cryptography

MDPI

Article

Threshold Lattice-Based Signature Scheme for Authentication by Wearable Devices

Anton Leevik [1,*], Vadim Davydov [1] and Sergey Bezzateev [2]

[1] Faculty of Secure Information Technologies, ITMO University, 197101 Saint Petersburg, Russia; vadimdavydov@outlook.com
[2] Department of Information Security, State University of Aerospace Instrumentation, 190000 Saint Petersubrg, Russia; sergey.bezzateev@gmail.com
* Correspondence: anton.leevik@gmail.com

Abstract: This paper presents a new threshold signature scheme based on Damgaard's work. The proposed scheme allows for changing the message signature threshold, thereby improving the flexibility of the original Damgaard scheme. This scheme can be applied as a user authentication system using wearable devices. Based on the hardness of lattice problems, this scheme is resistant to attacks on a quantum computer, which is an advantage over the currently used multi-factor authentication schemes. The scheme's security relies on the computational complexity of the Module-LWE and Module-SIS problems, as well as the Shamir secret sharing scheme's security.

Keywords: digital signature; lattice theory; lattice-based cryptography; threshold signature; secret sharing scheme

check for updates

Citation: Leevik, A.; Davydov, V.; Bezzateev, S. Threshold Lattice-Based Signature Scheme for Authentication by Wearable Devices. *Cryptography* **2023**, *7*, 33. https://doi.org/10.3390/cryptography7030033

Academic Editor: Josef Pieprzyk

Received: 4 May 2023
Revised: 22 June 2023
Accepted: 28 June 2023
Published: 4 July 2023

Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Distributed systems are becoming very popular these days. To ensure the security of such systems, threshold cryptographic schemes are used, for instance, threshold encryption or threshold signature. A threshold signature (t, n) is a cryptographic digital signature scheme in which any t or more of n possible participants can sign a message, but a smaller number of participants are not capable of it. Each participant keeps a part of the private signature key with which they partially sign the message.

Threshold schemes have found their application for multi-factor authentication using wearable devices [1]. Nowadays, wearable devices have become very popular and are used by hundreds of millions people daily. Therefore, it is essential to make these devices secure. Since they have low memory, processing capabilities and power, it is feasible to use lightweight cryptography to protect communications [2].

Lightweight cryptography is a subfield of cryptography where algorithms are designed for resource-constrained devices. According to the NIST report [3], lightweight cryptoprimitives include hash functions, block ciphers, stream ciphers, and message authentication codes (MAC). However, this list cannot be considered exhaustive.

There are some works about cryptographic primitives for secure communication between wearables. In [4], authors analyze the feasibility of using cryptographic primitives for wearable devices such as bilinear pairings. The impact of using lightweight block and stream cipher algorithms on power consumption is reviewed in [5]. Several papers are devoted to the safety of wearable medical devices [6–8]. Most works use elliptic curves cryptography (ECC) to ensure secure communication.

At the same time, it is reasonable to consider using other cryptoprimitives for wearable devices to solve various tasks. For example, consider a multi-factor authentication system. As illustrated in Figure 1, user authentication is performed through wearable devices, such as a smartwatch or a smartphone. Here, there are n different wearable devices, such as smart glasses or smartwatches. The main idea is to authenticate in the system using t

different wearable devices which provide information from the sensors. The user decides which device to use in for authentication. Threshold signature authentication is used to enable user authentication in the absence of one of the devices and also will not allow an attacker who has taken possession of one of the devices to authenticate.

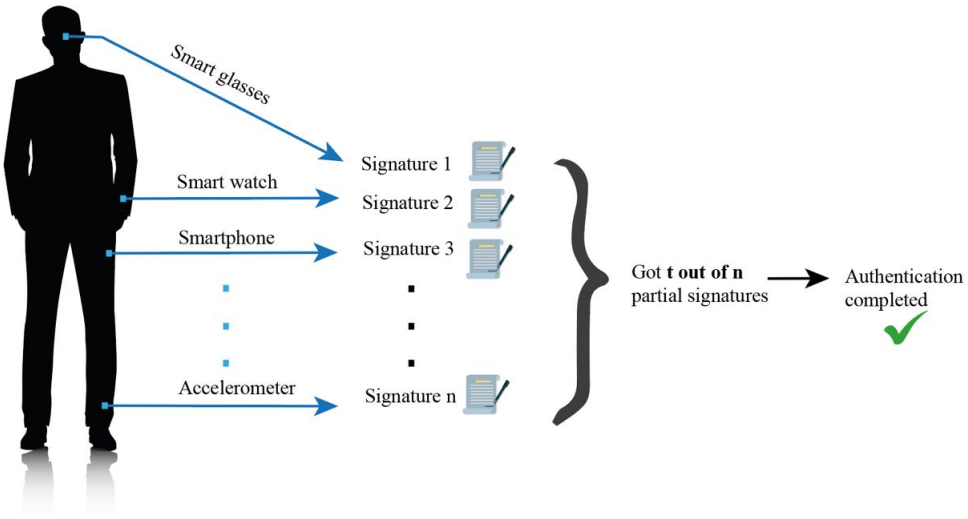

Figure 1. Authentication based on partial signatures of wearable devices.

Due to the advent of quantum computers and the invention of the quantum Shor's algorithm [9], the development of new post-quantum cryptographic schemes that will replace the existing ones has become an urgent task. Therefore, creating a post-quantum analog for the threshold signature scheme is also necessary.

Nowadays, there are five main mathematical constructions on which modern post-quantum cryptographic algorithms are based: error-correcting codes, isogenies, hash functions, multivariate equations, and lattices.

Code-based cryptography dates back to 1978 when the American scientist Robert McEliece presented a cryptosystem based on the syndrome decoding problem [10]. The first attempts to build a digital signature on the error-correcting codes belong to Alabbadi [11] and Wang [12]. However, it has been proven that such schemes are not secure [13]. The first secure algorithm in this area was the Courtois–Finiasz–Sendrier algorithm, published in 2001 [14]. However, in this scheme, there is the possibility of not signing the message the first time. Building a secure and efficient scheme based on error-correcting codes is still an area for improvement due to the inability to balance the sizes of keys, signatures, and the time spent on the signature.

Isogeny-based cryptography is a relatively new area—the first algorithm was presented in 2002 by Rostovtsev and Makhovenko [15]. Nowadays, there exist several signature algorithms, the main algorithms are CSI-FiSh [16] and SQISignHD [17]. Threshold variants of CSI-FiSh signature [18,19] were also presented. The main disadvantage of such algorithms is a long signature and key generation running time. However, the key sizes of such algorithms are the smallest compared with other post-quantum classes.

Hash-based cryptography is a special class of post-quantum cryptography where the construction of signature schemes on hash functions does not depend on complex mathematical and algorithmic problems in algebra or number theory. In 1979, Leslie Lamport published the concept of one-time signatures [20]. In the same year, Ralph Merkle described the MSS [21], the security of which is based on the security of the hash function used. Nowadays, the main algorithm is SPHINCS+ [22], although key sizes are very large. It is possible to use hash functions in many different applications. However, creating a threshold signature based on hash functions is impossible, and it requires an additional mathematical problem.

Multivariate cryptography dates back to 1988 when Matsumoto and Imai presented a cryptosystem that could be practically implemented [23]. However, this scheme was broken in 1995 by Patarin [24]. Nowadays, the most promising signature algorithm is Rainbow [25]. Several threshold signature schemes are based on multivariate equations [26,27]. The main issue of these schemes is large key sizes.

Lattice theory is also one of the promising areas of post-quantum cryptography. The first ideas were presented in 1997 by Ajtai, Dwork [28] and Goldreich, Goldwasser, and Halevi [29]. Nowadays, the most promising lattice-based signature algorithms are Falcon [30] and Crystals-Dilithium [31], which were presented in the NIST post-quantum algorithms competition.

As the review above showed, building an efficient signature based on error-correcting codes is hard. Signing on isogenies requires much time and may not apply to constrained devices. Hash-based cryptography cannot be applied for threshold signatures itself and requires additional use of a mathematical problem. Multivariate cryptography, compared with lattices, has larger key sizes. In this paper, we propose using lattice-based cryptography to build an effective threshold signature scheme.

Currently, several works are already offering lattice-based threshold signature schemes. One of the first threshold signatures on lattices can be considered in [32]. The authors present this work as an improvement of their previous threshold signature scheme based on error-correcting codes. The security of the previous scheme was based on the syndorme decoding problem, in the new work, the authors transformed the problem into an ISIS (inhomogeneous short independent solution) problem on a lattice. The CLRS scheme, conventionally named after its authors' initials, is an interactive threshold signature scheme. In this case, the signature creation algorithm is presented as an interactive proof protocol, where the "Prover" is the signer. The main disadvantage of this scheme is the large size of signatures. In [33], Bettaib and Sherk improve this algorithm by reducing the signature size.

