Signature Algorithms with a Hidden Group, Based on Difficulty of Solving Systems of Quadratic Equations

Alla B. Levina [1], Aleksandr A. Moldovyan [2], Dmitrii N. Moldovyan [3], and Nicolay A. Moldovyan [2,*]

[1] Fundamental Foundations of Intelligent Systems Lab, Saint Petersburg Electrotechnical University "LETI", St. Petersburg, Russia; Email: alla_levina@mail.ru (A.B.L.)
[2] St. Petersburg Federal Research Center of the Russian Academy of Sciences, St. Petersburg, Russia; Email: maa1305@yandex.ru (A.A.M.)
[3] Department of Information Systems, Saint Petersburg Electrotechnical University "LETI", St. Petersburg, Russia; Email: mdn.spectr@mail.ru (D.N.M.)
*Correspondence: nmold@mail.ru (N.A.M.)

***Abstract*—This research introduces two novel algebraic signature algorithms with a hidden group, which are based on the computational difficulty of finding a solution of a large system of quadratic multivariate equations. Like signature algorithms of multivariate cryptography, the developed ones represent interest as post-quantum cryptoschemes, the latter having a significant merit consisting of a 100 or more times smaller size of public key. The introduced algorithms represent interest as candidates for a practical post-quantum signature standard with small sizes for both the signature and the public key. Their security is estimated to be vulnerable to direct attacks using the known algorithms for solving the systems of many quadratic equations. The development of structural attacks exploiting properties of the used algebraic support is estimated as an independent research task composing the next stage of the analysis of the proposed signature algorithms on finite non-commutative associative algebras. Relatively the known algebraic signature algorithms with a hidden group, which are based on the computational difficulty of the so called hidden discrete logarithm problem, the developed signature algorithms differ in the use of the signature verification equations with multiple entry of signature. This feature defines a specific technique for the signature generation. The next peculiarity is the use of algebras of higher dimensions set over the finite fields of smaller orders.**

***Keywords*—computer security, post-quantum cryptography, multivariate cryptography, digital signature, finite algebra, associativity, non-commutativity, hidden group**

I. Introduction and Preliminaries

Information security items are very important in telemedicine [1], e-government, the Internet of Things [2], cloud computing [3], and fog computing systems [4]. Information protection and information authentication can be provided by different techniques; some of the latter use "blockchain" technology [5, 6], but the most flexible and efficient techniques relate to the use of algorithms and protocols for public-key cryptography. Today, cryptographic schemes based on the computational difficulty of the Discrete Logarithm Problem (DLP) and the Factorization Problem (FP) have found the greatest practical application. The current official standards on the public-key cryptographic algorithms are also based on the computational difficulty of the DLP and FP, although polynomial algorithms have been known for more than 20 years [7–9].

With the appearance of a quantum computer in practice, DLP-based and FP-based cryptographic algorithms cease to be secure. The possibility of the practical use of these algorithms was determined by the generally accepted assumption of a negligible probability of the appearance in the near future of a practical multiqubit quantum computer. However, the development of technology for creating quantum computers forced the National Institute of Standards and Technology (NIST) to declare at the end of 2016 that after 2025 a quantum computer could unexpectedly appear in practice and to announce a worldwide competition for the development of post-quantum standards for public-key cryptalgorithms in the following two nominations:

- Public key-agreement and public encryption algorithms,
- Digital signature algorithms [10].

Despite the fact that about 20 candidates for the post-quantum signature algorithm were submitted for consideration, after three stages of the competition, NIST announced the submission of additional candidates for the post-quantum signature standard within the fourth stage of the competition [11]. This is due to some practical issues with the signature algorithms selected as finalists. Their disadvantage is the large size of the signature and/or public key [12].

Manuscript received March 8, 2023; revised May 29, 2023; accepted July 3, 2023; published November 22, 2023.

Recently proposed algebraic digital signature algorithms with a hidden group, based on the computational difficulty of so-called hidden DLP [13, 14], have fairly small signature and public key sizes. However, there are problems justifying their post-quantum security associated with the potential possibility of reducing the hidden DLP to the usual DLP. At the beginning of 2022, another type of hidden-group algebraic algorithm was proposed, characterized by the fact that it uses the computational difficulty of finding a solution to a system of many quadratic equations with many unknowns [15]. Taking into account that quantum computers are not efficient for solving the latter problem [16, 17], we can say that the second type of algebraic algorithm with a hidden group is of special interest for developing practical post-quantum signature algorithms.

In 1988, for the first time, the computational difficulty of solving systems of many quadratic equations was used to construct digital signature and public encryption algorithms in [18]. The method used in the paper [18] led to the appearance of a new type of cryptosystems called Multivariate Public Key Cryptosystems (MPKCs). A large number of multivariate signature algorithms are currently known [19–21], including Rainbow, which is one of the finalists in the NIST competition [22]. A merit of the multivariate signature algorithms is the small size of the signature. However, their significant drawback is the very large size of the public key, which is associated with a specific method for constructing them, including specifying the public key as a set of power (usually square and cubic) polynomials that describe a trapdoor one-way mapping of vectors of large dimensions (from 30 to 200), given over a finite field of comparatively small order (from 2^2 to 2^{16}).

Over the past three decades of research in the field of MPKCs, the cryptographic community has well worked out the basic methods for constructing MPKCs and proposed various algorithms of this type: several versions of Rainbow [22–24], Unbalanced Oil and Vinegar (UOV) signature schemes [25], Square [26], FLASH [27], ZHFE [28], GeMSS [29], and others. Algorithm cryptanalysis methods of the specified type are also well worked out. Typically, the following two types of attacks are distinguished [16, 19]: (1) direct attacks based on the algorithms for solving systems of many power equations with many unknowns and (2) structural attacks using the structural features of the design of MPKCs.

The most effective direct attack is the use of algorithms for solving systems of many power equations based on the calculation of the Gröbner basis [30, 31]. Structural attacks use the features of the superposition of linear and nonlinear transformations as a public key. Several types of structural attacks have been introduced, considering attacks on Rainbow [32, 33], Square [34], FLASH [35], and the Tame Transformation Signatures family [36].

A high assessment of the results obtained in the MPKC research area is the selection of the Rainbow algorithm as one of the finalists and of the GeMSS as one of the alternative algorithms in the NIST competition [11] in the nomination of post-quantum digital signature algorithms. The MPKC algorithms represented the single type of signature schemes based on the computational difficulty of solving a system of large square equations until 2022 when an alternative method for constructing algorithms based on the said problem was proposed in [15]. That paper introduced a novel method for designing digital signature algorithms with a hidden group, using Finite Non-commutative Associative Algebras (FNAA) as algebraic support, and proposed a specific algorithm implementing that method, using a verification equation with two entries of the signature. A significant merit of the method [15] is the small size of both the signature and the public key, which provides the potential possibility of developing practical post-quantum signature algorithms and a post-quantum signature standard.

