbf{e}_1$	$\lambda\mathbf{e}_0$	$\mathbf{e}_5$	$\lambda\mathbf{e}_4$	$\mathbf{e}_3$	$\lambda\mathbf{e}_2$
$\mathbf{e}_2$	$\mathbf{e}_2$	$\mathbf{e}_3$	$\mathbf{e}_4$	$\mathbf{e}_5$	$\mathbf{e}_0$	$\mathbf{e}_1$
$\mathbf{e}_3$	$\mathbf{e}_3$	$\lambda\mathbf{e}_2$	$\mathbf{e}_1$	$\lambda\mathbf{e}_0$	$\mathbf{e}_5$	$\lambda\mathbf{e}_4$
$\mathbf{e}_4$	$\mathbf{e}_4$	$\mathbf{e}_5$	$\mathbf{e}_0$	$\mathbf{e}_1$	$\mathbf{e}_2$	$\mathbf{e}_3$
$\mathbf{e}_5$	$\mathbf{e}_5$	$\lambda\mathbf{e}_4$	$\mathbf{e}_3$	$\lambda\mathbf{e}_2$	$\mathbf{e}_1$	$\lambda\mathbf{e}_0$

■ ***Table 3.*** Defining the 8-dimensional FNAA ($\lambda \neq 0$)

$\circ$	$\mathbf{e}_0$	$\mathbf{e}_1$	$\mathbf{e}_2$	$\mathbf{e}_3$	$\mathbf{e}_4$	$\mathbf{e}_5$	$\mathbf{e}_6$	$\mathbf{e}_7$
$\mathbf{e}_0$	$\mathbf{e}_0$	$\mathbf{e}_1$	$\mathbf{e}_2$	$\mathbf{e}_3$	$\mathbf{e}_4$	$\mathbf{e}_5$	$\mathbf{e}_6$	$\mathbf{e}_7$
$\mathbf{e}_1$	$\mathbf{e}_1$	$\lambda\mathbf{e}_0$	$\mathbf{e}_7$	$\lambda\mathbf{e}_6$	$\mathbf{e}_5$	$\lambda\mathbf{e}_4$	$\mathbf{e}_3$	$\lambda\mathbf{e}_2$
$\mathbf{e}_2$	$\mathbf{e}_2$	$\mathbf{e}_3$	$\mathbf{e}_4$	$\mathbf{e}_5$	$\mathbf{e}_6$	$\mathbf{e}_7$	$\mathbf{e}_0$	$\mathbf{e}_1$
$\mathbf{e}_3$	$\mathbf{e}_3$	$\lambda\mathbf{e}_2$	$\mathbf{e}_1$	$\lambda\mathbf{e}_0$	$\mathbf{e}_7$	$\lambda\mathbf{e}_6$	$\mathbf{e}_5$	$\lambda\mathbf{e}_4$
$\mathbf{e}_4$	$\mathbf{e}_4$	$\mathbf{e}_5$	$\mathbf{e}_6$	$\mathbf{e}_7$	$\mathbf{e}_0$	$\mathbf{e}_1$	$\mathbf{e}_2$	$\mathbf{e}_3$
$\mathbf{e}_5$	$\mathbf{e}_5$	$\lambda\mathbf{e}_4$	$\mathbf{e}_3$	$\lambda\mathbf{e}_2$	$\mathbf{e}_1$	$\lambda\mathbf{e}_0$	$\mathbf{e}_7$	$\lambda\mathbf{e}_6$
$\mathbf{e}_6$	$\mathbf{e}_6$	$\mathbf{e}_7$	$\mathbf{e}_0$	$\mathbf{e}_1$	$\mathbf{e}_2$	$\mathbf{e}_3$	$\mathbf{e}_4$	$\mathbf{e}_5$
$\mathbf{e}_7$	$\mathbf{e}_7$	$\lambda\mathbf{e}_6$	$\mathbf{e}_5$	$\lambda\mathbf{e}_4$	$\mathbf{e}_3$	$\lambda\mathbf{e}_2$	$\mathbf{e}_1$	$\lambda\mathbf{e}_0$

To compute $\mathbf{H}(M)$ from a document M, other hash function of the usual type $f'(\cdot)$ is used to calculate coordinates of the vector $\mathbf{H} = (h_0, h_1, \ldots, h_{m-1})$. The size of the values of the usual type hash functions is equal to $2|p|$. Calculation of the value of $\mathbf{H}(M)$ is performed as follows:

1. Compute the value $h'_{(0)} = h'_{(0)1}||h'_{(0)2} = f'(M)$ represented as concatenation of two $|p|$-bit values.
2. Calculate the values $h'_{(j)} = h'_{(j)1}||h'_{(j)2} = f'(h'_{(j-1)})$ for $j = 1, \ldots, (m-2)/2$.
3. Set values of the coordinates $h_{2i} = h'_{(i)1}$ and $h_{2i+1} = h'_{(i)2}$, where $i = 0, 1, \ldots, (m-2)/2$.

Setting the hidden group and formation of the public key

A commutative group contained in the FNAA used as algebraic support can serve as a hidden group in the proposed method and algorithm. However, we will consider the case of setting the hidden group with two-dimensional cyclicity, i. e., the group containing the minimum generator system <**G**, **J**> including two vectors of the order $p-1$ (in the case $m = 4$, this is the Γ_1-type group).

Algorithm 1: Setting a hidden group possessing two-dimensional cyclicity in the case $m \geq 4$.

1. Generate a random reversible vector $\mathbf{G}$ of order $p-1$.
2. If $\mathbf{G}$ is a scalar vector, then go to step 1.
3. Generate at random an integer k ($k < p-1$) and a primitive element β in $GF(p)$.
4. Compute the vector $\mathbf{J} = \beta\mathbf{G}^k$.
5. Output the pair $\mathbf{J}$ and $\mathbf{G}$ as a basis <**G**, **J**> of the hidden group $\Gamma_{<\mathbf{G},\mathbf{J}>}$.

In the case $m = 4$ the efficiency of Algorithm 1 can be estimated as follows. Taking into account that the number of the Γ_1-type groups in the 4-dimensional FNAA used as algebraic support is equal to $p(p+1)/2$ [10], one can estimate that the number of the vectors contained in all groups of the Γ_1-type is equal to $\approx p^4/2$. Therefore, a random vector $\mathbf{G}$ is a non-scalar vector and is contained in a group of the Γ_1-type with probability ≈ 0.5. For the case of prime $p = 2q + 1$ (where q is also a prime), one can easily show that a fixed Γ_1 group contains $\approx q^2$ of vectors of the order q and $\approx 3q^2$ of vectors of the order $p-1$, i. e., a random vector from the fixed Γ_1 group has order $p-1$ with the probability $\approx 3/4$. Thus, a random vector $\mathbf{G}$ passes the steps 1 and 2 with the probability $\approx 3/8$ and generation of the required vector $\mathbf{G}$ requires on the average performing steps 1 and 2 about 8/3 times.