The threshold signature scheme described in [34], called Feng's scheme, had an additional property, namely the ability to change the threshold required for signing a message. This scheme is based on NTRUSign [35], and its main disadvantage is the sequential signature of the message, which does not allow parallelizing the signing process. In [36], the authors propose a centralized threshold signature scheme. In [37], a threshold scheme is proposed, where the original message is divided into several blocks signed randomly. One of the most recently developed lattice-based threshold signature schemes presented in [38] deserves special attention. This scheme is based on the previously proposed lattice-based secret sharing scheme described in [39] by the same group of authors. The scheme's security is based on the Micciancio and Peikert function presented in [40], namely, on the SIS problem. The main drawbacks of this scheme are that the scheme is centralized and the secret sharing scheme is not verifiable, which means that an attacker can easily disrupt the process of signing a message by substituting the wrong part of the secret.

In [41], an anonymous and verifiable threshold signature scheme is presented, in which the private key is shared using a lattice-based threshold multi-stage secret sharing scheme. In [42], a universal approach was presented for generating a threshold signature based on the existing signature schemes using fully homomorphic encryption schemes, but this scheme is quite labor-intensive. One of the well-known paradigms for constructing signatures on lattices is the Fiat–Shamir with abortions paradigm, mentioned for the first time in the work of Lyubashevsky [43,44]. This paradigm is based on Schnorr's signature [45] and is used in a standardized signature algorithm [31].

In 2022, Damgaard [46] published a new (n, n) threshold signature scheme. This scheme is a two-round protocol based on the Fiat-Shamir with aborts paradigm. The protocol is a distributed version of the Dilithium-G signature scheme, used with the Baum commitment scheme [47]. This commitment scheme is additively homomorphic and allows for generating a commitment key with a trapdoor. Due to such properties of the commitment scheme and its security and resistance to quantum attacks, this distributed scheme

was built and proved to be secure. This scheme is based on the Module-LWE and Module-SIS tasks, and its theoretical security of UF-CMA (unforgeability against chosen-message attacks) is proven in the original work. The main disadvantage of the Damgaard scheme is the inability to change the message signature threshold. That is, only all users of the system can sign a message.

The scheme proposed in this paper extends the Damgaard scheme and adds a threshold change property. To implement this property, the Shamir secret sharing scheme [48] is used. However, it is also possible to use the secret sharing scheme on the Newton polynomial [49].

The rest of this paper is organized as follows. Section 2 gives some definitions in lattice theory. Section 3 shows the threshold signature scheme and the corresponding commitment scheme algorithms. Section 4 provides the security analysis of the scheme. Section 5 discusses the benefits and drawbacks of the proposed scheme. Section 6 concludes the paper.

2. Preliminaries

Definition 1 (Lattice [50]). *The set of all integer combinations of n linearly independent vectors* $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n \in \mathbb{R}^m$, *where* $n \leq m$, *is called a lattice. The vectors are called the basis of the lattice. Formally, it can be written as follows:*

$$L(\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n) = \{\sum_{i=1}^{n} \boldsymbol{b}_i \cdot x_i : x_i \in \mathbb{Z}\}. \tag{1}$$

The basis vectors can be represented as a matrix $\boldsymbol{B} = [\boldsymbol{b}_1, \ldots, \boldsymbol{b}_n]$, *where vectors are represented as vector-columns, and then the definition of the lattice looks like this:*

$$L(\boldsymbol{B}) = \{\boldsymbol{B} \cdot \boldsymbol{x} : \boldsymbol{x} \in \mathbb{Z}^n\}. \tag{2}$$

One of the important lattice invariants is the minimum distance. The minimum distance of a lattice L is the length of the shortest nonzero vector, denoted as λ_1:

$$\lambda_1 = \min_{\mathbf{v} \in L \setminus \mathbf{0}} \|\mathbf{v}\|. \tag{3}$$

Cryptographic schemes do not use classical integer lattices but use either q-ary lattices or special algebraic lattices.

Definition 2 (q-ary lattice [50]). *Given a matrix* $\boldsymbol{A} \in \mathbb{Z}_q^{m \times n}$, *for the given numbers* q, m, n, *two q-ary lattices can be defined:*

$$L_q(\boldsymbol{A}) = \{\boldsymbol{y} \in \mathbb{Z}^m : \boldsymbol{y} = \boldsymbol{A} \cdot \boldsymbol{x}, \boldsymbol{x} \in \mathbb{Z}^n\}, \tag{4}$$

$$L_q^{\perp}(\boldsymbol{A}) = \{\boldsymbol{y} \in \mathbb{Z}^m : \boldsymbol{A}^T \cdot \boldsymbol{y} \equiv \boldsymbol{0} \mod q\}. \tag{5}$$

The first lattice is given by a linear combination of the rows of the matrix **A**, and the second lattice is orthogonal modulo q to it. However, in order for the q-ary lattice to cover the entire Euclidean space, a special construction A [51] is used to construct the lattice.

Definition 3 (Construction A [51]). *Given a matrix* $\boldsymbol{A} \in \mathbb{Z}_q^{m \times n}$, *for the given numbers* q, m, n *we define a lattice* $L_q(\boldsymbol{A}) = \boldsymbol{A} \cdot \mathbb{Z}_q^n + q \cdot \mathbb{Z}^m$. *Alternatively, we can define it as*

$$L_q(\boldsymbol{A}) = \{\boldsymbol{y} \in \mathbb{Z}^m : \boldsymbol{y} = [\boldsymbol{A}|q\boldsymbol{I}_m] \cdot \boldsymbol{x}, \boldsymbol{x} \in \mathbb{Z}^{n+m}\}. \tag{6}$$

Let $R = \mathbb{Z}[X]/(X^N + 1)$ be a ring of polynomials modulo a polynomial of degree N, where N is a power of two, then $R_q = \mathbb{Z}_q[X]/(X^N + 1)$ is a ring of polynomials with

coefficients of $\{0, \ldots, q-1\}$. Module lattices are defined in a similar way as q-ary lattices. We define the necessary sets of polynomials:

- A set of keys S with the parameter η, consisting of polynomials with small coefficients: $S_\eta = \{x \in R : \|x\|_\infty \leq \eta\}$, where $\|x\|_\infty = \max_{0 \leq i \leq N-1} |x_i|$;
- A set C with parameter k consisting of binary and sparse polynomials: $C = \{c \in R_2 : \|c\|_1 = k\}$, where $\|c\|_1 = \sum_{0 \leq i \leq N-1} |c_i|$.

There are classical computational problems in lattice theory. The main problems are SVP (shortest vector problem) and CVP (closest vector problem). However, these problems are hard in the worst case [52–54] and can not be used in cryptography. Therefore, average-case hard problems were formulated, such as LWE (learning with errors) and SIS (short integer solution). In [55], the reduction from the LWE problem to the Gap-SVP problem was proved. The developed scheme is based on two average-case hard computational problems, namely, module learning with errors (M-LWE) and module short integer solution (M-SIS) [56–58].

These problems are based on special module lattices. A module is a special algebraic structure constructed over a ring that generalizes rings and vector spaces, and a module lattice, in turn, generalizes both arbitrary and ideal lattices (lattices constructed on the ideal in the polynomial ring). Let the matrix $\mathbf{B} \in R_q^{n \times n}$ of rank n be the basis of the module M, then the module M over the ring R_q is given by the following formula [57]:

$$M = \{\mathbf{B} \cdot \mathbf{x} : \mathbf{x} \in R_q^n\}. \tag{7}$$

In turn, modular lattices are defined as embeddings of the module vectors by coefficients in the $\mathbb{Z}^{n \cdot N}$, or canonical embeddings in $\mathbb{C}^{n \cdot N}$. The LWE problem for modules is defined as follows.

Definition 4 (M-LWE$_{n,m,q,\eta}$ [57]). *Given a matrix $\boldsymbol{A} \in R_q^{m \times n}$ and a vector $\boldsymbol{t} \in R_q^m$, it is required to find a vector $\boldsymbol{s} \in S_\eta^n$ such that $\boldsymbol{t} = \boldsymbol{A} \cdot \boldsymbol{s} + \boldsymbol{e}$, where the vector $\boldsymbol{e}$ is obtained from a discrete Gaussian distribution D_s^m with mathematical expectation 0 and standard deviation s.*

The discrete Gaussian distribution with mathematical expectation $\mathbf{v} \in R^m$ and with standard deviation s is defined as follows:

$$D_{\mathbf{v},s}^m(\mathbf{z}) = \frac{\rho_{\mathbf{v},s}(\mathbf{z})}{\rho_{\mathbf{v},s}(R^m)}, \tag{8}$$

where $\rho_{\mathbf{v},s}(\mathbf{z}) = e^{\left(\frac{-\pi\|\mathbf{x}-\mathbf{v}\|^2}{s^2}\right)}$ is a Gaussian function and $\rho_{\mathbf{v},s}(R^m) = \sum_{\mathbf{x} \in R^m} \rho_{\mathbf{v},s}(\mathbf{x})$. Let us define D_s^m as a discrete Gaussian distribution with mathematical expectations equal to 0.