However, in Ref. [15], the issues of justification of the selection of the dimension of the FNAA used as algebraic support and the choice of the 256-bit order of the finite field, over which the FNAA is set, had not been considered.

This article discusses the mentioned issues and develops two new practical post-quantum algorithms with three and four entries of the signature in the verification equation. The design of the algorithms allows using the FNAAs set over the ground fields $GF(p)$, with the order p having a size of 81 to 129 bits. The latter provides a smaller public key and signature in comparison with the signature algorithm from [15] at the same security level.

The article is organized as follows: Section II describes the used algebraic supports and the main provisions of the design concept [15], Sections III and IV introduce two new algebraic algorithms with a hidden group, using verification equations with three and four entries of the signature. Section V discusses the features of the proposed algorithms in comparison with MPKC algorithms and with algorithms based on hidden DLP, Section VI concludes the article.

The authors hope that the concept [15] and proposed two new signature algorithms will attract attention from the cryptographic community, and someone will be inspired to use the paradigm of the algebraic signature algorithms with a hidden group to develop an application for submitting to the NIST additional post-quantum signature submission call [11].

II. Finite Non-commutative Associative Algebras Used as Algebraic Support

Suppose in a finite m-dimensional vector space over a finite ground field $GF(p)$ in which a vector multiplication operation possessing the property of distributivity at the left and at the right relative to the addition operation, is defined additionally to the scalar multiplication and addition operations. Then the said vector space is called m-dimensional algebra. A vector $\mathbf{A}$ can be represented as an ordered set of its coordinates: $\mathbf{A} = (a_0, a_1, \ldots, a_{m-1})$ or as a sum of its components: $\mathbf{A} = a_0\mathbf{e}_0 + a_1\mathbf{e}_1 + \ldots + a_{m-1}\mathbf{e}_{m-1}$, where $\mathbf{e}_i$ ($i = 0, 1, \ldots, m - 1$) are formal basis vectors.

Usually, the multiplication of the vectors $\mathbf{A} = \sum_{i=0}^{m-1} a_i \mathbf{e}_i$ and $\mathbf{B} = \sum_{j=0}^{m-1} b_j \mathbf{e}_j$ is defined by the following Eq. (1):

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

in which the values a_i and b_i are multiplied as elements of the field $GF(p)$ and every of the products $\mathbf{e}_i\mathbf{e}_j$ is replaced by a single-component vector $\lambda\mathbf{e}_k$, where $\lambda \in GF(p)$, given in the cell located in the i-th row and in the j-th column of a specially constructed table called Basis Vector Multiplication Table (BVMT), like Table I in the case $m = 4$ [12] and Table II in the case $m = 6$ [37]. If $\lambda \neq 1$, then the value λ is called the structural constant. If $\lambda = 0$, then in the corresponding cell of the BVMT it is indicated as zero without any basis vector. A BVMT with cells containing zeros is called a sparse BVMT. The use of sparse BVMTs to set FNAAs represents a technique for reducing the computational complexity of the multiplication operation.

In the algebraic signature schemes with a hidden group, the exponentiation operations are used in the signature generation and verification procedures. This implies the possibility of using a fast exponentiation algorithm, therefore, the multiplication operation must be associative. From Eq. (1) it can be seen that the vector multiplication operation is associative if the BVMT is constructed so that the condition:

$$(\mathbf{e}_i\mathbf{e}_j)\mathbf{e}_k = \mathbf{e}_i(\mathbf{e}_j\mathbf{e}_k) \quad (2)$$

is true for all possible triples of indices (i, j, k).

A vector **E**, such that the equalities $\mathbf{EV} = \mathbf{V}$ and $\mathbf{VE} = \mathbf{V}$ hold true for every element of some FNAA, is called a two-sided global unit of the algebra.

In the framework of some FNAA, one can consider different subsets of reversible vectors that compose multiplicative groups. A finite set of reversible algebraic elements with an associative binary operation (for example, multiplication) is called a finite (multiplicative) group.

In FNAAs with global two-sided units, an algebra element (vector) **V** is called reversible, if the algebra contains the single element, denoted as $\mathbf{V}^{-1}$, which satisfies the equalities $\mathbf{V}\mathbf{V}^{-1} = \mathbf{E}$ and $\mathbf{V}^{-1}\mathbf{V} = \mathbf{E}$. All reversible elements of the FNAA with the multiplication operation compose a group called the multiplicative group of the algebra.

A minimum set of the group elements, products of different powers which take on all values in the group, is called a minimum generator system (basis) of the group. A finite group generated by a single element (the generator of the group) is called a cyclic group. A finite group with a basis containing two elements of the same order is called a group with two-dimensional cyclicity.

For each of the signature algorithms introduced in Sections III and IV, it is assumed that the use of the FNAAs of four different dimensions: $m = 4$, 6, 8, and 10. For the case $m = 4$, the used FNAA is set by a sparse BVMT shown in Table I. Using a BVMT with eight cells containing the zero structural constants provides a two-times reduction in the computational complexity of the multiplication and exponentiation operations.

Decomposition of the said 4-dimensional FNAA into commutative subalgebras had been studied in detail in [12], where the following results had been obtained:

- The 4-dimensional FNAA contains $p^2 + p + 1$ of commutative subalgebras of the order p^2, every pair of which intersects exactly in the set of scalar vectors $\{\mathbf{L}: \mathbf{L} = h\mathbf{E}, h = 0, 1, \ldots, 2^z - 1\}$, where $\mathbf{E} = (1, 1, 0, 0)$ is the global two-sided unit;
- The order of multiplicative group Γ of the algebra is equal to:
$$\Omega = p(p-1)(p^2-1); \quad (3)$$
- The group Γ contains $p(p + 1)/2$ commutative subgroups Γ_1 possessing two-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:
$$\Omega_1 = (p-1)^2; \quad (4)$$
- The group Γ contains $p(p - 1)/2$ commutative cyclic subgroups Γ_2 of the order:
$$\Omega_2 = p^2 - 1 = (p-1)(p+1); \quad (5)$$
- The group Γ contains $p + 1$ commutative cyclic subgroups Γ_3 of the order:
$$\Omega_3 = p(p-1); \quad (6)$$
- The condition of invertibility of the vector $\mathbf{A} = (a_0, a_1, \ldots, a_{m-1})$ is given by the non-equality:
$$a_0 a_1 \neq \lambda a_2 a_3. \quad (7)$$

To define the FNAAs of the dimensions $m = 6$, 8, and 10, which are suitable for using them as algebraic support of the developed signature algorithms with a hidden group, we use the unified method [37] for setting FNAAs of arbitrary even dimensions. That method consists in using the following Eq. (8) for generating BVMTs setting non-commutative associative vector multiplication operations for arbitrary even values $m \geq 6$:

$$e_i e_j = \begin{cases} e_{i+j \bmod m}, \text{ if } i \bmod 2 = 0; \\ e_{i-j \bmod n}, \text{ if } i \bmod 2 = 1,\ j \bmod 2 = 0; \\ \lambda e_{i-j \bmod m}, \text{ if } i \bmod 2 = 1,\ j \bmod 2 = 1. \end{cases} \quad (8)$$

The respective proof is provided in [37], using the Eqs. (1) and (2). For the case of dimensions $m \geq 6$ we have not found a suitable sparse BVMT to provide a lower computational complexity for the multiplication and exponentiation operations used in the developed signature algorithm.