Taking into account that step 3 is performed on the average two times, one can conclude that generation of a random basis <**G**, **J**> of the hidden group $\Gamma_{<\mathbf{G},\mathbf{J}>}$ requires on the average performing Algorithm 1 less than 3 times.

Estimate of the efficiency of Algorithm 1 for the cases $m > 4$ requires using the detailed information about decomposition of the corresponding FNAAs into commutative subalgebras. Because of the lack of such information we have experimentally found the average number ψ of performing Algorithm 1 for generating the base <**G**, **J**> of the hidden group with two-dimensional cyclicity. For every of the cases $m = 6$, $m = 8$, and $m = 10$, we have get $\psi < 5$ for different values of structural constant λ and of prime p. Thus, Algorithm 1 has acceptable efficiency, since it is intended to be used only at the stage of forming the public key.

Algorithm 2: Alternative procedure for setting a hidden group of the Γ_1 *type in the case* $m = 4$.

1. Generate a random reversible vector $\mathbf{G} = (g_0, g_1, g_2, g_3)$ such that $g_2 \neq 0$ and $g_3 \neq 0$ and compute the value of $\Delta = (g_0 - g_1)^2 + 4\lambda g_2 g_3$.
2. If $\Delta = 0$ or Δ is a quadratic non-residue in $GF(p)$, then go to step 1.
3. Generate two random integers $d \neq 0$ and $k \neq 0$ and, using formula (6), compute the vector $\mathbf{T}$.
4. If the order of $\mathbf{T}$ is not equal to $p-1$, then go to step 3.
5. Generate a random integer k ($0 < k < p-1$) and a random primitive element β in $GF(p)$.
6. Compute the vector $\mathbf{J} = \beta\mathbf{T}^k$.
7. Output the pair $\mathbf{J}$ and $\mathbf{G} = \mathbf{V}$ as a basis <**G**, **J**> of the hidden group $\Gamma_{<\mathbf{G},\mathbf{J}>}$.

The probability that a random vector $\mathbf{G}$ defines the value of $\Delta \neq 0$ that is a quadratic residue is equal to ≈ 0.5, therefore the steps 1 and 2 are performed on the average two times. Like in the case of Algorithm 1, one can easily show the probability that a random vector $\mathbf{T}$ has order $p-1$ is equal to $\approx 3/4$. Therefore the steps 3 and 4 are performed on the average $\approx 4/3$ times. Like in the case of Algorithm 1, step 5 is performed on the average two times. Thus, generation of a random basis <**G**, **J**> requires performing Algorithm 2 approximately two times.

The vectors from the hidden group are used to calculate two parts of the public key. The elements of the first part are included in the first verification equation, and the elements of the second part are included in the second verification equation. The vector hash function and two similar verification equations, performed on the same signature value (e, $\mathbf{S}$), are used to prevent signature-fitting attacks.

The next procedure for generating the public key has been developed:

1. Generate at random a hidden group $\Gamma_{<\mathbf{G},\mathbf{J}>}$.
2. Generate at random reversible vectors $\mathbf{A}$, $\mathbf{B}$, $\mathbf{F}$, and $\mathbf{P}$ such that $\mathbf{AG} \neq \mathbf{GA}$, $\mathbf{AB} \neq \mathbf{BA}$, $\mathbf{AF} \neq \mathbf{FA}$, $\mathbf{AP} \neq \mathbf{PA}$, $\mathbf{BG} \neq \mathbf{GB}$, $\mathbf{BF} \neq \mathbf{FB}$, $\mathbf{BP} \neq \mathbf{PB}$, $\mathbf{FG} \neq \mathbf{GF}$, $\mathbf{FP} \neq \mathbf{PF}$, and $\mathbf{PG} \neq \mathbf{GP}$ (for random vectors $\mathbf{A}$, $\mathbf{B}$, $\mathbf{F}$, and $\mathbf{P}$ these ten inequalities holds true with a high probability).
3. Calculate the public key elements $\mathbf{Y}_1$, $\mathbf{U}_1$, $\mathbf{Y}_2$, and $\mathbf{U}_2$:

$$\mathbf{Y}_1 = \mathbf{AGA}^{-1},\ \mathbf{U}_1 = \mathbf{BJB}^{-1},\ \mathbf{Y}_2 = \mathbf{FGF}^{-1},\ \text{and } \mathbf{U}_2 = \mathbf{PJP}^{-1}. \quad (8)$$

4. Generate at random a reversible vector $\mathbf{D}$ and hidden group elements $\mathbf{G}_1$, $\mathbf{G}_2$, $\mathbf{J}_1$, and $\mathbf{J}_2$ such that $\mathbf{DG} \neq \mathbf{GD}$, $\mathbf{DA} \neq \mathbf{AD}$, $\mathbf{DB} \neq \mathbf{BD}$, $\mathbf{DF} \neq \mathbf{FD}$, $\mathbf{DP} \neq \mathbf{PD}$.
5. Calculate the public key elements $\mathbf{Z}_1$, $\mathbf{W}_1$, $\mathbf{Z}_2$, and $\mathbf{W}_2$:

$$\mathbf{Z}_1 = \mathbf{AG}_1\mathbf{B}^{-1},\ \mathbf{W}_1 = \mathbf{BJ}_1\mathbf{D}^{-1},\ \mathbf{Z}_2 = \mathbf{FG}_2\mathbf{P}^{-1},\ \text{and } \mathbf{W}_2 = \mathbf{PJ}_2\mathbf{D}^{-1}. \quad (9)$$

Thus, the secret key (private key) represents the next set of vectors $\{\mathbf{G}, \mathbf{J}, \mathbf{G}_1, \mathbf{G}_2, \mathbf{J}_1, \mathbf{J}_2, \mathbf{A}, \mathbf{B}, \mathbf{F}, \mathbf{P}, \mathbf{D}\}$ with the total size of $110m$ ($176m$) bytes for 80-bit (128-bit) prime p. The public key represents the set of vectors $\{\mathbf{Y}_1, \mathbf{Z}_1, \mathbf{U}_1, \mathbf{W}_1, \mathbf{Y}_2, \mathbf{Z}_2, \mathbf{U}_2, \mathbf{W}_2\}$ with the total size of $80m$ ($128m$) bytes for 80-bit (128-bit) prime p.