If the rank of the basis of the module is equal to 1, then such a basis sets an ideal over the ring, and the M-LWE problem is now considered within the framework of ideal lattices; in the literature, such a problem is called Ring-LWE [59]. Another problem built on integer lattices, SIS, can also be defined on module lattices.

Definition 5 (M-SIS$_{n,m,q,B}$ [57]). *Let a random matrix $\boldsymbol{A} \in R_q^{m \times n}$ be given, it is required to find a nonzero vector $\boldsymbol{z} \in R_q^m$ such that $\|\boldsymbol{z}\| \leq B$ and $\boldsymbol{A} \cdot \boldsymbol{z} \equiv \boldsymbol{0} \mod q$, where $\|\boldsymbol{z}\| = \sqrt{\sum_{0 \leq i \leq m-1} z_i^2}$.*

According to the M-SIS problem definition, for signature validity in the proposed scheme, we define the upper bound B of $\|\mathbf{z}\|$, which is a signature vector. From [44] this bound is defined for parameter $\gamma > 1$ as follows:

$$B = \gamma\sigma\sqrt{mN}, \tag{9}$$

where $\sigma = \frac{s}{\sqrt{2\pi}}$ is a standard deviation of Gaussian function. Parameter γ is chosen such that the probability $\gamma^{mN} e^{mN(1-\gamma^2)/2}$ is negligible.

The (t, n) threshold scheme that we use in our scheme was proposed by Shamir in 1979 [48]. In his work, he gives it the following definition:

Definition 6 ((t, n) threshold scheme [48]). *Let D be secret data, our goal is to divide D into n pieces $D_1, \ldots, D_n$ in such a way that:*

1. *Knowledge of any t or more D_i pieces makes D easily computable;*
2. *Knowledge of any $t-1$ or fewer D_i pieces leaves D completely undetermined (in the sense that all its possible values are equally likely).*

Such a scheme is called the (t, n) threshold scheme.

Shamir's work presents a mechanism for dividing a secret into n parts and assembling it from t or more parts. The Lagrange interpolation formula is used for this. This scheme is centralized; that is, the dealer who owns the secret divides it between the participants, who are gathering together (or sending their parts to the dealer) to collect the secret. The scheme consists of the following algorithms:

- Secret sharing.
  Let p is a prime number such that $p > D$, the dealer builds a ring of polynomials $\mathbb{Z}_p[x]$ on it and generates a polynomial $f(x)$ of degree $t-1$ as follows:

$$f(x) = a_{t-1}x^{t-1} + a_{t-2}x^{t-2} + \cdots + a_1 x + D, \tag{10}$$

  where $a_i \in \mathbb{Z}_p$.
  Let each user have their unique identifier uid_i, such that there are no two uid_i and uid_j such that $uid_i \equiv uid_j \mod p$; the dealer sends each participant his share of the secret as the value of the previously generated polynomial $f(x)$ at the point of his uid_i, calculated as follows:

$$y_i = f(uid_i) \mod p. \tag{11}$$

  Thus, each participant eventually gets a pair (uid_i, y_i), which is his part of the secret.
- Recovering a secret.
  In order to recover the secret, a group of t participants gathers together and calculates a polynomial using the Lagrange interpolation formula. Each user computes the Lagrange coefficient, using uid_i of each user, which is known by the following formula:

$$l_i = (-1)^{t-1} \prod_{j \neq i, 1 \leq j \leq t} \frac{uid_j}{uid_i - uid_j}. \tag{12}$$

  Next, the polynomial $f(x)$ is restored using the following formula:

$$f(x) = \sum_{1 \leq i \leq t} y_i \cdot l_i. \tag{13}$$

  The resulting polynomial $f(x)$ as a free term will contain a secret value D, i.e., the group of participants successfully obtains the secret data.

3. Threshold Lattice-Based Signature Scheme

As mentioned earlier, the proposed scheme is based on the work of Damgaard [46]. The paper also uses a lattice-based commitment scheme with a trapdoor, presented in [47].

Commitment schemes are used when there is a need to fix some values at the current stage without disclosing them. The received commitment value is disclosed. When the moment comes, and the values are revealed, anyone can ensure they are not being deceived, and indeed the correct values have been used to create a commitment in the past.

In threshold signature, the commitment scheme is used for the scheme's security. After all, if users sent messages to each other without commitments, then an attacker who compromised one of the users would be able to choose parameters based on the received messages and send messages to other users in such a way that it is possible to

find out the other users' private keys or forge the signature. However, when using the commitment scheme, users first send each other commitments and then the values that they have calculated. In this case, the attacker cannot select such parameters that allow him to break the system if the commitment scheme is secure and unbreakable. For this purpose, this work also uses a lattice-based commitment scheme because breaking the commitment scheme will completely violate the system's security. However, this commitment scheme can also generate a trapdoor corresponding to the commitment key, which allows one to calculate the randomness of a commitment and the corresponding message. This property of the commitment scheme will be used to prove the security of the signature scheme.

The trapdoor commitment scheme consists of the following algorithms:

1. *Parameter setting.* Receives the security parameter λ, which defines the security level of the scheme, as input and returns the parameters (q, N, k, l, w, η) [47].
2. *Key generation.* Generates the commitment key *ck*, consisting of matrix $\hat{\mathbf{A}} \in R_q^{2\times(l+2w)}$, which is defined as follows:

$$\hat{\mathbf{A}} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1(l+2w)} \\ 0 & 1 & a_{23} & \dots & a_{2(l+2w)} \end{bmatrix}, \tag{14}$$

   where $a_{ij} \in R_q$ and a_{11} is invertible in R_q.
3. *Commitment generation.* Receives a value $x \in R_q$ as input, randomly calculates $\mathbf{r} \leftarrow D_s^{l+2w}$, where $\|\mathbf{r}\| \leq B$, and returns the commitment $\mathbf{f} \in R_q^2$:

$$\mathbf{f} = \hat{\mathbf{A}} \cdot \mathbf{r} + \begin{bmatrix} 0 \\ x \end{bmatrix}. \tag{15}$$

   It is known from [47] that the commitment scheme has the binding property; that is, it is hard for a published commitment $\mathbf{f}$, obtained by the vector $\mathbf{r}$ and the value x, to find the vector $\mathbf{r}'$ and the value x' for which $\mathbf{f}' = \mathbf{f}$ since it reduces to solving the Ring-SIS problem, which is a hard problem. It is also proved in [47] that the commitment scheme has the hiding property since the distribution $\hat{\mathbf{A}} \cdot D_s^{l+2w}$ is close to uniform.
4. *Commitment opening.* Receives a commitment, a value $x \in R_q$, and a random vector $\mathbf{r}$ as input and checks that $\|\mathbf{r}\| \leq B$ and the Equation (15) is being fulfilled.
5. *Key generation with a trapdoor.* Generates the matrix $\bar{\mathbf{A}}$ according to (14) and randomly chooses a trapdoor, *td*, which is equal to a matrix $\mathbf{R} \leftarrow D_s^{l\times 2w}$. Then, the commitment key *tck* is formed as follows $tck = \hat{\mathbf{A}} = [\bar{\mathbf{A}}|\mathbf{G} - \bar{\mathbf{A}}|\mathbf{R}]$, where $\mathbf{G} \in R^{2\times 2w}$ is a gadget matrix, which is defined as follows:

$$\mathbf{G} = \begin{bmatrix} 1 & 2 & \dots & 2^{w-1} & 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 & 1 & 2 & \dots & 2^{w-1} \end{bmatrix}. \tag{16}$$

6. *Commitment generation with a trapdoor.* Randomly chooses a vector $\mathbf{f} \in R_q^2$ and outputs as a commitment.
7. *Equivocation algorithm.* Uses the trapdoor *td* and the Micciancio–Peikert algorithm [40] in order to generate a vector $\mathbf{r}$ from a discrete Gaussian distribution on the coset of the lattice $\Lambda_{\mathbf{u}}^{\perp}(\hat{\mathbf{A}})$, which is defined as follows:

$$\Lambda_{\mathbf{u}}^{\perp}(\hat{\mathbf{A}}) = \{\mathbf{z} \in R^{l+2w} : \hat{\mathbf{A}} \cdot \mathbf{z} \equiv \mathbf{u} \bmod q\}, \tag{17}$$

   where $\mathbf{u} = \mathbf{f} - \begin{bmatrix} 0 \\ x \end{bmatrix}$.