For the cases $m = 6$ and $m = 8$ Eq. (8) gives the BVMTs shown in Tables II and III. The structure of the FNAAs set by Tables II and III (from the point of view of the decomposition into commutative subalgebras) is an

open problem, but from [37] it is known that the FNAA set by Table II contains commutative groups of the orders described by the Eqs. (4) and (5). We have also experimentally checked that for the cases $m = 8$ and $m = 10$ the said types (Γ_1 and Γ_2) of commutative groups are also contained in the FNAAs. Besides, we also use the following two experimental facts:

- A non-scalar vector selected at random in each of the said 8-dimensional and 10-dimensional algebras with sufficiently high probability (>0.1) has order $p - 1$ for values p having different sizes;
- A vector selected at random in each of the said 8-dimensional and 10-dimensional FNAAs with sufficiently high probability (>0.1) has order $p^2 - 1$ for values p having different sizes.

At this point, we should note that a more convincing justification for a sufficiently high probability of choosing vectors with the required order value must be based on theoretical consideration. The latter requires consideration of the structure of FNAAs for cases of dimensions $m \geq 8$, for example, using the approach [12]. However, the study of the decomposition of KNAA into commutative subalgebras for the dimensions $m \geq 8$ is a more time-consuming independent study.

The generation of the hidden group of the Γ_2 type consists of selecting a random vector of the order equal to $p^2 - 1$. Generation of the hidden group of the Γ_1 type can be performed using the following algorithm:

Algorithm 1. Generation of the Basis of a Random Commutative Group of the Γ1 Type

- Select a random non-scalar vector **G** of order $p - 1$.
- Select a random primitive element α in $GF(p)$ and generate two random integer values, $k < p - 1$ and $t < p - 1$.
- Compute the vector $\mathbf{H} = \alpha^k \mathbf{G}^t$ (evidently, vector **H** has an order equal to $p - 1$).
- Output the basis <**G**, **H**> of a random commutative group of the Γ_1-type.

TABLE I. DEFINING VECTOR MULTIPLICATION OPERATIONS IN THE 4-DIMENSIONAL FNAA WITH THE TWO-SIDED GLOBAL UNIT $E = (1, 1, 0, 0)$ [12] ($\lambda \neq 0$)

·	$\mathbf{e}_0$	$\mathbf{e}_1$	$\mathbf{e}_2$	$\mathbf{e}_3$
$\mathbf{e}_0$	$\mathbf{e}_0$	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 II. DEFINING VECTOR MULTIPLICATION OPERATION IN THE 6-DIMENSIONAL FNAA WITH THE TWO-SIDED GLOBAL UNIT E = (1, 0, 0, 0, 0, 0) [37] ($\lambda \neq 0$)

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

TABLE III. DEFINING VECTOR MULTIPLICATION OPERATION IN THE 8-DIMENSIONAL FNAA WITH THE TWO-SIDED GLOBAL UNIT E = (1, 0, 0, 0, 0, 0, 0, 0) ($\lambda \neq 0$)

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

The method proposed in [15] for designing algebraic signature algorithms with a hidden group implies the use of the signature in the form of two elements one of which is a number e and the other one is a vector **S**. The value **S** is used as a multiplier in the signature verification equation. Such an occurrence of the value **S** in the verification equation creates prerequisites for signature forgery by solving the verification equation concerning an unknown value **S**. To prevent such potential attacks, the idea of multiple entries of the vector **S** is proposed in [15].

Namely, the power verification equation is given in a non-commutative algebra and includes two or more entries of the vector **S**. To provide the possibility for the signer to calculate a signature satisfying the verification equation, specific mechanisms for computing the public key and generating the signature are proposed. A four-dimensional FNAAs set by a sparse BVMT over the field $GF(p)$ with prime characteristic $p = 2q + 1$, where q is also a prime, is used as algebraic support in [15]. The decomposition of the used FNAA into the set of commutative subalgebras had also been studied in [15], the results similar to [12] had been obtained, i.e., the FNAA used in [15] contains a sufficiently large number of commutative groups of Γ_1-type.

To generate a public key, a potential signer generates a secret basis <**G**, **H**> (where the vectors **G** and **H** have the same order equal to q) of the hidden group, generates random secret pairwise non-permutable vectors **A**, **B**, and **C** such that every one of them is non-permutable with **G** and **H**. Then he calculates the public key in the form of the following four vectors: $\mathbf{Y}_1 = \mathbf{AG}^u\mathbf{B}$; $\mathbf{Z}_1 = \mathbf{CHA}^{-1}$; $\mathbf{Y}_2 = \mathbf{AH}^w\mathbf{B}$; $\mathbf{Z}_1 = \mathbf{CGA}^{-1}$.

The signature generation procedure includes the following steps:

- Select the random integers $k < q$ and $t < q$ and calculate the vector **R**:

$$\mathbf{R} = \mathbf{AG}^k\mathbf{H}^t\mathbf{A}^{-1}.$$

- Using a specified hash function f, calculate the value $e = f_H(M, \mathbf{R})$, where M is a document to be signed.
- Calculate the powers n and r using the following Equations:

$$n = \frac{k - ue - e^2}{e + e^2} \bmod q;$$

$$r = \frac{t - we^2 - e}{e + e^2} \bmod q.$$

- Compute the second signature element **S**: $\mathbf{S} = \mathbf{B}^{-1}\mathbf{G}^{n}\mathbf{H}^{r}\mathbf{C}^{-1}$.

Verification of the signature (e, **S**) is performed as follows:

- Compute the vector **R***:

$$\mathrm{R}^* = \left(\mathrm{Y}_1\mathrm{SZ}_1\right)^{e}\left(\mathrm{Y}_2\mathrm{SZ}_2\right)^{e^2}.$$

- Calculate the value $e^* = f(M, \mathbf{R}^*)$.
- If $e^* = e$, then the signature (e, **S**) is genuine. Otherwise, the signature is false.

Thus, the elements of the public key ($\mathbf{Y}_1$, $\mathbf{Z}_1$, $\mathbf{Y}_2$, $\mathbf{Z}_2$) are calculated in such a way that the right part of the verification equation can be presented in the form $\mathbf{AG}^{a}\mathbf{H}^{b}\mathbf{A}^{-1}$, if the signature element has the form $\mathbf{S} = \mathbf{B}^{-1}\mathbf{G}^{n}\mathbf{H}^{r}\mathbf{C}^{-1}$ and the power values n and r can be computed so that $a = k$ and $b = t$, providing correctness of the signature scheme.

