Post-Quantum Cryptography Standardization

Helen Tera
Supervised by Dominique Unruh
University of Tartu

Abstract. This paper gives an overview of the different types of quantum-resistant algorithms for public key encryption and signature schemes, using the examples from the NIST's post-quantum cryptography standardization program.

1. Introduction

Ever since the research in the field of quantum computing was initiated in the early 1980s, it has been an area of great interest for many scientists. Today, most of the scientist believe that a fully functional quantum computer will be built and ready for use in a wide variety of fields in the coming few decades. Quantum computers can solve problems that are not feasible for conventional computers in a reasonable time by using particles that can be in superposition. Instead of using binary digits (*bits*) to encode data, quantum computers use quantum bits (*qubits*) which can take on the binary values 0 or 1 or both simultaneously.

While quantum computers can be used efficiently in scientific research and many other fields to advance the humankind, a large-scale quantum computer will pose many new problems, one of them being the security of digital communications. Quantum computers will be able to break most of the public-key cryptosystems that are in use today. Due to that, many scientists have started researching the possibilities of quantum-resistant cryptography (also called post-quantum cryptography) in order to create cryptosystems that would endure attacks from both conventional computers and quantum computers.

In order to create utilizable quantum-proof cryptosystems, scientists need to overcome various challenges. For example, it is likely that quantum-resistant algorithms will need to have larger key sizes than the algorithms that are in use today. This may result in the need to change some of the Internet protocols. Due to that, the future standards of post-quantum cryptography need to go through thorough examination and consideration.

As the need for stronger cryptography is getting more substantial, different measures are taken to address the problem. Even though transitions from smaller key sizes and algorithms have already been proposed, they will not be enough to endure attacks by quantum computers. Thus, in 2016 NIST (*National Institute of Standards and Technology*) started a competition that aims to find, develop and standardize quantum-resistant cryptosystems that would in the future replace our current cryptographic standards. Proposals for quantum-resistant public key encryption, digital signature and key exchange algorithms were accepted until the submission deadline late in 2017. Those submission will have to go through three rounds of serious examination and testing over the next few years. The final draft standards will assumingly be released somewhere between 2022-2024.

This report aims to give an overview of the submissions to the NIST's post-quantum cryptography standardization program. Firstly an examination of the submissions is presented, introducing the generalities of the competition. Then we look into the most common types of algorithms used to

provide post-quantum security using the examples from the NIST's post-quantum cryptography standardization program submissions.

2. Submissions

Due the submission deadline of late 2017 in total 69 ideas were submitted and deemed proper, including 20 digital signature algorithms and 49 public key encryption or key encapsulation schemes. Only two submissions provided all key encapsulation, public key encryption and digital signature algorithms, namely DME and Post-quantum RSA.

Since then, 5 submissions have been withdrawn. In addition, two submissions – HILA5 and ROUND2 – were merged into a new submission called ROUND5.

Proposed algorithms fall into four main categories based on the type of the proposed algorithm: lattice-based, hash-based, code-based and multivariate. Most popular algorithm types in the first round submissions were based on lattice-based cryptography with a total of 25 submissions using lattice-based cryptography, including 5 digital signature algorithms and 20 public key encryption or key encapsulation algorithms. 19 submissions were using code-based cryptography, out of which only 2 are digital signature algorithms. To the contrary, hash-based algorithms were only used in digital signature algorithms.

The table below presents the submissions by type that are under consideration after the first round.

	KEM/PKE	**Signature**	**Total**
Lattice-based	Compact LWE CRYSTALS-KYBER Ding Key Exchange EMBLEM and R.EMBLEM FrodoKEM KCL KINDI LAC LIMA Lizard LOTUS NewHope NTRUEncrypt NTRU-HRSS-KEM NTRU Prime Odd Manhattan ROUND5 SABER Three Bears Titanium	CRYSTALS-DILITHIUM DRS FALCON pqNTRUSign qTESLA	25
Hash-based		Gravity-SPHINCS SPHINCS+	2

Code-based	BIG QUAKE BIKE Classic McEliece DAGS HQC LAKE LEDAkem LEDApkc Lepton LOCKER McNie NTS-KEM Ourobors-R QC-MDPC KEM Ramstake RLCE-KEM RQC	pqsigRM RaCoSS	19
Multivariate	CFPKM Giophantus DME	DualModeMS GeMSS Gui HiMQ-3 LUOV MQDSS Rainbow	10
Other	SIKE Guess Again Mersenne-756839 pqRSA	WalnutDSA Picnic pqRSA	7
			63

Table 1. Overview of the submission by type.