Next, we describe the threshold signature scheme itself. It includes the following algorithms:

1. *Parameters setting.* Having received the security parameter λ as input, the public parameters of the system are generated, namely, the rings of polynomials, the public matrix rank l and dimension k, the sets S and C, the parameters of distributions,

the boundary B for the length of the signature vector, as well as random oracles $H_0 : \{0,1\}^* \to C$, $H_1 : \{0,1\}^* \to \{0,1\}^{l_1}$, $H_2 : \{0,1\}^* \to \{0,1\}^{l_2}$ and $H_3 : \{0,1\}^* \to S_{ck}$ [46].

2. *Key generation.* After initializing public parameters, keys are generated, consisting of two phases: matrix generation and key pair creation. All subsequent steps of the algorithm are performed by each P_i user of the system, where $i \in \{1, \ldots, n\}$ and n is the total number of users.
   (a) Matrix generation
      i. A random matrix $\mathbf{A}_i \leftarrow R_q^{k \times l}$ is calculated and a commitment $g_i = H_1(\mathbf{A}_i, i)$ is generated and sent to other users.
      ii. After receiving all g_j for each $j \neq i$, $\mathbf{A}_i$ matrix is sent out for each one.
      iii. After obtaining all $\mathbf{A}_j$ matrices for each $j \neq i$, the equalities $g_j = H_1(\mathbf{A}_j, j)$ are checked. If at least one equality is not met, then an ABORT is sent, otherwise a public matrix $\bar{\mathbf{A}} = [\mathbf{A}|\mathbf{I}] \in R_q^{k \times (k+l)}$ is set, where $\mathbf{A} = \sum_{1 \leq j \leq n} \mathbf{A}_j$.
   (b) Generation of a key pair
      i. A secret vector $\mathbf{s}_i \leftarrow S_\eta^{l+k}$ is randomly selected, and a part of the public key is calculated: $\mathbf{t}_i = \bar{\mathbf{A}} \cdot \mathbf{s}_i$. A commitment $g_i' = H_2(\mathbf{t}_i, i)$ is generated and sent to other users.
      ii. After receiving all g_j for each $j \neq i$, the vector $\mathbf{t}_i$ is sent to other users.
      iii. After obtaining all vectors $\mathbf{t}_j$ for each $j \neq i$, the equalities $g_j' = H_2(\mathbf{t}_j, j)$ are checked. If at least one equality is not met, then an ABORT is sent, otherwise a public key $\mathbf{t} = \sum_{1 \leq j \leq n} \mathbf{t}_j$ is set.

   If the protocol does not return ABORT, then each user, P_i, gets $(sk_i, pk) = (\mathbf{s}_i, (\mathbf{A}, \mathbf{t}))$.

3. *Secret sharing.* To separate the secret, the Shamir secret sharing scheme is used [48]. The P_i user has a unique own uid_i and knows the uid_j of other users. Then it performs the following actions:
   (a) Generates $k + l$ polynomials f_z^i ($z \in \{1, \ldots, k + l\}$; i is an index of P_i) of degree $(t - 1)$, where free terms are specified as entries of secret vector $\mathbf{s}_i$.
   (b) For each user P_j, including himself, the user P_i generates a vector consisting of polynomials generated in advance with uid_j values substituted in them, which is a vector $\mathbf{f}_j^i = (f_1^i(uid_j), f_2^i(uid_j), \ldots, f_{k+l}^i(uid_j))$, and sends this vector only to user P_j.
   (c) After receiving all the vectors $\mathbf{f}_i^j$ for each $j \neq i$, user P_i calculates his secret key share $\mathbf{x}_i = \sum_{1 \leq j \leq n} \mathbf{f}_i^j$, with which he will then carry out the signature procedure.

   As it can be seen, users, in this case, perform distributed secret sharing; that is, they get the share of a common secret without calculating the secret polynomial directly. This approach differs from the classical one when the dealer forms a secret polynomial and distributes shares of the secret to users.

4. *Signature generation.* For signing message μ t users are selected. Let the users $\{P_i, i \in \{1, \ldots, t\}\}$ be selected. Each P_i receives a unique session ID (sid) and a message μ that needs to be signed. The user checks that the sid has not been used before and calculates locally the key for the commitment scheme $ck = H_3(\mu, pk)$. A new random oracle function is also used for the signature procedure $H_4 : \{0,1\} \to \{0,1\}^{l_4}$. Next, the user performs the following actions.
   (a) Randomly selects a vector $\mathbf{y}_i \leftarrow D_s^{l+k}$ and calculates $\mathbf{w}_i = \bar{\mathbf{A}} \cdot \mathbf{y}_i$.
   (b) Calculates the commitment $com_i = Commit_{ck}(\mathbf{w}_i, r_i)$, where $r_i \leftarrow D(S_\eta)$, and sends it to all other users.

(c) After receiving all com_j calculates $com = \sum\limits_{1 \le j \le t} com_j$.

(d) Next, the user calculates the Lagrange coefficient

$$l_i = (-1)^{t-1} \prod_{j \neq i, 1 \le j \le t} uid_j (uid_i - uid_j)^{-1}$$

and the value $\bar{\mathbf{y}}_i = \mathbf{y}_i l_i^{-1}$ by modulo q.

(e) Then receives the challenge $c = H_0(com, \mu, pk)$ and calculates the partial signature $\mathbf{z}_i = c\mathbf{x}_i + \bar{\mathbf{y}}_i$. For the next step user also computes vector $\mathbf{z}'_i = c\mathbf{s}_i + \mathbf{y}_i$.

(f) For the received value $\mathbf{z}'$, the user checks that $\|\mathbf{z}'\| < B$; if the condition is not met, then the user sends out RESTART. If the condition is met, then the user with the probability

$$\min\left(1, D_s^{l+k}(\mathbf{z}'_i) / (M \cdot D_{c\mathbf{s}_i, s}^{l+k}(\mathbf{z}'_i))\right) \tag{18}$$

generates $g''_i = H_4(\mathbf{z}_i, r_i)$ and sends out it, or otherwise sends RESTART and returns to step (a). This rejection sampling technique is used to counter statistical attacks that can restore the secret $c\mathbf{s}_i$ vector by obtaining multiple $\mathbf{z}_i$.

(g) After obtaining all g''_j for each $j \neq i$, the partial signature $(\mathbf{z}_i, r_i)$ is sent to other users.

(h) After receiving all the partial signatures $(\mathbf{z}_j, r_j)$, checks that $g''_j = H_4(\mathbf{z}_j, r_j)$, and if all conditions are met, calculates the values $\mathbf{z} = \sum\limits_{1 \le j \le t} \mathbf{z}_j \cdot l_j$ and $r = \sum\limits_{1 \le j \le t} r_j$. Then, calculates $\mathbf{w} = \bar{\mathbf{A}}\mathbf{z} - c\mathbf{t}$, checks that $\|\mathbf{z}\| \le t \cdot B$ and $Open_{ck}(com, r, \mathbf{w}) = 1$. If errors occur, then send ABORT.

If the protocol is not interrupted, the signature $\sigma = (com, \mathbf{z}, r)$ will be received at the end of the protocol.

5. *Signature verification.* Having received the message μ, signature σ and public key pk, the commitment key is generated $ck = H_3(\mu, pk)$, and the challenge is calculated $c = H_0(com, \mu, pk)$ and restored $\mathbf{w} = \bar{\mathbf{A}}\mathbf{z} - c\mathbf{t}$. The signature is accepted if $\|\mathbf{z}\| \le t \cdot B$, and $Open_{ck}(com, r, \mathbf{w}) = 1$.

Let us show the correctness of the scheme. Let $\|\mathbf{z}\| \le t \cdot B$, then

$$\bar{\mathbf{A}} \cdot \mathbf{z} = \bar{\mathbf{A}} \cdot \sum_{1 \le j \le t} \mathbf{z}_j \cdot l_j = \sum_{1 \le j \le t} \bar{\mathbf{A}} \cdot \mathbf{x}_j \cdot c \cdot l_j + \frac{\bar{\mathbf{A}} \cdot \mathbf{y}_j \cdot l_j}{l_j}. \tag{19}$$

According to the Shamir secret sharing scheme $\sum_{1 \le j \le t} \mathbf{x}_j \cdot l_j = \sum_{1 \le j \le n} \mathbf{s}_j = \mathbf{s}$, thus,

$$\sum_{1 \le j \le t} \bar{\mathbf{A}} \cdot \mathbf{x}_j \cdot c \cdot l_j + \frac{\bar{\mathbf{A}} \cdot \mathbf{y}_j \cdot l_j}{l_j} = \bar{\mathbf{A}} \cdot \mathbf{s} \cdot c + \sum_{1 \le j \le t} \mathbf{w}_j = \mathbf{t} \cdot c + \mathbf{w}. \tag{20}$$

4. Security

To prove the security of the scheme, we introduce the concepts of the forking lemma proposed in [60].