When calculating the public key, the exponentiation operations are used as an efficient technique for selecting random elements from the hidden group, which improves the performance of the signature generation algorithm. In the signature generation and verification procedures, exponentiation operations are used as a technique for calculating the genuine value of the signature element S after the signature element e is calculated depending on the given document and the random vector **R** (the latter is calculated at the first step of the signature generation procedure). Actually, the secret key represents the set of vectors **G**, $\mathbf{G}_u$, **H**, $\mathbf{H}_w$, **A**, **B**, and **C** (where $\mathbf{G}_u = \mathbf{G}$ and $\mathbf{H}_w = \mathbf{H}^w$).

Suppose $|q|$ denotes the bit size of the value q (the order of the vectors **G** and **H**). A limitation of the algorithm [15] is connected with the fact that the size of $|q|$ should be equal to or higher than the size of the used 256-bit hash function f_H. The latter determines the 257-bit size of the prime p. In the proposed signature algorithms we use the verification equations with two exponentiations to independent power degrees e_1 and e_2 such that $e_1||e_2 = e = f_H(M||\mathbf{R})$, where || denotes the concatenation operation. This technique allows one to use the value of $|q|$ which is half the size of the hash value. Thanks to the latter, the size of the public key and signature is reduced, and the performance of the digital signature algorithm is also increased.

In this paper, we also define the FNAAs over the field $GF(p)$ with characteristic $p = 2q + 1$, where q is a prime. The value p is selected depending on the value of the dimension m. We specify the use of the characteristic p having the size 129 (for $m = 4$), 97 to 129 (for $m = 6$), and 81 to 129 bits (for $m = 8$ and $m = 10$). For larger values m we have more quadratic equations and unknowns. The computational complexity of the direct attacks depends slightly on the size of the characteristic p. However, we suppose that the security against structural attacks, which can appear in the future, will be significantly dependent on the values m and p.

III. The Signature Algorithm with Three Entires of the Value S

The secret key is generated in the form of five vectors **A**, **B**, **D**, **G**, **H,** and three natural numbers u, w, x ($1 < u, w, x < q$). The vectors **G** and **H** have order equal to the prime q and compose the basis <**G**, **H**> of the hidden group. The vectors **A**, **B**, and **D** are generated at random so that the following non-equalities hold: $\mathbf{AB} \neq \mathbf{BA}$, $\mathbf{AD} \neq \mathbf{DA}$, $\mathbf{AG} \neq \mathbf{GA}$, $\mathbf{BD} \neq \mathbf{DB}$, $\mathbf{BG} \neq \mathbf{GB}$, $\mathbf{DG} \neq \mathbf{GD}$.

The public key is computed in the form of five vectors **Y**, **Z**, **Q**, **U**, and **T** as follows:

$$\mathbf{Y} = \mathbf{AGB},\ \mathbf{Z} = \mathbf{DHB},\ \mathbf{Q} = \mathbf{AG}^{u}\mathbf{D}^{-1},$$

$$\mathbf{U} = \mathbf{DG}^{x}\mathbf{A}^{-1} \text{ and } \mathbf{T} = \mathbf{B}^{-1}\mathbf{H}^{w}\mathbf{A}^{-1}. \quad (9)$$

Eq. (9) describes the method used for masking the vectors contained in the hidden group, with the left and right masking multipliers representing the elements of the private key.

A. Signature Generation Procedure

- Generate at random the natural numbers k and t such that $1 < k < q$ and $1 < t < q$ and calculate the vector $\mathbf{R} = \mathbf{AG}^{k}\mathbf{H}^{t}\mathbf{A}^{-1}$.
- Using a specified collision resistant $2|q|$-bit hash-function f_H, calculate the value $e = e_1||e_2 = f_H(M||\mathbf{R})$, where M is a document to be signed and the hash-value e is represented in the form of the concatenation of two 128-bit numbers e_1 and e_2.
- Compute the integer numbers n and d by the following Eqs. (10) and (11):

$$n = \frac{k - \left(e_1 + xe_1 + ue_2\right)}{2e_1 - e_2} \bmod q \quad \text{and} \quad (10)$$

$$d = \frac{t - \left(e_1 + we_2\right)}{2e_1 - e_2} \bmod q. \quad (11)$$

- Calculate the vector **S** by the Eq. (12)

$$\mathbf{S} = \mathbf{B}^{-1}\mathbf{G}^{n}\mathbf{H}^{d}\mathbf{D}^{-1}. \quad (12)$$

The vector **S** represents the fitting parameter of the signature, which is calculated so that the signature verification equation will be satisfied. Eqs. (11)–(13) are derived from consideration of the algorithm correctness proof presented below.

The signature represents the integer e and the vector **S**. The main contribution to the computational difficulty of the signature generation procedure is introduced by four exponentiation operations (about 6200 multiplications in $GF(p)$ for the case $m = 4$ and a 128-bit value q).

B. Signature Verification Procedure

- Compute the vector **R′**:

$$\mathbf{R}' = \left(\mathbf{YSZSU}\right)^{e_1}\left(\mathbf{QS}^{-1}\mathbf{T}\right)^{e_2} \quad (13)$$

with three entries of the vector **S** and two exponentiation operations.

- Compute the hash-function value $e' = f_H(M||\mathbf{R}')$.
- Compare the values e' and e. If $e' = e$, then the signature is accepted as genuine. Otherwise ($e' \neq e$) the signature is rejected as a false one.

The computational difficulty of the signature verification procedure is defined by two exponentiation operations (about 3100 multiplications in $GF(p)$ for the case $m = 4$ and a 128-bit value q).

C. The Correctness Proof of the Signature Scheme

The signature scheme is correct if the correctly computed signature passes the verification procedure as a genuine one. Taking into account Eqs. (10)–(13) we have:

$$\begin{aligned}
&\mathrm{R}' = (\mathrm{YSZSU})^{e_1}\left(\mathrm{QS}^{-1}\mathrm{T}\right)^{e_2} = \\
&= \left(\mathrm{AG}...\mathrm{G}^{n}\mathrm{H}^{d}...\mathrm{H}...\mathrm{G}^{n}\mathrm{H}^{d}...\mathrm{G}^{x}\mathrm{A}^{-1}\right)^{e_1} \times \\
&\times \left(\mathrm{AG}^{u}...\mathrm{G}^{-n}\mathrm{H}^{-d}...\mathrm{H}^{w}\mathrm{A}^{-1}\right)^{e_1} = \\
&= \left(\mathrm{AG}^{e_1+xe_1+2ne_1}\mathrm{H}^{2de_1+e_1}\mathrm{A}^{-1}\right)\times \\
&\times \left(\mathrm{AG}^{ue_2-ne_2}\mathrm{H}^{-de_2+we_2}\mathrm{A}^{-1}\right) = \\
&= \mathrm{AG}^{e_1+xe_1+2ne_1+ue_2-ne_2}\mathrm{H}^{2de_1+e_1-de_2+we_2}\mathrm{A}^{-1} = \\
&= \mathrm{AG}^{n(2e_1-e_2)+e_1+xe_1+ue_2}\mathrm{H}^{d(2e_1-e_2)+e_1+we_2}\mathrm{A}^{-1} = \\
&= \mathrm{AG}^{k}\mathrm{H}^{t}\mathrm{A}^{-1} = \mathrm{R}.
\end{aligned}$$