Together with five withdrawn and two merged submissions, 63 proposals are under consideration today. Since post-quantum RSA scheme was submitted as two separate submissions for digital signature algorithm and for KEM/PKE algorithm, it is counted twice in the table above, while DME that was submitted only once is counted just once.

It is also important to note, that even though post-quantum RSA scheme was accepted as a submission by NIST, it is considered to be a satirical submission, since for it to be feasible and provide reasonable security, the key sizes would have to be too large to use effectively in real world.

3. Overview of the ideas

As mentioned above, proposed algorithms fall into four main families: lattice-based, hash-based, code-based and multivariate. In this section we give a short overview of these families with examples from the round 1 submissions to the NIST's post-quantum cryptography standardization program.

3.1. Code-based algorithms

McEliece cryptosystem is the first cryptosystem based on notions of coding theory and it has still not been broken since it was first published by Robert McEliece in 1978. Since then, the use of coding theory in cryptography has been widely researched.

The McEliece cryptosystem is a one-way cryptosystem – that means that an attacker without any knowledge of the target plaintext cannot reconstruct the randomly chosen codeword from a ciphertext and public key. Still, it has not been adopted widely due to its large key size.

McEliece cryptosystem is a potential alternative to current cryptography standards in the post-quantum world on account of the algorithm being based on the NP-hard problem of decoding a general linear code, even though it does not imply directly that breaking McEliece cryptosystem is NP-hard. Still, despite years of research and analysis of the McEliece cryptosystem, it has preserved its security level. Due to that, multiple submissions to the NIST's post-quantum cryptography standardization program are based directly on the classic McEliece cryptosystem, providing various improvements.

Other submissions using code-based cryptography exploit the advantages of various other codes, including different quasi–cyclic codes, Goppa codes and also multiple newly introduced codes. In this report, a submission based on McEliece and Niederreiter cryptosystems is introduced in more detail.

3.1.1. McNie: Compact McEliece-Niederreiter Cryptosystem

The submission McNie is a hybrid version of McEliece and Niederreiter cryptosystems, providing a cryptosystem with smaller key sizes while still ensuring a high security level.

General algorithm specifications

The parameters used in McNie:

- $(n-k) \times n$ matrix H,
- $n \times n$ matrix P,
- $(n-k) \times (n-k)$ matrix S,
- $l \times n$ matrix G',
- block matrix size blk,
- and $l \times (n-k)$ matrix F

over a finite field $\mathbb{F}_{q^m}$, where q is a power of a prime and $l > n-k$.

Key generation

In order to generate a secret key and a public key, Alice needs to generate an $(n-k) \times n$ parity check matrix H for an r-error correcting code over a finite field $\mathbb{F}_{q^m}$ belonging to a family of codes with a known decoding algorithm Φ_H. Her secret key consists of H, an invertible $(n-k) \times (n-k)$ matrix S and an $n \times n$ permutation matrix P.

Alice also has to randomly generate an $l \times n$ matrix G' with dimension l over $\mathbb{F}_{q^m}$. She then computes $F = G'P^{-1}H^TS$ which is an $l \times (n-k)$ matrix, publishing G' and F as her public key.

Encryption

Bob wishes to send a secret message to Alice in the form of a vector m of length l over $\mathbb{F}_{q^m}$:

- Bob generates a random error vector e of length n and weight r at most, which can be decoded by an appropriate decoding algorithm.
- The vector m is multiplied by G', then the error vector e is added to mG'. The result is a vector c_1 of length n, where $c_1 = mG' + e$.
- Then m is multiplied by F. The resulting vector is $c_2 = mF$ of length $n - k$.
- The ciphertext of length $2n - k$ is $c = (c_1, c_2)$.