Lemma 1 (Forking lemma [60]). *Let (G, S, V) be a digital signature algorithm with a security parameter k. Let $\mathcal{A}$ be a probabilistic, polynomial–time Turing machine whose input data are public. We will denote as Q the number of requests that $\mathcal{A}$ can send to a random oracle. Suppose that during time T, machine $\mathcal{A}$ can generate a valid signature $(m, \sigma_1, h, \sigma_2)$ with probability $\epsilon \ge 7Q/2^k$. Then there is another Turing machine that controls machine $\mathcal{A}$, generating two valid signatures $(m, \sigma_1, h, \sigma_2)$ and $(m, \sigma_1, h', \sigma'_2)$ such that $h \neq h'$, in time $T' \le 84480TQ/\epsilon$.*

Now we formulate a theorem about the security of the system.

Theorem 1. *Let us assume that the trapdoor commitment scheme is secure, as well as additively homomorphic. Then, if there is an algorithm $\mathcal{A}$ such that it can forge the signature of the system with a non-negligible probability ϵ, then there is an algorithm $\mathcal{B}$ such that it can solve the M-SIS problem with a non-negligible probability ϵ.*

Proof. The proof is based on a random oracle model and uses definitions of some oracles from Damgaard's work [46]. Algorithm $\mathcal{B}$ simulates the work of an honest user of the system, and algorithm $\mathcal{A}$ is a signature forgery algorithm and is an adversary, and it controls $t-1$ users of the system. Adversary $\mathcal{A}$ can send to $\mathcal{B}$ requests for the use of random oracles H_0, H_1, H_2, H_3, H_4, as well as requests for the signature generation of the message.

First, we define the simulation of the random oracles H_0, H_1, H_2, H_3, H_4. For each of the oracle, we create tables $HT_0, HT_1, HT_2, HT_3, HT_4$, in which values are stored as a key-value pair, and for oracle H_3, we also create a TDT table in which trapdoors for the created commitment keys are stored. The tables are supplemented when referring to the oracles. The simulation of the oracles H_0, H_1, H_2 can be generally described as generating random values for input data and setting them into the corresponding tables. The simulation of a random oracle H_3 consists in generating a commitment key, also with a certain probability ω, and a trapdoor is generated for the obtained commitment key, which is placed in the TDT table, and the commitment key is placed in the HT_3 table. A more detailed description of the simulation of random oracle algorithms is described in Damgaard's work [46]. Here, we detail the simulation of the random oracle H_4.

$H_4(x)$:

1. Split the incoming value x into $(\mathbf{z}, r)$;
2. If $HT_4[\mathbf{z}, r] = \perp$ then set $HT_4[\mathbf{z}, r] \leftarrow \{0,1\}^{l_4}$;
3. Return $HT_4[\mathbf{z}, r]$.

Now let us describe the simulation of the algorithms for generating the key and signing the message by $\mathcal{B}$. Let the input of algorithm $\mathcal{B}$ be a matrix $\mathbf{A}'$, for which it is required to solve the M-SIS problem. For the key generation algorithm, $\mathcal{B}$ takes the matrix $\mathbf{A}'$ and presents it in the following form: $\mathbf{A}' = [\mathbf{A}|\mathbf{t}|\mathbf{I}]$, where $\mathbf{A}$ is used as the open matrix of the whole system, and $\mathbf{t}$ is used as the public key of the system.

Since all requests to access a random oracle are displayed in the corresponding tables, when generating a public matrix and a public key, algorithm $\mathcal{B}$ takes data from the tables HT_i and restores the values of partial public keys and matrices of each user and forms its partial matrix and partial key according to the following formulas:

$$\mathbf{A}_t = \mathbf{A} - \sum_{j=1, j\neq t}^{n} \mathbf{A}_j, \tag{21}$$

$$\mathbf{t}_t = \mathbf{t} - \sum_{j=1, j\neq t}^{n} \mathbf{t}_j, \tag{22}$$

where $\mathbf{A}_t$ and $\mathbf{t}_t$ are the matrix and the public key of $\mathcal{B}$, respectively. Thereby, parameters for which $\mathcal{B}$ wants to solve the M-SIS problem are set as general parameters of the system. A more detailed description of these algorithms can be found in Damgaard's work [46].

Now let us describe a signing simulation algorithm. Before $\mathcal{B}$ starts generating the signature, $\mathcal{B}$ locally calculates the commitment key without using the trapdoor $ck \leftarrow S_{ck}$. Next, $\mathcal{B}$ receives on input a unique session ID sid and a message μ, which has to be signed. The user checks that the sid has not been used before and calculates locally the commitment key $tck \leftarrow H_3(\mu, pk)$. If $TDT[\mu, pk] = \perp$; that is, instead of generating a key with a trapdoor, the previously generated key ck was obtained, then the signature generation process ends with an error. Otherwise, it receives a trapdoor for the newly generated key $td \leftarrow TDT[\mu, pk]$. Next, $\mathcal{B}$ performs the following actions.

1. The commitment $com_t \leftarrow TCommit_{tck}(td)$ is calculated and sent to other users.

2. After receiving all com_j for each $j \in [t-1]$, the message signature is calculated as follows:
   (a) $com = \sum_{j\in[t]} com_j$ is set.
   (b) Challenge $c \leftarrow H_0(com, \mu, pk)$ is calculated.
   (c) $g_t'' \leftarrow \{0,1\}^{l_4}$ is generated randomly and sent to other users.
   (d) After receiving all g_j'' for each $j \in [t-1]$, the following actions are performed:
      i. The HT_4 table is searched for values $(\mathbf{z}_1, r_1), (\mathbf{z}_2, r_2), \ldots, (\mathbf{z}_{t-1}, r_{t-1})$ according to the obtained g_j''.
      ii. Then vector $\mathbf{z} \leftarrow D_s^{l+k}$ is generated and the Lagrange coefficients l_i are calculated.
      iii. The vector $\mathbf{z}_t' = \mathbf{z} - \sum_{j=1}^{t-1} \mathbf{z}_j \cdot l_j$ is calculated, and then the vector of partial signature $\mathbf{z}_t = \mathbf{z}_t' \cdot l_t^{-1}$ is obtained.
      iv. Next, the vector $\mathbf{w} = \hat{\mathbf{A}}\mathbf{z} - c\mathbf{t}$ is calculated and with the trapdoor td, and the value of randomness $r \leftarrow Eqv_{tck}(td, com, \mathbf{w})$ is obtained.
      v. The value of $r_t = r - \sum_{j=1}^{t-1} r_j$ is obtained, using the property of homomorphism by addition of the commitment function.
      vi. If $HT_4[\mathbf{z}_t, r_t] = \perp$, then signature generation fails, otherwise $HT_4[\mathbf{z}_t, r_t] = g_t''$ is set, and a partial signature $(\mathbf{z}_t, r_t)$ is sent with probability $1/M$. Otherwise, RESTART is sent, and the algorithm returns to step 1.
3. After receiving all partial signatures $(\mathbf{z}_j, r_j)$ for each $j \in [t-1]$, the final signature of the message is formed:
   (a) It is checked that $g_j'' = H_4(\mathbf{z}_j, r_j)$. If all the equalities are met, then the values $\mathbf{z} = \sum_{j\in[t]} \mathbf{z}_j \cdot l_j$ and $r = \sum_{j\in[t]} r_j$ are calculated.
   (b) Next, the value $\mathbf{w} = \hat{\mathbf{A}}\mathbf{z} - c\mathbf{t}$ is calculated and it is checked that $\|\mathbf{z}\| \leq tB$ and $Open_{ck}(com_j, r_j, \mathbf{w}) = 1$. If one of the checks fails, an ABORT is sent.

If the simulation algorithm is not interrupted, the output is the final signature $(com, \mathbf{z}, r)$. Thus, the interface of interaction between the adversary $\mathcal{A}$ and algorithm $\mathcal{B}$ was configured, resulting in a valid signature for the message.