Note all elements of the public key are calculated as masked elements $\mathbf{G}$, $\mathbf{J}$, $\mathbf{G}_1$, $\mathbf{G}_2$, $\mathbf{J}_1$, $\mathbf{J}_2$ of the hidden group, besides, elements $\mathbf{Y}_1$ and $\mathbf{Y}_2$ ($\mathbf{U}_1$ and $\mathbf{U}_2$) are calculated from the same hidden group element $\mathbf{G}$ ($\mathbf{J}$) using different masking factors $\mathbf{A}$ and $\mathbf{F}$ ($\mathbf{B}$ and $\mathbf{P}$). Such connection of the public key elements with the hidden group underlies the correctness of the developed signature algorithm.

Signature generation procedure

A digital signature (e, $\mathbf{S}$) to an electronic document M is calculated using the next randomized procedure:

1. Using the vector hash function $\mathbf{H}(\cdot)$ and concatenating the public key elements $\mathbf{Y}_1$ and $\mathbf{Y}_2$ to the document M, calculate the hash values $\mathbf{H}_1$ and $\mathbf{H}_2$:

$$\mathbf{H}_1 = \mathbf{H}(\mathbf{Y}_1||M) \text{ and } \mathbf{H}_2 = \mathbf{H}(M||\mathbf{Y}_2). \quad (10)$$

2. Generate at random a reversible vector $\mathbf{V}$ and two natural numbers k ($k < p - 1$) and t ($t < p - 1$) and calculate the vectors $\mathbf{R}_1$ and $\mathbf{R}_2$:

$$\mathbf{R}_1 = \mathbf{AG}^k\mathbf{J}^t\mathbf{G}_1\mathbf{J}_1\mathbf{VH}_1; \quad (11)$$

$$\mathbf{R}_2 = \mathbf{FG}^k\mathbf{J}^t\mathbf{G}_2\mathbf{J}_2\mathbf{VH}_2. \quad (12)$$

3. Using a specified $2|p|$-bit hash function f, generate the first signature element e as a hash-function value calculated from the document M to which the vectors $\mathbf{R}_1$ and $\mathbf{R}_2$ are concatenated: $e = e_1||e_2 = f(M||\mathbf{R}_1||\mathbf{R}_2)$, where the hash-value e is considered as concatenation of two $|p|$-bit natural numbers e_1 and e_2.
4. Calculate the integers s_1 and s_2:

$$s_1 = k - e_1 \bmod p - 1; \quad (13)$$

$$s_2 = t - e_2 \bmod p - 1. \quad (14)$$

5. Calculate the vector $\mathbf{S}$ as the second signature element:

$$\mathbf{S} = \mathbf{DG}^{s_1}\mathbf{J}^{s_2}\mathbf{V}. \quad (15)$$

6. Output the pair (e, $\mathbf{S}$) as the signature to the document M.

Note that the accumulation of many unique values of the signature element $\mathbf{S}$ does not make it possible to form a system of power vector equations from which the attacker would be able to calculate the secret vectors $\mathbf{D}$, $\mathbf{G}$, and $\mathbf{J}$. This is due to the fact that each unique signature (e, $\mathbf{S}$) defines an equation of the form (15) with a unique unknown vector $\mathbf{V}$ that has a random value in the FNAA used as an algebraic support (see step 2 of the signature generation procedure).

Without taking into account the computational difficulty of finding the hash values e, $\mathbf{H}_1$, and $\mathbf{H}_2$ (which depend on the size of document M), the computational difficulty of the rest of the signature generation procedure can be approximately evaluated as four exponentiation operations: i) two exponentiations are performed to calculate both of the vectors $\mathbf{R}_1$ and $\mathbf{R}_2$ [note that in (11) and (12) the same two exponentiations are performed] and ii) two exponentiations are performed to calculate the vector $\mathbf{S}$. Four exponentiations in the FNAA used as algebraic support take about $6m^2|p|$ multiplications in $GF(p)$ for $m \geq 6$ and $48|p| \approx 6150$ multiplications in $GF(p)$ for $m = 4$.

The signature size is equal to $(m + 2)|p|$ bits, for example, to $10(m + 2)$ bytes for the case of 80-bit prime p and $16(m + 2)$ bytes for 128-bit prime p.

Thus, the proposed signature algorithm has significantly lower signature size than many known post-quantum signature algorithms [15, 20].

Signature verification procedure

To verify a signature (e, $\mathbf{S}$) assigned to the document M, one is to use the public key of the signer $\{\mathbf{Y}_1, \mathbf{Z}_1, \mathbf{U}_1, \mathbf{W}_1, \mathbf{Y}_2, \mathbf{Z}_2, \mathbf{U}_2, \mathbf{W}_2\}$ and the following procedure:

1. Using the vector hash function $\mathbf{H}(\cdot)$ and formulas (10), calculate the values $\mathbf{H}_1$ and $\mathbf{H}_2$ from the document M to which the public key elements $\mathbf{Y}_1$ and $\mathbf{Y}_2$ are attached.
2. Calculate the vectors $\mathbf{R}'_1$ and $\mathbf{R}'_2$ by the next two formulas:

$$\mathbf{R}'_1 = \mathbf{Y}_1^{e_1}\mathbf{Z}_1\mathbf{U}_1^{e_2}\mathbf{W}_1\mathbf{S}\mathbf{H}_1; \tag{16}$$

$$\mathbf{R}'_2 = \mathbf{Y}_2^{e_1}\mathbf{Z}_2\mathbf{U}_2^{e_2}\mathbf{W}_2\mathbf{S}\mathbf{H}_2. \tag{17}$$

3. Calculate the hash-value e' from the document to which the vectors $\mathbf{R}'_1$ and $\mathbf{R}'_2$ are attached: $e' = f(M||\mathbf{R}'_1||\mathbf{R}'_2)$.
4. If $e' = e$, then the signature is genuine, else the signature is false.

The computational difficulty of the signature verification procedure is approximately equal to that of the signature generation algorithm.

Correctness of the signature scheme is proven as follows.

Signature scheme correctness proof.