Since McNie is a combination of McEliece and Niederreiter cryptosystems, $c_1 = mG' + e$ resembles the ciphertext in the McEliece cryptosystem while $c_2 = mF$ resembles the ciphertext in the Niederreiter cryptosystem.

Decryption

If Alice wants to retrieve the message, she need to decrypt (c_1, c_2) as follows:

- Alice computes $s' = c_1 P^{-1} H^T - c_2 S^{-1} = (mG' + e)P^{-1}H^T - (mG'H^T S)S^{-1} = eP^{-1}H^T$.
- The decoding algorithm Φ_H is applied to s' in order to obtain $e' = eP^{-1}$.
- e' is multiplied by P to get the error vector e.
- Alice obtains m by solving the linear system $mG' = c_1 - e$.

In order to reduce key sizes, McNie uses quasi-cyclic LRPC codes over $\mathbb{F}_{q^m}$. The detailed description of the key generation, encryption and decryption using 3-quasi-cyclic or 4-quasi-cyclic LRPC codes can be found in the documentation of McNie.

Combining the original decoding algorithm for LRPC codes introduced in the Gaborit-Ruatta-Schrek-Tillich-Zémor (GRSTZ) cryptosystem and various quasi-cyclic LRPC codes ensures that McNie is at least as secure as GRSTZ. Since in McNie the public key G' does not contain any information about the private key H, it is likely that McNie is harder to break than the GRSTZ cryptosystem. That also means that attacking G' does not expose the private code generated by H, which makes it secure against finding low-weight codewords.

It is proven that it is not easier to break McNie than the original McEliece cryptosystem.[5] McNie is also secure against various structural and information set decoding attacks, because a random code is used in the encryption.

3.2. Hash-based algorithms

Hash-based algorithms are quite different from other potential post-quantum schemes. So far, hash-based cryptography is limited to digital signatures schemes and is not used for key encapsulation or public key encryption. First hash-based signature schemes date back to late 1970s and thus their security is well understood, even against quantum attacks.

Hash-based functions rely completely on the security of the underling hash function. That makes hash-functions very adjustable and resistant against quantum attacks. If a hash function becomes insecure, it can be replaced by another, making the signature scheme safe to use once again.

The main disadvantage of hash-based schemes is that they can be used for a limited number of signatures only. The number of signatures can be increased, but only at the expense of signature size.

First hash-based signature scheme was introduced by Leslie Lamport in 1979. The one-time signature scheme was later extended by Ralph Merkle who combined it with hash trees and thus made it possible to use one Lamport key to sign multiple messages.

Merkle starts with a one-time signature scheme like the Lamport signature scheme and then uses a binary tree of height h to authenticate 2^h one-time signature key pairs. Thus, the root of the tree becomes the public key and one-time signature secret keys become the secret key of the new scheme. Using this scheme allows to sign 2^h messages.

This idea has since been used in many signature schemes. After more than 40 years of research eXtended Merkle Signature Scheme (XMSS) was introduced. XMSS is about to be standardised by CFRG, making it the first standardised quantum-resistant signature scheme. It has many strong points, but the main downside is that it is stateful. That means that signing with XMSS requires keeping state of the used one-time keys in order to make sure they are never used again. Unfortunately, being stateless is one of the requirements for the signature schemes proposed to NIST's post-quantum cryptography standardization program.

In 2015 a stateless signature scheme was proposed – SPHINCS. SPHINCS has become a baseline for modern hash-based signature schemes. SPHINCS uses randomized index selection – the index of the one-time signature key pair is chosen randomly, instead of applying a hash function to the message to determine the index. In order to increase the security of randomized index selection, one-time signature scheme in the leaf is replaced with a few-time signature scheme. This allows a few index collisions, which in turn allows a smaller tree height for the same security level. [7]

The hash-based digital signature schemes submitted to NIST's post-quantum cryptography standardization program are both based on SPHINCS, offering various improvements in security and speed. In this paper, one of two hash-based signature schemes is introduced in more detail – SPHINCS+.

3.2.1. SPHINCS+

SPHINCS+ works similarly to SPHINCS. The main idea remains the same – SPHINCS+ authenticates a big number of few-time signature (FTS) key pairs using a so-called hypertree. To sign a message, a random FTS key pair is chosen. The resulting signature consists of the authentication information for that FTS key pair and the FTS signature.