Suppose now adversary $\mathcal{A}$ has made a certain number of signature and hash requests to $\mathcal{B}$ and issued a signature forgery $(com^*, \mathbf{z}^*, r^*)$ for the message μ^*; then, algorithm $\mathcal{B}$ performs the following steps:

1. If $\mu^* \in Mset$, where $Mset$ is the set of messages for which the adversary $\mathcal{A}$ requested a signature from $\mathcal{B}$, then the algorithm $\mathcal{B}$ returns $\perp$.
2. Next, $\mathcal{B}$ calculates $ck^* \leftarrow H_3(\mu^*, pk)$ and $c^* \leftarrow H_0(com^*, \mu^*, pk)$.
3. If $Open_{ck}(com^*, r^*, \hat{\mathbf{A}}\mathbf{z} - c\mathbf{t}) \neq 1$ or $\|\mathbf{z}^*\| > t \cdot B$, then $\mathcal{B}$ returns $\perp$.
4. If $TDT[\mu^*, pk] \neq \perp$, that is, there was a request to generate a commitment key and a trapdoor for the message μ^*, then $\mathcal{B}$ also returns $\perp$.
5. If the signature successfully passes checks 1-4, then algorithm $\mathcal{B}$ returns $(com^*, \mathbf{z}^*, r^*, \mu^*, ck^*)$.

By the forking lemma, let $\mathcal{B}$ return two signature forgeries for the message μ^*, namely $(com^*, \mathbf{z}^*, r^*, \mu^*, ck^*)$ and $(com', \mathbf{z}', r', \mu^*, ck')$. It can be immediately noted that $ck^* = ck'$, since the other commitment keys are discarded by algorithm $\mathcal{B}$. The values of com^* and com' are also equal, but the values of the challenges c^* and c' are not equal. Since the signatures are valid, we obtain

$$Open_{ck}(com^*, r^*, \hat{\mathbf{A}}\mathbf{z}^* - c^*\mathbf{t}) = Open_{ck}(com^*, r', \hat{\mathbf{A}}\mathbf{z}' - c'\mathbf{t}) = 1. \quad (23)$$

If $\hat{\mathbf{A}}\mathbf{z}^* - c^*\mathbf{t} \neq \hat{\mathbf{A}}\mathbf{z}' - c'\mathbf{t}$, then the binding property of the commitment scheme on the key ck is violated, which cannot be from the problem condition (since the commitment scheme is safe). Therefore, $\hat{\mathbf{A}}\mathbf{z}^* - c^*\mathbf{t} = \hat{\mathbf{A}}\mathbf{z}' - c'\mathbf{t}$. Rewriting this equation, we obtain

$$[\mathbf{A}|\mathbf{I}|\mathbf{t}]\begin{bmatrix} \mathbf{z}^* - \mathbf{z}' \\ c^* - c' \end{bmatrix} = 0. \quad (24)$$

Since the matrix $[\mathbf{A}|\mathbf{t}|\mathbf{I}]$ was submitted to the input of algorithm $\mathcal{B}$, and the vector $\begin{bmatrix} \mathbf{z}^* - \mathbf{z}' \\ c^* - c' \end{bmatrix}$ is small, we found a solution to the M-SIS problem. Thus, the theorem is proved. □

Based on the proof of this theorem, we can say that the developed threshold scheme is UF-CMA secure. In addition, it is necessary to consider the classical vectors of attacks on lattice-based signature schemes, for example, an attack on a lattice using basis reductions. To counteract these attacks, it is required to select parameters for the system for which resistance to such attacks has been proven, that is, for example, for which the BKZ algorithm for a polynomial approximating factor works in exponential time. It is recommended to take the NIST parameters proposed for the CRYSTALS-Dilithium scheme [31], since the scheme proposed in this paper is based on this signature scheme.

5. Discussion

In this section, the effectiveness of the developed scheme is analyzed, and its quantitative indicators are evaluated. The signature generation and verification consist of multiplying and adding by modulo q and multiplying polynomials in a polynomial ring. These operations are not expensive, and with the fast Fourier transform, they are calculated quickly enough. Therefore this scheme can be used for devices with a limitation on the processor clock frequency. For example, the Dilithium signature [61], built on the same paradigm as the presented scheme, uses 508K and 175K CPU cycles, respectively, for the signature generation and verification processes.

The proposed scheme was implemented using Python with Sagemath to demonstrate the efficiency and high operational speed of calculations. Algorithms of key generation, secret sharing, signature generation, and verification were tested on the same device for different security levels, defined by NIST [31]. Table 1 shows the execution time of each stage of the scheme for different security levels.

Table 1. Experimental data on the running time of key generation, secret sharing, signature generation, and verification for different security levels.

Security Level	Algorithm	Execution Time, ms
2	Key generation	22.1
	Secret sharing	1.1
	Signature generation	20.4
	Signature verification	2.7
3	Key generation	27.8
	Secret sharing	1.5
	Signature generation	26.5
	Signature verification	2.8
5	Key generation	37.9
	Secret sharing	1.8
	Signature generation	37.8
	Signature verification	4.0

As seen from Table 1, the time spent on key generation, secret sharing, signature generation, and signature verification operations does not exceed 40 ms for the latest security level, and the time spent on signature verification and secret sharing is less than

5 ms. The results obtained during the experiments confirm the high speed of algorithm operations in lattice theory, which positively distinguishes this area from other areas of post-quantum cryptography. In addition, it should be said that the speed of calculations is important in distributed systems, since such systems should process a large amount of information quickly.

However, the scheme has several shortcomings that must be eliminated in future research. First, it requires intensive communication between users, which can negatively affect network congestion. Second, like other lattice-based schemes, the signature size is significant, requiring additional storage and transmission resources. The stored and transmitted data sizes were calculated for the recommended parameters from [31,47] and are presented in Table 2. As can be seen from Table 2, the data sizes are really large in comparison with the classical threshold elliptic curve digital signature algorithm (ECDSA) scheme [62]. For example, the amount of data transferred for one user is about 15 kilobytes, although the scheme on ECDSA requires about 3 kilobytes per user [62]. This is a consequence of using recommended parameters that provide a 128-bit security level. This is a disadvantage of the system itself, as well as of lattice-based post-quantum schemes in general. Therefore, reducing the size of keys is an urgent task for all lattice-based systems.

Table 2. Stored and transmitted data sizes.

Parameter	Actual Size of Proposed Scheme, Bytes	Actual Size of tECDSA Scheme, Bytes
Partial signature size	7360	64
Signature size	11,775	64
Secret data size	13,247	32
Size of transmitted data by signature generation	$15,700 \cdot t$	$3300 \cdot t$

It is worth noting that the scheme proposed in this paper is more efficient regarding data sizes than other threshold lattice-based signature schemes. In Table 3, the sizes of partial and full signatures, secret data, as well as transmitted data for various studied schemes are shown, and estimates for the developed scheme are also given. The sizes of stored and transmitted data for each of the schemes were calculated taking into account the NIST's recommended parameters.

Table 3. Stored and transmitted data sizes for the observed schemes and comparison with the proposed scheme.

Parameter	CLRS Scheme [32]	Feng's Scheme [34]	Choi Scheme [37]	PET Scheme [38]	Proposed Scheme
Partial signature size, kB	451	1.8	$0.8 \cdot N$	40.5	7.2
Signature size, kB	$451 \cdot t$	1.8	$0.8 \cdot N \cdot t$	60.4	11.5
Secret data size, kB	128	3.5	1,081,600	275,808	12.9
Size of transmitted data by signature generation, kB	$451 \cdot t$	$1.8 \cdot t$	$0.8 \cdot N \cdot t$	$40.5 \cdot t$	$15.3 \cdot t$

As seen from Table 3, the developed scheme has one of the best indicators among the schemes considered. The CLRS scheme and the PET scheme exceed the proposed scheme in terms of data sizes in all indicators by several times. The Choi scheme greatly exceeds the size of the secret data, and despite the initially small size of the signature, it increases rapidly with an increasing number of users in the system and the size of the threshold t. Therefore, starting with certain parameters, this scheme will greatly lose to the proposed scheme. However, Feng's scheme surpasses all schemes in quantitative parameters. Due to

the centralization of Feng's scheme and the fact that the original secret is restored during the signature generation process, this scheme cannot be used in distributed systems from a security point of view. Thus, the developed scheme is the best of the presented schemes regarding quantitative indicators.

It is worth noting that since Shamir's secret sharing scheme is not a verified secret sharing scheme, an attacker can seize control of one of the nodes and send incorrect messages to other users, which makes it impossible to form a common signature. Therefore, to eliminate the third drawback, a special system is required to verify the correctness of partial signatures and block unwanted participants.

6. Conclusions

The threshold signature scheme developed in this paper is an improvement over the Damgaard scheme [46]. However, as with other post-quantum threshold schemes, this scheme has certain drawbacks, such as the large size of the signature and transmitted data. Despite these disadvantages, the scheme provides significant advantages, including scalability and resistance to attacks on quantum computers. This makes it a valuable tool for protecting users' private keys in distributed systems and providing multi-factor authentication for wearable devices.