Compute the vectors $\mathbf{R}'_1$ and $\mathbf{R}'_2$:

$$\begin{aligned}\mathbf{R}'_1 &= \mathbf{Y}_1^{e_1}\mathbf{Z}_1\mathbf{U}_1^{e_2}\mathbf{W}_1\mathbf{S}\mathbf{H}_1 = \\ &= \mathbf{A}\mathbf{G}^{e_1}\mathbf{A}^{-1}\mathbf{A}\mathbf{G}_1\mathbf{B}^{-1}\mathbf{B}\mathbf{J}^{e_2}\mathbf{B}^{-1}\mathbf{B}\mathbf{J}_1\mathbf{D}^{-1}\mathbf{S}\mathbf{H}_1 = \\ &= \mathbf{A}\mathbf{G}^{e_1}\mathbf{G}_1\mathbf{J}^{e_2}\mathbf{J}_1\mathbf{D}^{-1}\mathbf{D}\mathbf{G}^{s_1}\mathbf{J}^{s_2}\mathbf{V}\mathbf{H}_1 = \\ &= \mathbf{A}\mathbf{G}^{e_1}\mathbf{G}_1\mathbf{J}^{e_2}\mathbf{J}_1\mathbf{G}^{k-e_1}\mathbf{J}^{t-e_2}\mathbf{V}\mathbf{H}_1 = \\ &= \mathbf{A}\mathbf{G}^{k}\mathbf{J}^{t}\mathbf{G}_1\mathbf{J}_1\mathbf{V}\mathbf{H}_1 = \mathbf{R}_1;\end{aligned}$$

$$\begin{aligned}\mathbf{R}'_2 &= \mathbf{Y}_2^{e_1}\mathbf{Z}_2\mathbf{U}_2^{e_2}\mathbf{W}_2\mathbf{S}\mathbf{H}_2 = \\ &= \mathbf{F}\mathbf{G}^{e_1}\mathbf{F}^{-1}\mathbf{F}\mathbf{G}_2\mathbf{P}^{-1}\mathbf{P}\mathbf{J}^{e_2}\mathbf{P}^{-1}\mathbf{P}\mathbf{J}_2\mathbf{D}^{-1}\mathbf{S}\mathbf{H}_2 = \\ &= \mathbf{F}\mathbf{G}^{e_1}\mathbf{G}_2\mathbf{J}^{e_2}\mathbf{J}_2\mathbf{D}^{-1}\mathbf{D}\mathbf{G}^{s_1}\mathbf{J}^{s_2}\mathbf{V}\mathbf{H}_2 = \\ &= \mathbf{F}\mathbf{G}^{e_1}\mathbf{G}_2\mathbf{J}^{e_2}\mathbf{J}_2\mathbf{G}^{k-e_1}\mathbf{J}^{t-e_2}\mathbf{V}\mathbf{H}_2 = \\ &= \mathbf{F}\mathbf{G}^{k}\mathbf{J}^{t}\mathbf{G}_2\mathbf{J}_2\mathbf{V}\mathbf{H}_2 = \mathbf{R}_2.\end{aligned}$$

Then compute the hash value $e' = f(M||\mathbf{R}'_1||\mathbf{R}'_2)$:

$$\begin{aligned}&\{\mathbf{R}'_1 = \mathbf{R}_1;\ \mathbf{R}'_2 = \mathbf{R}_2\} \Rightarrow \\ &\Rightarrow f\left(M\|\mathbf{R}'_1\|\mathbf{R}'_2\right) = f\left(M\|\mathbf{R}_1\|\mathbf{R}_2\right) \Rightarrow \\ &\Rightarrow\ e' = e.\end{aligned}$$

The latter equality determines validity of the verified signature.

Forging-signature attacks

Let's consider two attacks related to the formation of a genuine signature without knowing the secret key. In the first attack the forger selects arbitrary values of $\mathbf{S}$ and of e_1 and e_2 and, using the signature verification equations (16) and (17) computes the vectors $\mathbf{R}'_1$ and $\mathbf{R}'_2$ (note the attacker can also fix the values of e_1 and e_2 and modify only the value of $\mathbf{S}$). Then he computes the hash value $e' = e'_1||e'_2 = f(M||\mathbf{R}'_1||\mathbf{R}'_2)$ until $e'_1 = e_1$ and $e'_2 = e_2$. Probability that both of the latter two equalities hold true is equal to $\approx p^{-2}$, therefore, the computational difficulty of such attack can be estimated as $O(p^2)$ or as $O(2^{2|p|})$ multiplications in FNAA used as algebraic support. For the cases of 80-bit and 128-bit prime numbers p such attack is computationally infeasible.

One can propose some versions of the first attack, in which the vectors $\mathbf{R}'_1$ and $\mathbf{R}'_2$ are selected at random. Then the value $e' = f(M||\mathbf{R}'_1||\mathbf{R}'_2)$ and signatures $\mathbf{S}_1$ and $\mathbf{S}_2$ are computed from (16) and (17), correspondingly, until the equality $\mathbf{S}_1 = \mathbf{S}_2 = \mathbf{S}$ holds true. The latter equality take place with probability $\approx p^{-m}$, causing the computational difficulty of the attack equal to $O(p^m)$, where $m \geq 4$.

In the second attack the forger uses some known genuine signature (e, $\mathbf{S}$) assigned to the document M (thus, he can compute $\mathbf{H}_1 = \mathbf{H}(\mathbf{Y}_1||M)$ and $\mathbf{H}_2 = \mathbf{H}(M||\mathbf{Y}_2)$) and attempts to calculate a valid signature (e'', $\mathbf{S}''$) assigned to the document M''. He calculates the vectors $\mathbf{H}''_1 = \mathbf{H}(\mathbf{Y}_1||M'')$ and $\mathbf{H}''_2 = \mathbf{H}(M''||\mathbf{Y}_2)$. Then, using the formulas (16) and (17) and value of $e = e_1||e_2$ he calculates the vectors $\mathbf{R}'_1$ and $\mathbf{R}'_2$ and the value $e'' = e''_1||e''_2 = f(M, \mathbf{R}'_1||\mathbf{R}'_2)$. At the next step of the attack the forger tries to compute the vector $\mathbf{S}''$ satisfying the both verification equations. From (16) he has

$$\mathbf{R}'_1 = \mathbf{Y}_1^{e_1}\mathbf{Z}_1\mathbf{U}_1^{e_2}\mathbf{W}_1\mathbf{S}\mathbf{H}_1 = \mathbf{Y}_1^{e''_1}\mathbf{Z}_1\mathbf{U}_1^{e''_2}\mathbf{W}_1\mathbf{S}''\mathbf{H}''_1. \tag{18}$$

The equality (18) gives

$$\mathbf{S}'' = \mathbf{D}\mathbf{G}^{e_1-e''_1}\mathbf{J}^{e_2-e''_2}\mathbf{D}^{-1}\mathbf{S}\mathbf{H}_1\mathbf{H}_1''^{-1}. \tag{19}$$

In a similar way from (17) the forger gets

$$\mathbf{S}'' = \mathbf{D}\mathbf{G}^{e_1-e''_1}\mathbf{J}^{e_2-e''_2}\mathbf{D}^{-1}\mathbf{S}\mathbf{H}_2\mathbf{H}_2''^{-1}. \tag{20}$$