A hypertree consists of hash-based many-time signatures (MTS), which allow a key pair to sign a fixed number of N messages, where N is a power of 2. The many-time signature key pairs are held in a d-layer N-ary tree. The top layer holds a single many-time signature key pair which is used to sign the public keys of N many-time signature key pairs from the next layer, which are in order used to sign MTS public keys from the next layer. The N^{d-1} key pairs from the bottom layer are used to sign N FTS public keys, resulting in a total of N^d authenticated FTS key pairs.

As a result, the authentication information for an FTS key pair consists of the d MTS signatures that build a path from the FTS key pair to the top MTS tree. The OTS and FTS secret keys together fully determine the whole virtual structure of an SPHINCS+ key pair. [8]

An MTS signature used in SPHINCS+ is just a classical Merkle-tree signature consisting of a one-time signature (OTS) plus the authentication path in the binary hash-tree. Similarly to SPHINCS, SPHINCS+ uses WOTS+ as its one-time signature scheme.

The structure of the SPHINCS+ key pair is fully determined by the secret keys of its OTS and FTS.

3.3. Lattice-based algorithms

More than a third of algorithms proposed to NIST's post-quantum cryptography standardization program were built on lattice-based cryptography. In total, 25 lattice-based schemes were proposed. Lattice-based cryptosystems are favoured due to their worst-case hardness security proof and quantum-resistance. In addition, lattice-based systems are often more efficient, because they do not require any difficult computations.

One of the main drawbacks of lattice-based algorithms is their newness, hence the security parameters like key length are not well understood or established. This is a relatively small problem, since in the past years the number of publications on the topic of lattice-based algorithms has grown substantially.

3.3.1. Mathematical background

By definition, a **lattice** $\mathcal{L}$ of $\mathbb{R}^n$ is a discrete subgroup of $\mathbb{R}^n$. Intuitively, a lattice can be thought of as a regularly spaced infinite n-dimensional grid of points. In practice, n has to be rather large to provide security.

A **basis** of lattice $\mathcal{L}$ is an arbitrary set of linearly independent vectors $B = \{\overrightarrow{b_i}\}$ such that $\mathcal{L} = \{\sum a_i \overrightarrow{b_i} : a_i \in \mathbb{Z}\}$. We denote a lattice $\mathcal{L}$ with basis B as $\mathcal{L}(B)$, where basis B can be thought of as an $n \times n$ matrix with columns $\overrightarrow{b_i}$.

Bases are not unique – multiple bases can generate the same lattice. Two bases B_1 and B_2 are equivalent if $B_1 = B_2 U$, where U is a integer matrix with a determinant of ± 1 (unimodular matrix).

$\lambda_1(\mathcal{L})$ denotes the length of the shortest non-zero vector in $\mathcal{L}$. More generally, $\lambda_{\mathrm{k}}(\mathcal{L})$ denotes the smallest radius of a sphere containing k linearly independent vectors:

$$\lambda_{\mathrm{k}}(\mathcal{L}) \coloneqq \min\{r: \mathcal{L} \text{ contains k linearly independent vectors of length } \leq \mathrm{r}\}$$

The cryptographic systems using lattices are based on various computational problems:

- Shortest vector problem (SVP) – given a basis B, find $\lambda_1(\mathcal{L}(B))$. One of the most common variations of SVP is SVP$_\gamma$ – given a basis B, find a vector of length $\leq \gamma\lambda_1(\mathcal{L}(B))$.
- Shortest independent vectors problem (SIVP$_\gamma$) – given a basis B, find n linearly independent vectors in $\mathcal{L}(B)$ of length $\leq \gamma\lambda_{\mathrm{n}}(\mathcal{L}(B))$.
- Closest vector problem (CVP) – given a basis B and a randomly chosen point v, find the closest lattice point to v in $\mathcal{L}(B)$. A less strict version of this is (CVP$_\gamma$) – given a basis B and a point v, find a lattice point that is at most γ times farther from v than the closest lattice point to v.

Multiple other variations of these problems are used in practice. One of the most common one is GapSVP$_\gamma$ – given a basis B and a real d, decide between $\lambda_1(\mathcal{L}(B)) \leq d$ and $\lambda_1(\mathcal{L}(B)) > \gamma d$.