IV. The Signature Algorithm with Four Entires of the Value S

The secret key is generated in the form of five vectors **A**, **B**, **D**, **G**, and **H**, and three natural numbers u, w, x ($1 < u, w, x < q$), where the vectors **G** and **H** have order equal to the prime q and compose the basis <**G**, **H**> of the hidden group. The vectors **A**, **B**, and **D** are generated at random so that the following non-equalities hold: $\mathbf{AB} \neq \mathbf{BA}$, $\mathbf{AD} \neq \mathbf{DA}$, $\mathbf{AG} \neq \mathbf{GA}$, $\mathbf{DB} \neq \mathbf{BD}$, $\mathbf{BG} \neq \mathbf{GB}$, $\mathbf{DG} \neq \mathbf{GD}$.

The public key is computed in the form of vectors **Y**, **Z**, **Q**, **T**, **U** by the following Eq. (14):

$$\mathbf{Y} = \mathbf{AGB},\ \mathbf{Z} = \mathbf{DHB},\ \mathbf{Q} = \mathbf{DG}^{x}\mathbf{A}^{-1},\ \mathbf{T} = \mathbf{AH}^{w}\mathbf{D}^{-1},\ \mathbf{U} = \mathbf{B}^{-1}\mathbf{G}^{u}\mathbf{A}^{-1}. \quad (14)$$

A. Signature Generation Procedure

- Generate at random the natural numbers k and t ($1 < k < q$; $1 < t < q$) and calculate the vector:

$$\mathbf{R} = \mathbf{AG}^{k}\mathbf{H}^{t}\mathbf{A}^{-1}. \quad (15)$$

- Using a specified collision resistant $2|q|$-bit hash-function f_H, calculate the value $e = e_1||e_2 = f_H(M||\mathbf{R})$, where M is a document to be signed and the hash-value e is represented in the form of the concatenation of two 128-bit numbers e_1 and e_2.
- Compute the integer numbers n and d by the following two Eqs. (16) and (17):

$$n = \frac{k - (e_1 + e_2 + u + xe_1 + xe_2)}{e_1 + 2e_1 - 1} \bmod q \quad (16)$$

$$d = \frac{t - (w + e_2)}{e_1 + 2e_1 - 1} \bmod q \quad (17)$$

- Calculate the vector **S** by Eq. (18):

$$\mathbf{S} = \mathbf{B}^{-1}\mathbf{G}^{n}\mathbf{H}^{d}\mathbf{D}^{-1}. \quad (18)$$

The signature represents the pair (e, **S**). The computational difficulty of the signature generation procedure is defined by four exponentiation operations (about 27700 multiplications in $GF(p)$ for the case $m = 6$ and a 128-bit value q). Two exponentiations are performed for each of the value's **R** and **S**.

B. Signature Verification Procedure

- Compute the vector $\mathbf{R}'$:

$$\mathbf{R}' = (\mathbf{YSQ})^{e_1}\,\mathbf{TS}^{-1}\mathbf{U}(\mathbf{YSZSQ})^{e_2} \quad (19)$$

with four entries of the vector **S** and two exponentiation operations.

- Compute the hash-function value $e' = f_H(M||\mathbf{R}')$.
- Compare the values e' and e. If $e' = e$, then the signature is accepted as a genuine one. Otherwise ($e' \neq e$) the signature is rejected as a false one.

C. The Correctness Proof of the Signature Scheme

The signature scheme is correct if the correctly computed signature passes the verification procedure as a genuine one. Taking into account Eqs. (14)–(19) we have:

$$\begin{aligned}
&\mathrm{R}' = (\mathrm{YSQ})^{e_1}\,\mathrm{TS}^{-1}\mathrm{U}(\mathrm{YSZSQ})^{e_2} = \\
&= \left(\mathrm{AG}...\mathrm{G}^{n}\mathrm{H}^{d}...\mathrm{G}^{x}\mathrm{A}^{-1}\right)^{e_1} \times \\
&\times \mathrm{AH}^{w}...\mathrm{G}^{-n}\mathrm{H}^{-d}...\mathrm{G}^{u}\mathrm{A}^{-1} \times \\
&\times \left(\mathrm{AG}...\mathrm{G}^{n}\mathrm{H}^{d}...\mathrm{H}...\mathrm{G}^{n}\mathrm{H}^{d}...\mathrm{G}^{x}\mathrm{A}^{-1}\right)^{e_2} = \\
&= \left(\mathrm{AG}^{e_1+ne_1+xe_1}\mathrm{H}^{de_1}\mathrm{A}^{-1}\right)\mathrm{AH}^{w-d}...\mathrm{G}^{-n+u}\mathrm{A}^{-1} \times \\
&\times \left(\mathrm{AG}^{e_2+2ne_2+xe_2}\mathrm{H}^{2de_2+e_2}\mathrm{A}^{-1}\right) = \\
&= \mathrm{AG}^{e_1+ne_1+xe_1-n+u+e_2+2ne_2+xe_2} \times \\
&\times \mathrm{H}^{de_1+w-d+2de_2+e_2}\mathrm{A}^{-1} = \\
&= \mathrm{AG}^{n(e_1+2e_2-1)+e_1+e_2+u+xe_1+xe_2} \times \\
&\times \mathrm{H}^{d(e_1+2e_2-1)+w+e_2}\mathrm{A}^{-1} = \\
&= \mathrm{AG}^{k}\mathrm{H}^{t}\mathrm{A}^{-1} = \mathrm{R}.
\end{aligned}$$

In correspondence with the signature verification procedure, the latter equality means the correctness of the signature described scheme with four entries of the signature element **S**.

V. Discission

The multiple entries of the signature element **S** in the verification equation represent a technique for preventing signature forging attacks based on using the value of **S** as a fitting parameter. Due to multiple entries of the vector **S** in the verification equation and due to the non-

commutativity of the multiplication operation, it is computationally hard to solve the verification equation with the relatively unknown vector **S**; therefore, the mentioned attacks are prevented. Consideration of the methods for solving the power equations with multiple entries of the unknown, which are given in a FNAA represents a specific mathematical task. In the design concept of the algebraic signature algorithms with a hidden group [15], it is assumed that polynomial algorithms for solving the said equations will not be developed. One can suppose that the use of the verification equations with a larger number β of signature entries provides better protection of the signature scheme against the mentioned types of forging attacks. Therefore, we have proposed the signature algorithms with $\beta = 3$ and $\beta = 4$.