The second attack is successful, if the values of $\mathbf{S}''$ calculated from (19) and (20) are equal, i. e., if the following equality is true:

$$\mathbf{H}_2''^{-1}\mathbf{H}''_1 = \mathbf{H}_2^{-1}\mathbf{H}_1.$$

The latter takes place with a probability p^{-m} and determines the computational complexity of the attack equals to p^m, when a limited number (for example, less than 2^{40}) of valid signatures are available for the attacker, or to $p^{m/2}$, when an extremely large number ($\approx p^{m/2}$) of valid signatures is available for the attacker (consider, for example, a model of the oracle that signs documents generated by the attacker). In the latter case, the attacker selects $p^{m/2}$ different documents $M''^{(i)}$ ($i = 1, 2, ..., p^{m/2}$) and computes for every document $M''^{(i)}$ the value of $(\mathbf{H}_2''^{(i)})^{-1}\mathbf{H}_1''^{(i)}$. He also computes the values of $(\mathbf{H}_2^{(j)})^{-1}\mathbf{H}_1^{(j)}$($j = 1, 2, ..., p^{m/2}$) connected with the corresponding documents $M^{(j)}$ signed with the signatures ($e^{(j)}$, $\mathbf{S}^{(j)}$).

In correspondence with the birthday paradox, with probability equal to ≈ 0.5 the first set contains a value $(\mathbf{H}_2''^{(k)})^{-1}\mathbf{H}_1''^{(k)}$ such that $(\mathbf{H}_2''^{(k)})^{-1}\mathbf{H}_1''^{(k)} = (\mathbf{H}_2^{(t)})^{-1}\mathbf{H}_1^{(t)}$ for some natural number $t \le p^{m/2}$. The attacker orders all values from the first and second sets (performing $\approx m|p|p^{m/2}/2$ operations of comparison) and finds the values $(\mathbf{H}_2''^{(k)})^{-1}\mathbf{H}_1''^{(k)}$ and $(\mathbf{H}_2^{(t)})^{-1}\mathbf{H}_1^{(t)}$. Then, using formula (19) or (20) and the signature $\mathbf{S}^{(t)}$, he calculates the valid signature $\mathbf{S}''^{(k)}$ to the document $M''^{(k)}$ connected with the product $(\mathbf{H}_2''^{(k)})^{-1}\mathbf{H}_1''^{(k)}$. Calculation of the said two sets takes $4p^{m/2}$ operations of computing the vector hash function and $2p^{m/2}$ multiplications in the FNAA used as algebraic support. Computational complexity of the birthday-paradox-based attack can be roughly estimated as $O(p^{m/2})$.

Thus, the second forging attack is also impractical for the cases of 80-bit and 128-bit prime numbers p.

Attacks connected with calculation of the private key

The formulas (8) and (9) define connection between elements of the private and public keys. Namely, from (8) and (9) we have the following system of eight quadratic vector equations with eleven unknown vectors $\mathbf{G}$, $\mathbf{J}$, $\mathbf{G}_1$, $\mathbf{G}_2$, $\mathbf{J}_1$, $\mathbf{J}_2$, $\mathbf{A}$, $\mathbf{B}$, $\mathbf{F}$, $\mathbf{P}$, and $\mathbf{D}$:

$$\begin{cases} \mathbf{Y}_1\mathbf{A} = \mathbf{AG}; & \mathbf{Z}_1\mathbf{B} = \mathbf{AG}_1; \\ \mathbf{U}_1\mathbf{B} = \mathbf{BJ}; & \mathbf{W}_1\mathbf{D} = \mathbf{BJ}_1; \\ \mathbf{Y}_2\mathbf{F} = \mathbf{FG}; & \mathbf{Z}_2\mathbf{P} = \mathbf{FG}_2; \\ \mathbf{U}_2\mathbf{P} = \mathbf{PJ}; & \mathbf{W}_2\mathbf{D} = \mathbf{PJ}_2. \end{cases} \qquad (21)$$

In this system, the vectors $\mathbf{G}$, $\mathbf{J}$, $\mathbf{G}_1$, $\mathbf{G}_2$, $\mathbf{J}_1$, and $\mathbf{J}_2$ are elements of the hidden commutative group. This fact is to be taken into account when representing the system (21) in the form of scalar equations in $GF(p)$. Such a reduction of the system (21) is performed using the BVMT setting the FNAA used as algebraic support.

Consider the case of 4-dimensional FNAA for which detailed decomposition into the set of commutative subalgebras is known [10]. One can consider the coordinates of one vector (for example, $\mathbf{G}$) from the hidden group as four scalar unknowns and then represent the coordinates of every of the other unknown vectors from the hidden group (for example, $\mathbf{J}$, $\mathbf{G}_1$, $\mathbf{G}_2$, $\mathbf{J}_1$, and $\mathbf{J}_2$) as four scalar values depending on two unique scalar values d and k in accordance with the formula (6). Such representation leads to transformation of quadratic equations into cubic ones, however, this does not increase the computational difficulty of solving the system of power equations, since the best known methods (based on the F4 [26] and F5 [27] algorithms) for finding a solution of such system have about the same efficiency for quadratic and cubic equations

The system (21) reduces to the system of $8m$ power equations (quadratic and cubic). In the case $m = 4$ we have the system of 32 equations with 34 unknowns in $GF(p)$. For the FNAAs of the dimensions $m \ge 6$ their decomposition into commutative subalgebras has not yet been investigated in detail, however, we have get preliminary results for the cases $m = 6$ and $m = 8$, which show these FNAAs contain commutative subalgebras every one of which can be described as a vector subspace (that is set by coordinates of a vector contained in the subalgebra) of the dimension $m/2$.