All of these problems are hard in the worst case.

3.3.2. LWE and RLWE

In 2005 Regev published a paper, in which a reduction from worst-case lattice problems such as GapSVP and SIVP to a certain learning problem was presented. This learning problem, called learning with errors (LWE), has become the basis for most modern lattice-based cryptosystems.

Fix a size parameter $n \geq 1$, a modulus $q \geq 2$ and an error probability distribution χ on $\mathbb{Z}_q$. The learning with errors problem consists of recovering a secret $s \in \mathbb{Z}_q^n$ given a sequence of approximate random linear equations on s:

$$a_1 \leftarrow \mathbb{Z}_q^n, b_1 = \langle s, a_1 \rangle + e_1$$

$$a_2 \leftarrow \mathbb{Z}_q^n, b_2 = \langle s, a_2 \rangle + e_2$$

$$\vdots$$

$a_i \in \mathbb{Z}_q^n$ is chosen uniformly at random and $e_i \in \mathbb{Z}_q$ is chosen according to χ. The error distribution χ is a normal distribution rounded to the nearest integer of standard deviation αq where $\alpha > 0$.

In order to provide worst-case hardness α must satisfy $\alpha q > \sqrt{n}$, as proven by Regev.

Let us note, that the problem of recovering secret s is equivalent to finding e, since without the noise, the system can be solved using Gaussian elimination.

This adaptation of LWE is referred to as **search-LWE**. Another very common variation of LWE is **decision-LWE**, which is roughly equivalent in hardness**.** The aim of decision-LWE is to distinguish pairs (a_i, b_i), where $b_i = \langle s, a_i \rangle + e_i$ from uniform pairs (a_i, b_i), where b_i is chosen uniformly at random.

Using the problem of learning with errors, a simple cryptosystem can be built. This cryptosystem is parameterized by the security parameter n, number of equations m, modulus q and a real noise parameter $\alpha > 0$.

Key generation

The private key is a vector s chosen uniformly from $\mathbb{Z}_q^n$. The public key consists of m samples (a_i, b_i) from the LWE distribution, where $b_i = \langle s, a_i \rangle + e_i$, using the secret s, modulus q and a noise parameter α.

Encryption

For each bit of the message a random set S is uniformly chosen among all 2^m subsets of $[m]$. If the bit is 0, the encryption is $(\sum_{i \in S} a_i, \sum_{i \in S} b_i)$, otherwise if the bit is 1, the encryption of the bit is $(\sum_{i \in S} a_i, \left\lfloor \frac{q}{2} \right\rfloor + \sum_{i \in S} b_i)$.

Decryption

The decryption of a pair (a, b) is 0 if $b - \langle a, s \rangle$ is closer to 0 than to $\left\lfloor \frac{q}{2} \right\rfloor$ modulo q, and 1 otherwise.[14]

A possible choice of parameters proposed by Regev that guarantee both security and correctness is the following:

- q is a prime between n^2 and $2n^2$
- $m = 1.1 \cdot n \log q$
- $\alpha = 1/(\sqrt{n} \log^2 n)$

Even though the cryptosystem proposed above is rather inefficient, it gives a good insight into the field of lattice-based cryptography based on the problem on LWE. The idea of using LWE problem as a basis of the cryptosystem has been very popular since, which has led to big amount of follow-up work and multiple improvements.

One of the most researched versions of LWE is **ring-LWE** or more correctly learning with errors over rings. RING-LWE is much more efficient than the regular LWE problem, but it also requires the use of lattices that possess extra algebraic structure – ideal lattices, the description of which is beyond the scope of this work.

The ring-LWE problem is quite similar to the classic search-LWE. Let n be a power of two and q be a prime modulus satisfying $q = 1\ mod\ 2n$. Then R_q is defined as the ring $\mathbb{Z}_q[x]/\langle x^n + 1\rangle$ which contains all polynomials over the field $\mathbb{Z}_q$ in which x^n is identified with −1. The goal is to recover the secret s from the samples of the form $(a,\ b = a \cdot s + e) \in R_q \times R_q$ where $s \in R_q$ is a fixed secret, $a \in R_q$ is chosen uniformly and e is an error term chosen independently from some error distribution over R_q.