More realistic attacks on the proposed algorithms relate to computing unknown vectors **A**, **B**, **D**, **G**, and **H**, representing the elements of the public key in the form of Eqs. (9)–(14) in the case of algorithms from Sections III and IV, respectively. Such an attack can be attributed to a direct attack on the algebraic cipher with a hidden group. The direct attack is connected with solving a system of quadratic vector equations in which the unknowns are vectors. Using the BVMTs, one can reduce the system of vector equations to the system of scalar equations.

In line with the design of the algorithms with a hidden group, the system of quadratic vector equations is defined by the equations used to compute the elements of the public key and the conditions of mutual permutability of the unknown vectors from the hidden group. For example, in the case of the signature scheme with $\beta = 3$, the system includes five Eq. (9) and the following four equations:

$\mathbf{GH} = \mathbf{HG}$, $\mathbf{GG}_u = \mathbf{GG}_u$, $\mathbf{GG}_x = \mathbf{GG}_x$, $\mathbf{GH}_w = \mathbf{GH}_w$. Thus, in the case of the signature scheme from Section III, we have a system of 9 vector equations with 8 unknowns: **A**, **B**, **D**, **G**, **H**, $\mathbf{G}_u$, $\mathbf{G}_x$, and $\mathbf{H}_w$ (where $\mathbf{G}_x = \mathbf{G}^x$, $\mathbf{G}_u = \mathbf{G}^u$, and $\mathbf{H}_w = \mathbf{H}^w$). When reducing the Eq. (9) in a scalar form, each of them gives m scalar quadratic equations. So, we have a system of $9m$ scalar quadratic equations with $8m$ scalar unknowns (coordinates of the unknown vectors).

Similarly, one can show that in the case of the signature algorithm from Section IV, we also have a system of $9m$ scalar quadratic equations with $8m$ scalar unknowns.

To estimate the security W of the introduced algorithms against the direct attack, one can take the number of equations μ equal to the number of the unknowns η, and use the recommended minimum values of μ presented in Table IV for different values of the order of the field $GF(n)$ in which the system of quadratic equations is given [16]. Since in the proposed signature algorithms we use the fields $GF(p)$ with $p > n$ one can use the values μ corresponding to $n = 256$. In this case, we get overstated requirements for the minimum value, however, this overestimation can be considered insignificant due to its relatively weak dependence on the order of the field.

TABLE IV. MINIMUM NUMBER OF EQUATIONS PROVIDING A GIVEN SECURITY LEVEL TO THE DIRECT ATTACK FOR DIFFERENT FIELDS $GF(N)$ IN THE CASE $\mu = \eta$ [16]

W =…	2^{80}	2^{100}	2^{128}	2^{192}	2^{256}
$n = 16$	30	39	51	80	110
$n = 31$	28	36	48	75	103
$n = 256$	26	33	43	68	93

The values of W for different versions (differing in the dimension of the FNAAs used as algebraic support) of the introduced signature algorithms and the algorithm from [15] are presented in Table V. A comparison of the sizes of the public key and signature in the said versions of the signature algorithms is shown in Table VI. From Tables V and VI, one can see that the two introduced algorithms provide significantly shorter signature and public key (at the same or higher security level) than the algorithm from [15].

TABLE V. SECURITY LEVEL TO THE DIRECT ATTACK OF THE PROPOSED SIGNATURE ALGORITHMS FOR DIFFERENT VALUES OF M IN COMPARISON WITH THE ALGORITHM FROM [15]

m =…	4	6	8	10
Section III $\beta = 3$	$\approx 2^{100}$	$>2^{128}$	$\approx 2^{192}$	$>2^{192}$
Section IV $\beta = 4$	$\approx 2^{100}$	$>2^{128}$	$\approx 2^{192}$	$>2^{192}$
[15] $\beta = 2$	$\approx 2^{80}$	$<2^{128}$	$<2^{192}$	2^{192}

TABLE VI. SIZE (IN BYTES) OF THE PUBLIC KEY (SIGNATURE) IN THE CASE OF USING FNAAS OF DIFFERENT DIMENSIONS M

m =…	4	6	8	10
Section III $\beta = 3$	320 (96)	≤480 (≤128)	≤640 (≤160)	≤800 (≤192)
Section IV $\beta = 4$	320 (96)	≤480 (≤128)	≤640 (≤160)	≤800 (≤192)
[15] $\beta = 2$	512 (160)	768 (224)	1024 (288)	1280 (354)

Security to the direct attack is poorly dependent on the value of the order of the field $GF(p)$, however, we define using different values q (that set different values $p = 2q + 1$), since we expect that the computational difficulty of potential structural attacks will be significantly dependent on the value p. Currently, there are no known structural attacks on the signature algorithms with a hidden group, but they can be developed in the future. The known structural attacks proposed for the crypt algorithms of multivariate cryptography are not applicable to the introduced algorithms since the design of the latter is significantly different.

Consider two attacks related to forging a signature: i.e., to generate a signature without knowing the secret key. In the first one, the forger selects arbitrary values of **S** and of $e = e_1 || e_2$ and, using the signature verification Eq. (13) for the algorithm from Section III (or Eq. (19) for the algorithm from Section IV) computes the vector $\mathbf{R}'$. Then the forger calculates the hash value $e' = e'_1 || e'_2 = f_H(M || \mathbf{R}')$ until $e'_1 = e_1$ and $e'_2 = e_2$. Due to the use of collision-resistant $2|q|$-bit hash-function f_H, the probability of the latter event is equal to $2^{-2|q|}$. Therefore, the computational

difficulty of such an attack can be estimated as $O(2^{2|q|})$, where $O(\cdot)$ is the order notation).

In the second attack, the forger initiates generating 2^{128} different signatures $(e^{(i)}, \mathbf{S}^{(i)})$ for documents M_i, where $i = 1, 2, \ldots, 2^{|q|}$ (consider, for example, a model of the oracle that signs random documents M_i generated by the forger). Then he computes the set of hash-function values $h_i = f_H(M_i)$, prepares $2^{|q|}$ different documents M'_i ($i = 1, 2, \ldots, 2^{|q|}$), and computes the second set of hash-function values $h'_i = f_H(M'_i)$. In correspondence with the birthday paradox, with a probability equal to ≈0.5 the first set of hash values contains a value h_k such that $h_k = h'_t$ for some natural number $t \le 2^{|q|}$. Due to iterated structure of the algorithm for calculating the hash values $f_H(\cdot)$, the signature $(e^{(k)}, \mathbf{S}^{(k)})$ is a valid signature to the document M'_t. The computational difficulty of the second attack equals $O(2^{|q|})$. Note that the security of two developed algorithms considered forgery attacks depends mainly on the size of the used collision-resistant hash function.