The latter means that, like in the case $m = 4$, in the cases $m = 6$ and $m = 8$ the system (21) reduces to the system of $\mu' = 8m$ scalar equations with $\eta = 6m + 5m/2$ scalar unknowns. Since the number of unknowns is larger than the number of equations, one can suppose the considered system of scalar power equations has many solutions. The latter fact is easily confirmed by considering system (21) at the level of the FNAA. Indeed, let system (21) has a solution $\mathbf{G}_0$, $\mathbf{J}_0$, $\mathbf{G}_{10}$, $\mathbf{G}_{20}$, $\mathbf{J}_{10}$, $\mathbf{J}_{20}$, $\mathbf{A}_0$, $\mathbf{B}_0$, $\mathbf{F}_0$, $\mathbf{P}_0$, and $\mathbf{D}_0$. Then for every reversible vector $\mathbf{X}$ (the number of reversible vectors is approximately equal to p^m) one has the following unique solution

$$\mathbf{G}_\mathbf{X} = \mathbf{X}^{-1}\mathbf{G}_0\mathbf{X};\ \mathbf{J}_\mathbf{X} = \mathbf{X}^{-1}\mathbf{J}_0\mathbf{X};$$
$$\mathbf{G}_{1\mathbf{X}} = \mathbf{X}^{-1}\mathbf{G}_{10}\mathbf{X};\ \mathbf{G}_{2\mathbf{X}} = \mathbf{X}^{-1}\mathbf{G}_{20}\mathbf{X};$$
$$\mathbf{J}_{1\mathbf{X}} = \mathbf{X}^{-1}\mathbf{J}_{10}\mathbf{X};\ \mathbf{J}_{2\mathbf{X}} = \mathbf{X}^{-1}\mathbf{J}_{20}\mathbf{X};$$
$$\mathbf{A}_\mathbf{X} = \mathbf{A}_0\mathbf{X};\ \mathbf{B}_\mathbf{X} = \mathbf{B}_0\mathbf{X};$$
$$\mathbf{F}_\mathbf{X} = \mathbf{F}_0\mathbf{X};\ \mathbf{P}_\mathbf{X} = \mathbf{P}_0\mathbf{X};\ \mathbf{D}_\mathbf{X} = \mathbf{D}_0\mathbf{X}.$$

At the level of a system of scalar equations, the presence of $\approx p^m$ solutions means the presence of dependent scalar equations. Namely, we can assume that the number (μ) of independent scalar equations is equal to the number of unknowns minus m. Thus, we have $\mu = \eta - m = 6m + 5m/2 - m$, i. e.

$$\mu = 6m + 3m/2. \qquad (22)$$

Like in the case of multivariate public-key algorithms, the attack based on solving a system of many power equations with many unknowns can be called direct attack. Computational difficulty of solving such systems depends mainly on the number of equations (μ) and number of unknowns (η). In our case we have $\mu < \eta$, therefore, to evaluate security W of the introduced signature scheme to the direct attack, one can take the number of equations μ equal to the number of the unknowns (for example, the values of $\eta - \mu$ unknowns are predetermined) and use the minimum number of equations $\mu_{\min}$ to get a given level of security, which has been calculated in [28], taking into account the best known algorithms for solving a system of power equations in the field $GF(n)$.The results of [28] are presented in Table 4. For the developed algorithm we have system of power equations in $GF(p)$, where $p >> 2^8$, therefore, for a rough estimate of the required dimension of the used FNAAs one can take the values of $\mu_{\min}$ from Table 4, which relate to the case $n = 2^8$.

Taking into account formula (22), for the case $m = 4$, $m = 6$, and $m = 10$ one gets $\mu = 30$, $\mu = 45$, and $\mu = 75$, correspondingly. Thus, for the said dimensions we have 80-bit, 128-bit, and 192-bit security of the introduced algorithm against the direct attack. To provide 256-bit security the introduced signature algorithms should be implemented on the FNAA of the dimension $m \geq 14$.

In the developed signature algorithm the signature element **S** is computed by formula (15), where the vector **V** is selected at random from the multiplicative group of the FNAA used as algebraic support. One can represent (15) in the form $\mathbf{V} = \mathbf{D}^{-1}\mathbf{G}^{-s_1}\mathbf{J}^{-s_2}\mathbf{S}$. The value of **V** is unique for every valid signature (e, **S**), therefore, arbitrary fixed triple of values of the private-key elements **D**, **G**, and **J** can be connected with every known valid signature. This means that no information about the values of **D**, **G**, and **J** can be obtained from a large set of valid signatures, until two different signatures are calculated using the same value of **V**. However, probability of the latter event is negligibly small ($\leq p^{-2}$, if the number of available signatures is equal or less $p^{m/2-1}$). In addition, it is not obvious how to establish signatures connected with the same value of **V**. This is due to the fact that the value of **S** depends not only on **V**, but also on the values of s_1 and s_2 that change from one signature to another.

The fundamental role of using a random value of **V** is easily seen when compared with the case of calculating the signature element **S** by the formula $\mathbf{S} = \mathbf{B}^{-1}\mathbf{G}^{n}\mathbf{J}^{r}\mathbf{C}^{-1}$ in [13], where natural numbers n and r are unique for every signature. The vector $\mathbf{G}^{n}\mathbf{J}^{r}$ is contained in the hidden group. For the case of using the 4-dimentional FNAA used as algebraic support, the unknown coordinates of the vector $\mathbf{G}' = (g'_0, g'_1, g'_2, g'_3) = \mathbf{G}^{n'}\mathbf{J}^{r'}$, where n' and r' relates to some given valid signature (e', **S**$'$), can be used to describe the vector $\mathbf{G}^{n}\mathbf{J}^{r}$ with two unknown scalar values d and k [see formula (6)]. Suppose we have five different valid signatures (e', **S**$'$), (e_1, $\mathbf{S}_1$), (e_2, $\mathbf{S}_2$), (e_3, $\mathbf{S}_3$), and (e_4, $\mathbf{S}_4$). Then the next system of five vector equations can be written:

$$\begin{cases} \mathbf{BS}'\mathbf{C} = \mathbf{G}'; \\ \mathbf{BS}_1\mathbf{C} = \mathbf{G}^{n_1}\mathbf{J}^{r_1}; \\ \mathbf{BS}_2\mathbf{C} = \mathbf{G}^{n_2}\mathbf{J}^{r_2}; \\ \mathbf{BS}_3\mathbf{C} = \mathbf{G}^{n_3}\mathbf{J}^{r_3}; \\ \mathbf{BS}_4\mathbf{C} = \mathbf{G}^{n_4}\mathbf{J}^{r_4}. \end{cases}$$

This system of the vector equations can be reduced to the system of 20 scalar power (quadratic and cubic) equations with 20 scalar unknowns (coordinates of the unknown vectors **B**, **C**, and **G**$'$ and four pairs of unknows (d_1, k_1), (d_2, k_2), (d_3, k_3), and (d_4, k_4)). Solving the latter system one gets the values of the private-key elements. Taking into account the required minimum number of equations (see Table 4) one can recommend to implement algorithm from [13] on FNAAs of the dimensions $m = 10$, $m = 16$, and $m \geq 20$ for the cases of 2^{128}, 2^{192}, and 2^{256} security levels, correspondingly.