Ring-LWE offers many improvements to the classical LWE. The size of the public key is substantially smaller than in the LWE based cryptosystem and it is also at least as secure as LWE. It is proven, that just as LWE, ring-LWE using ideal lattices reduces to worst-case lattice problems like SVP. Despite considerable effort, no significant progress in attacking these problems has been made.

3.4. Multivariate algorithms

While a number of multivariate encryption schemes have been proposed to NIST's post-quantum standardization program, multivariate cryptography has historically been more successful in signature schemes. Out of the 19 submitted signature schemes, multivariate algorithms take up the biggest part with seven multivariate signature schemes.

In this paper, the multivariate quadratic (MQ) signature schemes are introduced. The general structure of a MQ-signature (multivariate quadratic) scheme over $\mathbb{F}_q$ is as follows.

Let us define a system $\mathcal{P} = (P^{(1)}, \dots,\ P^{(m)})$ of multivariate quadratic polynomials of m equations and n variables by

$$\mathcal{P}^{(k)}(x_1, \dots, x_n) = \sum_{i=1}^{n}\sum_{j=1}^{n} p_{ij}^{(k)} x_i x_j + \sum_{i=1}^{n} p_i^{(k)} x_i + p_0^{(k)}$$

For $k = 1, \dots, m$ and $p_{ij}^{(k)}, p_i^{(k)}, p_0^{(k)} \in_R \mathbb{F}_q$.

The main idea for the construction of MQ-signature scheme is to choose a central map $\mathcal{F} = \left(\mathcal{F}^{(1)}, \dots,\ \mathcal{F}^{(m)}\right) : \mathbb{F}_q^n \to \mathbb{F}_q^m$ of multivariate quadratic polynomials, which can be easily inverted. After that two affine or linear invertible maps $S : \mathbb{F}_q^m \to \mathbb{F}_q^m$ and $T : \mathbb{F}_q^n \to \mathbb{F}_q^n$ are chosen, in order to hide the structure of the central map in a public key.

A public key is the composed quadratic map $\mathcal{P} = S \circ \mathcal{F} \circ T$ which is supposedly hardly distinguishable from a random system and therefore difficult to invert.

A secret key consists of $(S, \mathcal{F}, T)$ which allows to invert $\mathcal{P}$.[16]

The figure 1 below illustrates the process of generating and verifying a signature.

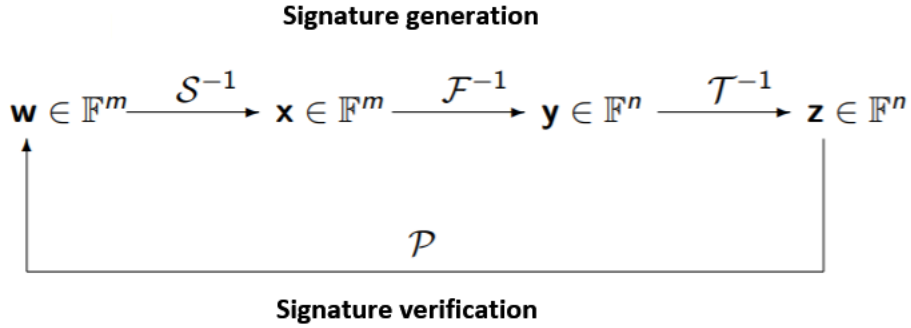

Figure 1. The process of generating and verifying a signature using the MQ-signature scheme.

Security. Security analysis of multivariate schemes is rather difficult, since no direct reduction from a NP-hard problem exists. The algorithm above is based on the following underlying problems:

- **Polynomial System Solving (PoSSo) Problem**. Given a system $\mathcal{P} = (P^{(1)}, \dots, P^{(m)})$ over $\mathbb{F}_q$ and $y = (y_1, \dots, y_m)$, find values $(x'_1, \dots, x'_n) \in \mathbb{F}_q^n$ such that

$$P^{(1)}(x'_1, \dots, x'_n) = y_1$$
$$\vdots$$
$$P^{(m)}(x'_1, \dots, x'_n) = y_m$$

- **EIP (Extended Isomorphism of Polynomials) Problem**. Given a nonlinear multivariate system $\mathcal{P}$ such that $\mathcal{P} = S \circ \mathcal{F} \circ T$ for linear or affine maps S and T and $\mathcal{F}$ belonging to a special class of nonlinear polynomial system $\mathcal{C}$, find a decomposition of $\mathcal{P}$ such that $\mathcal{P} = S' \circ \mathcal{F}' \circ T'$ for linear or affine maps S' and T', and $\mathcal{F}' \in \mathcal{C}$.