It is worth noting that data compactness is a crucial factor in distributed systems. As a result, the large size of signatures generated using this scheme can significantly reduce the efficiency of such systems. Therefore, it is important to continue exploring ways to improve the effectiveness of the proposed scheme to solve this problem. Future research should focus on developing methods to reduce the size of signatures while maintaining the same level of security and resistance to attacks on quantum computers.

Author Contributions: Conceptualization, A.L.; methodology, A.L.; validation, A.L. and V.D.; formal analysis, A.L.; investigation, A.L.; writing—original draft preparation, A.L.; writing—review and editing, V.D. and S.B.; visualization, A.L.; supervision, V.D. and S.B.; project administration, V.D. and S.B. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Not applicable.

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

References

1. Ometov, A.; Bezzateev, S.; Mäkitalo, N.; Andreev, S.; Mikkonen, T.; Koucheryavy, Y. Multi-Factor Authentication: A Survey. *Cryptography* **2018**, *2*, 1. [CrossRef]
2. Rao, V.; Prema, K. A review on lightweight cryptography for Internet-of-Things based applications. *J. Ambient Intell. Humaniz. Comput.* **2021**, *12*, 8835–8857. [CrossRef]
3. McKay, K.; Bassham, L.; Sönmez Turan, M.; Mouha, N. *Report on Lightweight Cryptography*; NIST Interagency/Internal Report (NISTIR); National Institute of Standards and Technology: Gaithersburg, MD, USA, 2017.
4. Ometov, A.; Masek, P.; Malina, L.; Florea, R.; Hosek, J.; Andreev, S.; Hajny, J.; Niutanen, J.; Koucheryavy, Y. Feasibility characterization of cryptographic primitives for constrained (wearable) IoT devices. In Proceedings of the 2016 IEEE International Conference on Pervasive Computing and Communication Workshops (PerCom Workshops), Sydney, Australia, 14–18 March 2016; pp. 1–6.
5. Coelho, K.; Damião, D.; Noubir, G.; Borges, A.; Nogueira, M.; Nacif, J. Cryptographic algorithms in wearable communications: An empirical analysis. *IEEE Commun. Lett.* **2019**, *23*, 1931–1934. [CrossRef]
6. Perumal, P.; Subha, S. An analysis of a secure communication for healthcare system using wearable devices based on elliptic curve cryptography. *World Rev. Sci. Technol. Sustain. Dev.* **2022**, *18*, 51–58. [CrossRef]
7. Hernández-Álvarez, L.; Bullón Pérez, J.J.; Batista, F.K.; Queiruga-Dios, A. Security Threats and Cryptographic Protocols for Medical Wearables. *Mathematics* **2022**, *10*, 886. [CrossRef]
8. Sowjanya, K.; Dasgupta, M.; Ray, S. An elliptic curve cryptography based enhanced anonymous authentication protocol for wearable health monitoring systems. *Int. J. Inf. Secur.* **2020**, *19*, 129–146. [CrossRef]
9. Shor, P.W. Algorithms for quantum computation: Discrete logarithms and factoring. In Proceedings of the 35th Annual Symposium on Foundations of Computer Science, Santa Fe, NM, USA, 20–22 November 1994; pp. 124–134.
10. McEliece, R.J. A public-key cryptosystem based on algebraic. *Coding Thv.* **1978**, *4244*, 114–116.

11. Alabbadi, M.; Wicker, S.B. Digital signature schemes based on error-correcting codes. In Proceedings of the IEEE International Symposium on Information Theory, San Antonio, TX, USA, 17–22 January 1993; p. 199.
12. Xinmei, W. Digital signature scheme based on error-correcting codes. *Electron. Lett.* **1990**, *26*, 898–899. [CrossRef]
13. Harn, L.; Wang, D. Cryptanalysis and modification of digital signature scheme based on error-correcting code. *Electron. Lett.* **1992**, *2*, 157–159.
14. Courtois, N.T.; Finiasz, M.; Sendrier, N. How to achieve a McEliece-based digital signature scheme. In *Advances in Cryptology—ASIACRYPT 2001, Proceedings of the 7th International Conference on the Theory and Application of Cryptology and Information Security Gold Coast, Australia, 9–13 December 2001*; Proceedings 7; Springer: Berlin/Heidelberg, Germany, 2001; pp. 157–174.
15. Rostovtsev, A.; Makhovenko, E. Cryptosysten based on category of isogenious elliptic curves. *Inf. Secur. Probl. Comput. Syst.* **2002**, *2006/145*, 74.
16. Beullens, W.; Kleinjung, T.; Vercauteren, F. CSI-FiSh: Efficient isogeny based signatures through class group computations. In *Advances in Cryptology—ASIACRYPT 2019, Proceedings of the 25th International Conference on the Theory and Application of Cryptology and Information Security, Kobe, Japan, 8–12 December 2019*; Proceedings, Part I; Springer: Berlin/Heidelberg, Germany, 2019; pp. 227–247.
17. Dartois, P.; Leroux, A.; Robert, D.; Wesolowski, B. SQISignHD: New Dimensions in Cryptography. *Cryptol. Eprint Arch.* **2023**, *2023*, 436.
18. De Feo, L.; Meyer, M. Threshold schemes from isogeny assumptions. In *Public-Key Cryptography–PKC 2020, Proceedings of the 23rd IACR International Conference on Practice and Theory of Public-Key Cryptography, Edinburgh, UK, 4–7 May 2020*; Proceedings, Part II 23; Springer: Berlin/Heidelberg, Germany, 2020; pp. 187–212.
19. Davydov, V.; Khutsaeva, A.; Ioganson, I.; Dakuo, Z.M.; Bezzateev, S. Improved threshold signature scheme CSI-FiSh with fast secret recovery. *Her. Sib. State Univ. Telecommun. Inf. Sci.* **2023**, *17*, 76–91. [CrossRef]
20. Lamport, L. *Constructing Digital Signatures from a One Way Function*; Computer Science Laboratory, SRI International: Menlo Park, CA, USA, 1979.
21. Merkle, R.C. *Secrecy, Authentication, and Public Key Systems*; Stanford University: Stanford, CA, USA, 1979.
22. Bernstein, D.J.; Hülsing, A.; Kölbl, S.; Niederhagen, R.; Rijneveld, J.; Schwabe, P. The SPHINCS+ signature framework. In Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security, London, UK, 11–15 November 2019; pp. 2129–2146.
23. Matsumoto, T.; Imai, H. Public quadratic polynomial-tuples for efficient signature-verification and message-encryption. In *Advances in Cryptology—EUROCRYPT'88, Proceedings of the Workshop on the Theory and Application of Cryptographic Techniques Davos, Switzerland, 25–27 May 1988*; Proceedings 7; Springer: Berlin/Heidelberg, Germany, 1988; pp. 419–453.
24. Patarin, J. Cryptanalysis of the Matsumoto and Imai public key scheme of Eurocrypt'88. In *Advances in Cryptology—CRYPT0'95, Proceedings of the 15th Annual International Cryptology Conference Santa Barbara, CA, USA, 27–31 August 1995*; Proceedings 15; Springer: Berlin/Heidelberg, Germany, 1995; pp. 248–261.
25. Ding, J.; Schmidt, D. Rainbow, a new multivariable polynomial signature scheme. *Proc. ACNS* **2005**, *5*, 164–175.
26. Wang, S.; Ma, R.; Zhang, Y.; Wang, X. Ring signature scheme based on multivariate public key cryptosystems. *Comput. Math. Appl.* **2011**, *62*, 3973–3979. [CrossRef]
27. Petzoldt, A.; Bulygin, S.; Buchmann, J. A multivariate based threshold ring signature scheme. *Appl. Algebra Eng. Commun. Comput.* **2013**, *24*, 255–275. [CrossRef]
28. Ajtai, M.; Dwork, C. A public-key cryptosystem with worst-case/average-case equivalence. In Proceedings of the Twenty-Ninth Annual ACM Symposium on Theory of Computing, El Paso, TX, USA, 4–6 May 1997; pp. 284–293.
29. Goldreich, O.; Goldwasser, S.; Halevi, S. Public-key cryptosystems from lattice reduction problems. In *Advances in Cryptology—CRYPTO'97, Proceedings of the 17th Annual International Cryptology Conference Santa Barbara, CA, USA, 17–21 August 1997*; Proceedings 17; Springer: Berlin/Heidelberg, Germany, 1997; pp. 112–131.
30. Fouque, P.A.; Hoffstein, J.; Kirchner, P.; Lyubashevsky, V.; Pornin, T.; Prest, T.; Ricosset, T.; Seiler, G.; Whyte, W.; Zhang, Z.; et al. Falcon: Fast-Fourier lattice-based compact signatures over NTRU. *Submiss. NIST's Post-Quantum Cryptogr. Stand. Process* **2018**, *36*.
31. Ducas, L.; Kiltz, E.; Lepoint, T.; Lyubashevsky, V.; Schwabe, P.; Seiler, G.; Stehlé, D. Crystals-dilithium: A lattice-based digital signature scheme. *IACR Trans. Cryptogr. Hardw. Embed. Syst.* **2018**, *2018*, 238–268. [CrossRef]
32. Cayrel, P.L.; Lindner, R.; Rückert, M.; Silva, R. A lattice-based threshold ring signature scheme. In *Progress in Cryptology–LATINCRYPT 2010, Proceedings of the First International Conference on Cryptology and Information Security in Latin America, Puebla, Mexico, 8–11 August 2010*; proceedings 1; Springer: Berlin/Heidelberg, Germany, 2010; pp. 255–272.
33. Bettaieb, S.; Schrek, J. Improved lattice-based threshold ring signature scheme. In *Post-Quantum Cryptography, Proceedings of the 5th International Workshop, PQCrypto 2013, Limoges, France, 4–7 June 2013, Proceedings 5*; Springer: Berlin/Heidelberg, Germany, 2013; pp. 34–51.
34. Feng, T.; Gao, Y.; Ma, J. Changeable threshold signature scheme based on lattice theory. In Proceedings of the 2010 International Conference on E-Business and E-Government, Guangzhou, China, 7–9 May 2010; pp. 1311–1315.
35. Hoffstein, J.; Howgrave-Graham, N.; Pipher, J.; Silverman, J.H.; Whyte, W. NTRUSIGN: Digital signatures using the NTRU lattice. In *Topics in Cryptology—CT-RSA 2003, Proceedings of The Cryptographers' Track at the RSA Conference 2003, San Francisco, CA, USA, 13–17 April 2003*; Springer: Berlin/Heidelberg, Germany, 2003; pp. 122–140.