■ ***Table 4.*** The value of $\mu_{\min}$ providing a given level of security against the direct attack [28]

$\log_2 W$	$\mu_{\min}$ at	
	$n = 2^4$	$n = 2^8$
80	30	26
100	39	33
128	51	43
192	80	110
256	68	93

Discussion and conclusion

The article proposes a new method for developing signature algorithms with a hidden group, which are based on computational difficulty of solving large systems of quadratic multivariate equations, and introduces a new post-quantum signature algorithm. In the latter, the fitting parameter **S** of the signature (e, **S**) is calculated by the formula (15) in which a random vector **V** is used as a multiplier. Due to the presence of a random multiplier, the accumulation of a large number of different signatures cannot be used to obtain additional equations, which made it possible to reduce the computational complexity of calculating the elements of the secret key, when using the known elements of the public key and formulas (8) and (9). Thus, the proposed method for developing algebraic signature algorithms with a hidden group eliminates the disadvantage (mentioned in the Introduction) of the algebraic algorithms [13, 14] that use a verification equation with multiple occurrences of the signature element **S**.

The introduced method and the post-quantum signature algorithm represent an attractive alternative for using it as a prototype of a post-quantum signature standard, due to comparatively small sizes of signature, public and private keys.

Table 5 presents a rough comparison of the introduced algorithm with the multivariate signature algorithms selected as finalists of the NIST world competition on the development of the post-quantum public-key algorithms [20, 21].

In the proposed signature scheme we have specified using the FNAAs set over the ground finite field $GF(p)$ with characteristic $p = 2q + 1$, where q is a prime. However, one can use the primes p of an arbitrary form (for example, $p = 2^{80} + c_1$ and $p = 2^{128} + c_2$, where c_1 and c_2 are some specified natural numbers having a small size), since the latter does not influence the security level.

From a practical point of view, it seems very interesting to implement the proposed algorithm on FNAAs defined over finite fields of characteristic two, for example, over $GF(2^{80})$ and $GF(2^{128})$. This will reduce the hardware implementation cost and improve performance.

The performed evaluation of the security of the proposed algorithm is rather preliminary. A more detailed analysis of the features of the emerging system of power equations is required, which could potentially lead to finding special solution methods and to refinement of the security evaluation. It should also be noted that in the framework of future research, attention should be also paid to studying the detailed decomposition of the FNAAs into a set of commutative subalgebras for the cases of dimension $m \geq 6$.

■ ***Table 5.*** Comparison with two finalists of the NIST competition

Signature algorithm	Signature size, bytes	Key size, bytes		W
		Public	Private	
3 versions of Rainbow [20]	66	158 000	101 000	2^{128}
	164	861 000	611 000	2^{192}
	204	1 885 000	1 375 000	2^{256}
3 versions of GeMSS [21]	29	358 000	16	2^{128}
	47	1 294 000	24	2^{192}
	64	3 223 000	32	2^{256}
Proposed $m = 8$ $\|p\| = 80$	100	640	880	2^{128}
Proposed $m = 10$ $\|p\| = 128$	192	1280	1760	2^{192}
Proposed $m = 14$ $\|p\| = 128$	256	1792	2464	2^{256}

Acknowledgement

The authors sincerely thank anonymous Referee for valuable remarks and comments.

References

1. *Call for Additional Digital Signature Schemes for the Post-Quantum Cryptography Standardization Process*. September 6, 2022. 99 pp. Available at: https://csrc.nist.gov/csrc/media/Projects/pqc-dig-sig/documents/call-for-proposals-dig-sig-sept-2022.pdf (accessed 01 March 2023).
2. Ikematsu Y., Nakamura S., Takagi T. Recent progress in the security evaluation of multivariate public-key cryptography. *IET Information Security*, 2022, pp. 1–17. doi:10.1049/ise2.12092
3. Shor P. W. Polynomial-time algorithms for prime factorization and discrete logarithms on quantum computer. *SIAM Journal of Computing*, 1997, vol. 26, pp. 1484–1509.
4. Ekert A., Jozsa R. Quantum computation and Shor's factoring algorithm. *Reviews of Modern Physic*, 1996, vol. 68, pp. 733–752.
5. Alamelou Q., Blazy O., Cauchie S., Gaborit Ph. A code-based group signature scheme. *Designs, Codes and Cryptography*, 2017, vol. 82, no. 1–2, pp. 469–493. doi:10.1007/s10623-016-0276-6
6. Kosolapov Y. V., Turchenko O. Y. On the construction of a semantically secure modification of the McEliece cryptosystem. *Prikl. Diskr. Mat.*, 2019, no. 45, pp. 33–43. doi:10.17223/20710410/45/4
7. Dahmen E., Okeya K., Takagi T., Vuillaume C. *Digital signatures out of Second-Preimage Resistant Hash Functions*. In: *Post-Quantum Cryptography*. Johannes Buchmann J. and J. Ding ed. Lecture Notes in Computer Science, Springer Berlin / Heidelberg, 2008, vol. 5299, pp. 109–123. Available at: http://dblp.uni-trier.de/db/conf/pqcrypto/pqcrypto2008.html#DahmenOTV08 (accessed 01 March 2023).
8. Moldovyan A. A., Moldovyan N. A. Post-quantum signature algorithms based on the hidden discrete logarithm problem. *Computer Science Journal of Moldova*, 2018, vol. 26, no. 3(78), pp. 301–313.
9. Moldovyan D. N., Moldovyan A. A., Moldovyan N. A. Digital signature scheme with doubled verification