Since the multivariate signature schemes are rather new, a lot of research is needed to prove their security. Many of the earlier multivariate algorithms have been broken. Still, due to their small signature sizes and fast signature verification (or encryption), multivariate cryptosystems remain as very strong competitors for the potential quantum-resistant electronic signature standards. The multivariate polynomial based algorithms submitted to NIST's post-quantum competition provide various improvements both in speed, key sizes and security. Due to the scope of this paper, none of them will be introduced in more detail.

4. Conclusion

It is hard to predict which family of quantum-resistant algorithms will prove to be the most efficient in the future. While lattice-based cryptosystems have found most research, code-based algorithms remain a solid choice for the future cryptographic standards, whilst both hash-based and multivariate algorithms provide secure signature schemes. The NIST's post-quantum standardization program gives a great overview of the field and presents us a variety of options for the future cryptostandards, leaving NIST with a difficult task of examining and testing all of the submissions to find the most efficient and secure algorithms.

Bibliography

1. L. Chen, S. Jordan, Y.K. Liu, D. Moody, R. Peralta, R. Perlner, D. Smith-Tone. (2016) *NISTIR 8105 Report on Post-Quantum Cryptography*. National Institute of Standards and Technology.
2. https://csrc.nist.gov/Projects/Post-Quantum-Cryptography
3. S. Siim. (2015) *Study of McEliece cryptosystem*. University of Tartu.
4. R. J. McEliece. (1978) *A Public-Key Cryptosystem Based on Algebraic Coding theory*. The Deep Space Network Progress Report.
5. L. Galvez, J.-L. Kim, M. J. Kim, Y.-S. Kim, N. Lee. (2017) *McNie: Compact McEliece-Niederreiter Cryptosystem*.
6. P. Gaborit, G. Murat, O. Ruatta, G. Zémor. (2013) *Low Rank Parity Check codes and their application to cryptography*.
7. D. J. Bernstein, D. Hopwood, A. Hülsing, T. Lange, R. Niederhagen, L. Papachristodoulou, P. Schwabe, Z. W. O'Hearn. (2015) *SPHINCS: practical stateless hash-based signatures*.
8. D. J. Bernstein, C. Dobraunig, M. Eichlseder, S. Fluhrer, S.-L. Gazdag, A. Hülsing, P. Kampanakis, S. Kölbl, T. Lange, M. M. Lauridsen, F. Mendel, R. Niederhagen, C. Rechberger, J. Rijneveld, P. Schwabe. (2017) *SPHINCS+ Submission to the NIST post-quantum project*.
9. H. Lipmaa. (2013) *Lecture in Cryptographic Protocols: Lattice-Based Cryptography*. University of Tartu.
10. J. Alwen. (2018) *What is Lattice-based cryptography & why should you care*. Medium.
11. D. P. Chi, J. W. Choi, J. S. Kim, T. Kim. (2015) *Lattice Based Cryptography for Beginners*. IACR Cryptology ePrint Archive.
12. O. Regev. (2012) *Introduction to Lattices. Winter School on Lattice-Based Cryptography and Applications*. Bar-Ilan University.
13. O. Regev. (2005) *On Lattices, Learning with Errors, Random Linear Codes, and Cryptography*. Tel-Aviv University.
14. O. Regev. (2010) *The Learning with Errors Problem*.
15. V. Lyubashevsky, C. Peikert, O. Regev. (2010) *On Ideal Lattices and Learning with Errors Over Rings*. EUROCRYPT 2010.
16. K.-A. Shim, C.-M. Park, A. Kim. (2017) *HiMQ-3: A High Speed Signature Scheme based on Multivariate Quadratic Equations*.