The main similarity of the proposed signature algorithms with the algorithms based on the computational difficulty of the hidden DLP [12–14] is that both types of algorithms use the same type of algebraic support, namely, the FNAAs, and a hidden group as an element of the secret key. The other similarity items include:

- Exploiting the exponentiation operations, when computing the public key and performing the signature generation and verification procedures;
- Using the left-sided and right-sided multiplications as a masking operation, when computing the public key.

The main differences between the compared types of algorithms with a hidden group include the following points:

- Their security is based on different computationally hard problems;
- Usually, in the algorithms based on the hidden DLP, the signature is computed as a pair of numbers, but in the proposed algorithms as a number and a vector **S**;
- Multiple entries of the signature element **S** in the verification equation are used in the proposed algorithm as a technique for preventing forging attacks.

TABLE VII. SIZE (IN BYTES) OF THE PUBLIC KEY AND SIGNATURE IN THE ALGORITHMS FROM SECTIONS III AND IV AND FROM [13] AND [14]

Algorithm	Signature form	Signature size, bytes	Public key size, bytes
Proposed $\beta = 3$ $\beta = 4$	$(e, \mathbf{S})$	≤128	≤480
Hidden [13]	(e, s, σ)	96	384
Hidden [14]	(e, s, d, σ)	≈ 128	768

Table VII presents a comparison of the introduced signature algorithms with some of those based on the computational difficulty of the hidden DLP, at the 2^{128}-bit security level (the integers s, d, and σ are fitting elements of the signature in the algorithms based on the hidden DLP).

Table VII shows that the proposed algorithms and algorithms based on the hidden DLP provide a sufficiently small signature and public key sizes. However, the former is preferred because they are free from the potential vulnerability to algebraic attacks associated with reducing the hidden DLP problem to the usual DLP.

The main similarity of the proposed algorithms with the algorithms of multivariate cryptography [16, 18–28] is that both types of algorithms are based on the computational difficulty of solving large systems of quadratic multivariate equations. This determines the similarity of the direct attack on these algorithms. However, their designs are completely different, defining different structural attacks on them. The difference includes the following points:

- In the proposed algorithms the public key represents a set of FNAAs elements, whereas in the multivariate algorithms the public key is a map given by a set of quadratic polynomials (the latter causes the very large size of the public key);
- The signature is computed as a number e and a vector **S** in the algebraic signature algorithms with a hidden group, but in the multivariate algorithms the signature is computed as a pre-image of a hash function value;
- When performing signature generation and verification procedures, in the proposed algorithms the computations are performed in a finite field with a significantly large size of order, then in the multivariate signature algorithms.

Table VIII presents a comparison of some parameters of the proposed and known algorithms based on the difficulty of finding a solution to a system of many quadratic equations with many unknowns.

TABLE VIII. A ROUGH COMPARISON WITH SOME VERSIONS OF THE MPKC SIGNATURE ALGORITHM RAINBOW

Signature algorithm	Signature size, bytes	Public key size, bytes	$\mu(\eta)$	Order of the field in which quadratic equations are set
Rainbow [20]	33	16065	27(33)	2^8
Rainbow [22] (3 versions)	66, 164, 204	>150000 >860000 >1900000	64(96) - (-) 128(204)	2^4, 2^8, 2^8
Moldovyan *et al.* [15] (m=4) $\beta = 2$	160	512	28(28)	$>2^{256}$
Proposed (m=6) $\beta = 3$	104–128	360–480	54(48)	2^{97}–2^{129}
Proposed (m=10) $\beta = 4$	144–192	500–800	54(48)	2^{81}–2^{129}

The multiplicative group of some KNAA with the global two-sided unit is actually used as an algebraic carrier of the developed algorithms. In the general case,

another finite non-commutative group can also be used. However, to ensure sufficient performance of signature algorithms with a hidden group, the most attractive case is the use of the multiplicative group of a suitable KNAA as an algebraic support. The use of finite quasigroups as an algebraic support for the proposed signature schemes does not seem justified, since the associativity of the group operation and the reversibility of algebraic elements are mandatory properties. In the general case, quasigroups do not have such properties.

In the performed study, the FNAAs defined over the ground finite fields $GF(p)$ were considered as algebraic supports of the signature algorithms with a hidden group. In the future, it is of practical interest to implement the considered algorithms on FNAAs set over finite fields $GF(2^s)$, where s is equal to 81 to 129. The latter will potentially provide an increase in the performance of signature generation and verification procedures at a given security level.

VI. Conclusion

The introduced algebraic signature algorithms represent interest as candidates for practical post-quantum cryptoschemes. In comparison with the known signature schemes based on the computational difficulty of solving systems of many quadratic equations with many unknowns, their merit is the sufficiently small size of the public key. One can expect that the introduced signature scheme and the used concept of the design of the algebraic signature algorithms with a hidden group will attract the attention cryptographic community and will lead to numerous studies on the development of structural attacks and to a more complete justification of their security since the problem of the development of practical post-quantum signature schemes is currently a significant challenge in the area of cryptography.

The considered method creates a new direction in the development of post-quantum crypto schemes; however, it has the following limitations:

- The method is not applicable for constructing an algorithm for the public encryption and public key distribution;
- When the dimension m increases, it is not possible to reduce the order of the field to less than 2^{80}, over which the used FNAA is set;
- A detailed estimation of the security of algorithms of the considered type is associated with the study of the decomposition of the FNAA into a set of commutative subalgebras, which for the case of large dimensions is an independent laborious task.

For future research, the following tasks are important:

- A detailed study of the decomposition of FNAAs into a set of commutative subalgebras for the case of dimensions $m > 4$;
- Completion of evaluations of software and hardware implementation of the proposed EDS algorithms;
- Search for new methods for implementing algebraic algorithms based on the computational complexity of solving large systems of quadratic equations.

Conflict of Interest

The authors declare no conflict of interest.

Author Contributions

All authors contributed to the formation of this article. Specifically, as follows: Alla B. Levina had developed the signature algorithm from Section III; Alexandr A. Moldovyan had developed the algorithm from Section IV; Dmitriy N. Moldovyan had performed security estimation of the algorithm from Section IV; Nikolay A. Moldovyan had performed security estimation of the algorithm from Section IV; all authors had approved the final version

Acknowledgment

Research was commissioned by the public corporation Russian Railways.

References

[1] K. N. Griggs, O. Ossipova, C. P. Kohlios, A. N. Baccarini, E. A. Howson, and T. Hayajneh, "Healthcare blockchain system using smart contracts for secure automated remote patient monitoring," *Journal of Medical Systems*, vol. 42, no. 130, 2018.

[2] G. Zhang *et al.*, "Femto: Fair and energy-minimized task offloading for fog-enabled IoT networks," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4388–4400, 2018.

[3] Q. Xia, E. B. Sifah, K. O. Asamoah, J. Gao, X. Du, and M. Guizani, "MeDShare: Trust-less medical data sharing among cloud service providers via blockchain," *IEEE Access*, vol. 5, pp. 14757–14767, 2017.