equation. *Computer Science Journal of Moldova*, 2020, vol. 28, no.1(82), pp. 80–103.
10. Moldovyan D. N. A practical digital signature scheme based on the hidden logarithm problem. *Computer Science Journal of Moldova*, 2021, vol. 29, no. 2(86), pp. 206–226.
11. Moldovyan D. N. New form of the hidden logarithm problem and its algebraic support. *Bulletin of Academy of Sciences of Moldova. Mathematics*, 2020, no. 2 (93), pp. 3–10.
12. Roman'kov V., Ushakov A., Shpilrain V. Algebraic and quantum attacks on two digital signature schemes. *Journal of Mathematical Cryptology*, 2023, vol. 17, no. 1, pp. 20220023. doi:10.1515/jmc-2022-0023
13. Moldovyan A. A., Moldovyan D. N., Moldovyan N. A. A novel method for developing post-quantum digital signature algorithms on non-commutative associative algebras. *Informatsionno-upravliaiushchie sistemy* [Information and Control Systems], 2022, no. 1, pp. 44–53. doi:10.31799/1684-8853-2022-1-44-53
14. Moldovyan D. N. A new type of digital signature algorithms with a hidden group. *Computer Science Journal of Moldova*, 2023, vol. 31, no. 1(91), pp. 111–124.
15. Ding J., Petzoldt A., Schmidt D. S. *Multivariate Cryptography*. In: *Multivariate Public Key Cryptosystems. Advances in Information Security*. Springer, New York, 2020, vol. 80, pp. 7–23. doi:10.1007/978-1-0716-0987-3_2
16. Ding J., Petzoldt A., Schmidt D. S. *Solving Polynomial Systems*. In: *Multivariate Public Key Cryptosystems. Advances in Information Security*. Springer, New York, 2020, vol. 80, pp. 185–248. doi:10.1007/978-1-0716-0987-3_8
17. Øygarden M., Felke P., Raddum H., Cid C. Cryptanalysis of the multivariate encryption scheme EFLASH. *Topics in Cryptology — CT-RSA 2020*. Lecture Notes in Computer Science, 2020, vol. 12006, pp. 85–105.
18. Ding J., Petzoldt A., Schmidt D. S. *MQDSS*. In: *Multivariate Public Key Cryptosystems. Advances in Information Security*. Springer, New York, 2020, vol. 80, pp. 153–168.
19. Ding J., Schmidt D. Rainbow, a new multivariable polynomial signature scheme. *Conference on Applied Cryptography and Network Security – ACNS 2005*. Springer Lecture Notes in Computer Science, 2005, vol. 3531, pp. 164–175.
20. *Rainbow Signature. One of Three NIST Post-quantum Signature Finalists*. 2021. Available at: https://www.pqcrainbow.org/ (accessed 01 March 2023).
21. *GeMSS: A Great Multivariate Short Signature*. Available at: https://www-polsys.lip6.fr/Links/NIST/GeMSS.html (accessed 01 March 2023).
22. Park A., Shim K.-A., Koo N., and Han D.-G. Side-channel attacks on post-quantum signature schemes based on multivariate quadratic equations. *Rainbow and UOV. IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2018, vol. 2018, no. 3, pp. 500–523. doi:10.46586/tches.v2018.i3.500-523
23. Moldovyan A. A., Moldovyan N. A. Post-quantum algebraic signature algorithms with a hidden group. *Informatsionno-upravliaiushchie sistemy* [Information and Control Systems], 2023, no. 1, pp. 29–40. doi:10.31799/1684-8853-2023-1-29-40
24. Moldovyan D. N., Moldovyan A. A., Moldovyan N. A. A novel method for development of post-quantum digital signature schemes. *Informatsionno-upravliaiushchie sistemy* [Information and Control Systems], 2020, no. 6, pp. 21–29. doi:10.31799/1684-8853-2020-6-21-29
25. Moldovyan N. A. Unified method for defining finite associative algebras of arbitrary even dimensions. *Quasigroups and Related Systems*, 2018, vol. 26, no. 2, pp. 263–270.
26. Faugére J.-C. A new efficient algorithm for computing Gröbner basis (*F*4). *J. Pure Appl. Algebra*, 1999, vol. 139, no. 1–3, pp. 61–88.
27. Faugére J.-C. A new efficient algorithm for computing Gröbner basis without reduction to zero (*F*5). *Proc. of the Intern. Symp. on Symbolic and Algebraic Computation*, 2002, pp. 75–83. doi:10.1145/780506.780516
28. Ding J., Petzoldt A. Current state of multivariate cryptography. *IEEE Security and Privacy Magazine*, 2017, vol. 15, no. 4, pp. 28–36.

УДК 003.26
doi:10.31799/1684-8853-2023-3-59-69
EDN: GXPTKZ

Постквантовые алгоритмы цифровой подписи со скрытой группой и удвоенным проверочным уравнением

А. А. Молдовян[а], доктор техн. наук, главный научный сотрудник, orcid.org/0000-0001-5480-6016
Н. А. Молдовян[а], доктор техн. наук, главный научный сотрудник, orcid.org/0000-0002-4483-5048, nmold@mail.ru
[а]Санкт-Петербургский Федеральный исследовательский центр РАН, 14-я линия В. О., 39, Санкт-Петербург, 199178, РФ

Введение: одной из текущих актуальных проблем криптографии является разработка постквантовых алгоритмов электронной цифровой подписи со сравнительно малыми размерами открытого ключа и подписи. **Цель:** разработать новый способ построения постквантовых алгебраических алгоритмов цифровой подписи со скрытой группой, основанных на вычислительной сложности решения больших систем квадратных уравнений с многими неизвестными, обеспечивающий уменьшение размеров открытого ключа и подписи по сравнению с известными аналогами. **Результаты:** предложен новый способ построения алгоритмов цифровой подписи с подписью вида (e, $\mathbf{S}$), где e — натуральное число (параметр рандомизации) и $\mathbf{S}$ — вектор (подгоночный параметр), позволяющий уменьшить размерность конечных некоммутативных ассоциативных алгебр, используемых в качестве алгебраического носителя. Способ отличается использованием приема удвоения проверочного уравнения для фиксирования скрытой группы, в котором формирование вектора $\mathbf{S}$ выполняется в зависимости от случайного обратимого вектора и тем самым устраняется влияние числа подписанных документов на стойкость в известных алгоритмах-аналогах. Способ апробирован разработкой конкретного постквантового алгоритма цифровой подписи, использующего алгебры различных размерностей в зависимости от требуемого уровня стойкости. Выполнена предварительная оценка безопасности предложенного алгоритма. **Практическая значимость:** благодаря сравнительно небольшим размерам подписи и открытого ключа рассмотренный алгоритм подписи представляет значительный практический интерес как прототип постквантового стандарта подписи.

Ключевые слова — постквантовые криптосхемы, компьютерная безопасность, электронная цифровая подпись, криптография, задача дискретного логарифмирования, конечные некоммутативные алгебры, ассоциативные алгебры, коммутативные группы.

Для цитирования: Moldovyan A. A., Moldovyan N. A. Post-quantum signature algorithms with a hidden group and doubled verification equation. *Информационно-управляющие системы*, 2023, № 3, с. 59–69. doi:10.31799/1684-8853-2023-3-59-69, EDN: GXPTKZ

УВАЖАЕМЫЕ АВТОРЫ!

Научные базы данных, включая Scopus и Web of Science, обрабатывают данные автоматически. С одной стороны, это ускоряет процесс обработки данных, с другой — различия в транслитерации ФИО, неточные данные о месте работы, области научного знания и т. д. приводят к тому, что в базах оказывается несколько авторских страниц для одного и того же человека. В результате для всех по отдельности считаются индексы цитирования, что снижает рейтинг ученого.

Для идентификации авторов в сетях Thomson Reuters проводит регистрацию с присвоением уникального индекса (ID) для каждого из авторов научных публикаций.

Процедура получения ID бесплатна и очень проста, есть возможность провести регистрацию на 12 языках, включая русский (чтобы выбрать язык, кликните на зеленое поле вверху справа на стартовой странице): https://orcid.org