36. Wang, K.; Xu, Q.; Zhang, G. A secure threshold signature scheme from lattices. In Proceedings of the 2013 Ninth International Conference on Computational Intelligence and Security, Emei Mountain, China, 14–15 December 2013; pp. 469–473.
37. Choi, R.; Kim, K. Lattice-based threshold signature with message block sharing. In Proceedings of the 31st Symposium on Cryptography and Information Security, Kagoshima, Japan, 21–24 January 2014; pp. 1–7.
38. Pilaram, H.; Eghlidos, T.; Toluee, R. An efficient lattice-based threshold signature scheme using multi-stage secret sharing. *IET Inf. Secur.* **2021**, *15*, 98–106. [CrossRef]
39. Pilaram, H.; Eghlidos, T. An efficient lattice based multi-stage secret sharing scheme. *IEEE Trans. Dependable Secur. Comput.* **2015**, *14*, 2–8. [CrossRef]
40. Micciancio, D.; Peikert, C. Trapdoors for Lattices: Simpler, Tighter, Faster, Smaller. *Eurocrypt* **2012**, *7237*, 700–718.
41. Agrawal, S.; Kirshanova, E.; Stehlé, D.; Yadav, A. Practical, round-optimal lattice-based blind signatures. In Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security, Los Angeles, CA, USA, 7–11 November 2022; pp. 39–53.
42. Boneh, D.; Gennaro, R.; Goldfeder, S.; Jain, A.; Kim, S.; Rasmussen, P.M.; Sahai, A. Threshold cryptosystems from threshold fully homomorphic encryption. In *Advances in Cryptology–CRYPTO 2018, Proceedings of the 38th Annual International Cryptology Conference, Santa Barbara, CA, USA, 19–23 August 2018*; Proceedings, Part I 38; Springer: Berlin/Heidelberg, Germany, 2018; pp. 565–596.
43. Lyubashevsky, V. Fiat-Shamir with aborts: Applications to lattice and factoring-based signatures. In *Advances in Cryptology–ASIACRYPT 2009, Proceedings of the 15th International Conference on the Theory and Application of Cryptology and Information Security, Tokyo, Japan, 6–10 December 2009*; Proceedings 15; Springer: Berlin/Heidelberg, Germany, 2009; pp. 598–616.
44. Lyubashevsky, V. Lattice signatures without trapdoors. In *Advances in Cryptology—EUROCRYPT 2012, Proceedings of the 31st Annual International Conference on the Theory and Applications of Cryptographic Techniques, Cambridge, UK, 15–19 April 2012*; Proceedings 31; Springer: Berlin/Heidelberg, Germany, 2012; pp. 738–755.
45. Schnorr, C.P. Efficient signature generation by smart cards. *J. Cryptol.* **1991**, *4*, 161–174. [CrossRef]
46. Damgård, I.; Orlandi, C.; Takahashi, A.; Tibouchi, M. Two-round n-out-of-n and multi-signatures and trapdoor commitment from lattices. *J. Cryptol.* **2022**, *35*, 14. [CrossRef]
47. Baum, C.; Damgård, I.; Lyubashevsky, V.; Oechsner, S.; Peikert, C. More efficient commitments from structured lattice assumptions. In *Security and Cryptography for Networks, Proceedings of the 11th International Conference, SCN 2018, Amalfi, Italy, 5–7 September 2018*; Springer: Berlin/Heidelberg, Germany, 2018; pp. 368–385.
48. Shamir, A. How to share a secret. *Commun. ACM* **1979**, *22*, 612–613. [CrossRef]
49. Bezzateev, S.; Davydov, V.; Ometov, A. On Secret Sharing with Newton's Polynomial for Multi-Factor Authentication. *Cryptography* **2020**, *4*, 34. [CrossRef]
50. Micciancio, D.; Regev, O. Lattice-based Cryptography. In *Post-Quantum Cryptography*; Bernstein, D.J., Buchmann, J., Dahmen, E., Eds.; Springer: Berlin/Heidelberg, Germany, 2009; pp. 147–191. [CrossRef]
51. *Sphere Packings, Lattices and Groups*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2013; Volume 290.
52. Ajtai, M. Generating hard instances of lattice problems. In Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing, Philadelphia, PA, USA, 22–24 May 1996; pp. 99–108.
53. Goldreich, O.; Micciancio, D.; Safra, S.; Seifert, J.P. Approximating shortest lattice vectors is not harder than approximating closest lattice vectors. *Inf. Process. Lett.* **1999**, *71*, 55–61. [CrossRef]
54. Ajtai, M. The shortest vector problem in L2 is NP-hard for randomized reductions. In Proceedings of the Thirtieth Annual ACM Symposium on Theory of Computing, Dallas, TX, USA, 23–26 May 1998; pp. 10–19.
55. Regev, O. On lattices, learning with errors, random linear codes, and cryptography. *J. ACM (JACM)* **2009**, *56*, 1–40. [CrossRef]
56. Brakerski, Z.; Gentry, C.; Vaikuntanathan, V. (Leveled) fully homomorphic encryption without bootstrapping. *ACM Trans. Comput. Theory (TOCT)* **2014**, *6*, 1–36. [CrossRef]
57. Langlois, A.; Stehlé, D. Worst-case to average-case reductions for module lattices. *Des. Codes Cryptogr.* **2015**, *75*, 565–599. [CrossRef]
58. Gentry, C.; Peikert, C.; Vaikuntanathan, V. Trapdoors for hard lattices and new cryptographic constructions. In Proceedings of the Fortieth Annual ACM Symposium on Theory of Computing, Victoria, BC, Canada, 17–20 May 2008; pp. 197–206.
59. Lyubashevsky, V.; Peikert, C.; Regev, O. On ideal lattices and learning with errors over rings. *J. ACM (JACM)* **2013**, *60*, 1–35. [CrossRef]
60. Pointcheval, D.; Stern, J. Security arguments for digital signatures and blind signatures. *J. Cryptol.* **2000**, *13*, 361–396. [CrossRef]
61. Ducas, L.; Lepoint, T.; Lyubashevsky, V.; Schwabe, P.; Seiler, G.; Stehle, D. CRYSTALS—Dilithium: Digital Signatures from Module Lattices. Cryptology ePrint Archive, Paper 2017/633, 2017. Available online: https://eprint.iacr.org/2017/633 (accessed on 15 January 2023).
62. Pettit, M. Efficient threshold-optimal ECDSA. In *Cryptology and Network Security, Proceedings of the 20th International Conference, CANS 2021, Vienna, Austria, 13–15 December 2021*; Springer: Berlin/Heidelberg, Germany, 2021; pp. 116–135.

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