[4] X. Yao, H. Kong, H. Liu, T. Qiu, and H. Ning, "An attribute credential based public key scheme for fog computing in digital manufacturing," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 4, pp. 2297–2307, 2019.

[5] H. Kaur, M. A. Alam, R. Jameel, A. K. Mourya, and V. Chang, "A proposed solution and future direction for blockchain-based heterogeneous medicare data in cloud environment," *Journal of Medical Systems*, vol. 42, no. 156, 2018.

[6] C. A. Shahnaz, U. Qamar, and A. Khalid, "Using blockchain for electronic health records," *IEEE Access*, vol. 7, pp. 147782–147795, 2019.

[7] P. W. Shor, "Polynomial-time algorithms for prime factorization and discrete logarithms on quantum computer," *SIAM Journal of Computing*, vol. 26, pp. 1484–1509, 1997.

[8] A. Ekert and R. Jozsa, "Quantum computation and Shor's factoring algorithm," *Reviews of Modern Physics*, vol. 68, pp. 733–752, 1996.

[9] J. A. Smolin, G. Smith, and A. Vargo, "Oversimplifying quantum factoring," *Nature*, vol. 499, no. 7457, pp. 163–165, 2013.

[10] F. Register. Announcing request for nominations for public-key post-quantum cryptographic algorithms. [Online]. Available: https://www.gpo.gov/fdsys/pkg/FR-2016-12-20/pdf/2016-30615.pdf

[11] Post-Quantum Cryptography: Digital Signature Schemes. [Online]. Available: https://csrc.nist.gov/Projects/pqc-dig-sig/standardization/call-for-proposals

[12] G. Alagic *et al.*, "Status report on the third round of the NIST post-quantum cryptography standardization process," *NIST Interagency/Internal Report (NISTIR)*, vol. 8309, 2020.

[13] N. A. Moldovyan and A. A. Moldovyan, "Digital signature scheme on the 2×2 matrix algebra," *Vestnik of Saint Petersburg University. Applied Mathematics, Computer Science, Control Processes*, vol. 3, 2021.

[14] N. A. Moldovyan and D. N. Moldovyan, "A novel method for developing post-quantum cryptoschemes and a practical signature algorithm," *Applied Computing and Informatics*, vol. 428, 2021.
[15] A. A. Moldovyan, D. N. Moldovyan, and N. A. Moldovyan, "A novel method for developing post-quantum digital signature algorithms on non-commutative associative algebras," *Information and Control Systems*, no. 1, pp. 44–53, 2022.
[16] J. Ding, "A Petzoldt current state of multivariate cryptography," *IEEE Security and Privacy Magazine*, vol. 15, no. 4, pp. 28–36, 2017.
[17] T. Matsumoto and H. Imai, "Public quadratic polynomial-tuples for efficient signature verification and message-encryption," *Advances in Cryptology*, pp. 419–453, 1988.
[18] J. Ding and D. Schmidt, "Rainbow, a new multivariable polynomial signature scheme," in *Proc. Conference on Applied* Cryptography *and Network Security*, 2005, vol. 3531, pp. 164–175.
[19] Q. Shuaiting, H. Wenbao, L. Yifa, and J. Luyao, "Construction of extended multivariate public key cryptosystems," *International Journal of Network Security*, vol. 18, no. 1, pp. 60–67, 2016.
[20] D. Jintai and S. Dieter. Multivariable public key cryptosystems. [Online]. Available: https://eprint.iacr.org/2004/350.pdf
[21] T. Yasuda and K. Sakurai, "A multivariate encryption scheme with rainbow," in *Proc. 18th Annual International Conference on Information Security and Cryptology*, 2015, vol. 9543, pp. 222–236.
[22] Rainbow Signature. One of three NIST post-quantum signature finalists. [Online]. Available: https://www.pqcrainbow.org/
[23] J. Ding and D. Schmidt, "Rainbow, a new multivariable polynomial signature scheme," in *Proc. Conference on Applied Cryptography and Network* Security, 2005, vol. 3531, pp. 164–175, Springer, Heidelberg.
[24] A. Petzoldt, S. Bulygin, and J. Buchmann, "Cyclic rainbow—A multivariate signature scheme with a partially cyclic public key," in *Proc. International Conference on Cryptology in India*, 2010, vol. 6498, pp. 33–48.
[25] A. Kipnis, J. Patarin, and L. Goubin, "Unbalanced oil and vinegar signature schemes," *Eurocrypt*, vol. 1592, pp. 206–222,1999.
[26] C. Clough, J. Baena, J. Ding, and B. Y. Yang, and M. Chen, "Square, a new multivariate encryption scheme," in *Proc. Cryptographers' Track at the RSA Conference*, vol. 5473, pp. 252–264, 2009.
[27] J. Patarin, N. T. Courtois, and L. Goubin, "FLASH, a fast multivariate signature algorithm," in *Proc. Cryptographers' Track at the RSA Conference*, 2001, p. 298.
[28] J. Porras, J. Baena, and J. Ding, "ZHFE, a new multivariate public key encryption scheme," *International Workshop on Post-Quantum Cryptography*, vol. 8772, pp. 229–245, 2014.
[29] GeMSS: A great multivariate short signature. [Online]. Available: https://www-polsys.lip6.fr/Links/NIST/GeMSS.html
[30] J. C. Faugére, "A new efficient algorithm for computing Gröbner basis," *J. Pure Appl. Algebra*, vol. 139, no. 1–3, pp. 61–88, 1999.
[31] J. C. Faugére, "A new efficient algorithm for computing Gröbner basis without to zero (*F*5)," in *Proc. the International Symposium on Symbolic and Algebraic Computation*, pp. 75–83, 2002.
[32] O. Billet and H. Gilbert, "Cryptanalysis of rainbow," in *Proc. International Conference on Security and Cryptography for Networks*, vol. 4116, 2006, pp. 336–347.
[33] J. Ding, B. Y. Yang, C. H. O. Chen, M. S. Chen, and C. M. Cheng, "New differential-algebraic attacks and reparametrization of rainbow," in *Proc. International Conference on Applied Cryptography and Network Security*, 2008, vol. 5037, pp. 242–257.
[34] O. Billet and G. M. Rat, "Cryptanalysis of the square cryptosystems," in *Proc. International Conference on the Theory and Application of Cryptology and Information Security*, 2009, vol. 5912, pp. 451–468.
[35] V. Dubois, P. A. Fouque, A. Shamir, J. Stern, "Practical cryptanalysis of SFLASH," in *Proc. Annual International Cryptology Conference*, 2007, vol. 4622, pp. 1–12.
[36] B. Y. Yang and J. M. Chen. TTS: Rank attacks in tame-like multivariate PKCs. [Online]. Available: http://eprint.iacr.org/2004/061
[37] N. A. Moldovyan, "Unified method for defining finite associative algebras of arbitrary even dimensions," *Quasigroups and Related Systems*, vol. 26, no. 2, pp. 263–270, 2018.