UNIVERSITY OF TARTU
Institute of Computer Science
Computer Science Curriculum

Helen Tera

Introduction to Post-Quantum Cryptography in scope of NIST's Post-Quantum Competition

Bachelor’s Thesis (9 ECTS)

Supervisor: Dominique Unruh, PhD

Tartu 2019

Introduction to Post-Quantum Cryptography in scope of NIST's Post-Quantum Competition

Abstract:

Nowadays, information security is essential in many fields, ranging from medicine and science to law enforcement and business, but the developments in the area of quantum computing have put the security of current internet protocols at risk. Since quantum computers will likely be able to break most of our current cryptostandards in trivial time, a need for stronger and quantum-resistant encryption algorithms has arisen. During the last decades, a lot of research has been conducted on the topic of quantum-resistant cryptography, yet none of the post-quantum algorithms have yet been standardized. This has encouraged NIST to start a program to select the future post-quantum cryptography standards. This thesis gives an overview of different types of quantum-resistant algorithms for public key encryption and signature schemes, using the examples from NIST's post-quantum cryptography standardization program. The aim of this paper is to compose a compact material, which gives a person with computer science background a basic understanding of the main aspects of post-quantum cryptography.

Keywords:

Post-quantum cryptography, post-quantum cryptography standardization, lattice based cryptography, code based cryptography, multivariate polynomials, hashes

CERCS: P170 - Computer science, numerical analysis, systems, control

Postkvantkrüptograafia alused NISTi standardiseerimisprogrammi põhjal

Lühikokkuvõte:

Tänapäevases veebipõhises maailmas on andmeturve paljudes valdkondades määrava tähtsusega, kuid hiljutised edasijõudmised kvantmehhaanika valdkonnas võivad tänased interneti turvaprotokollid ohtu seada. Kuna kvantarvutid on tõenäoliselt võimelised murdma meie praeguseid krüptostandardeid, tekib vajadus tugevamate krüpteerimisalgoritmide järele. Viimaste kümnendite jooksul on postkvantkrüptograafia saanud palju tähelepanu, kuid siiani pole ükski postkvantkrüptograafiline algoritm standardiseeritud ulatuslikuks kasutamiseks. Selle tõttu algatas NIST programmi, mille eesmärk on valida uued krüptostandardid, mis säilitaks oma turvalisuse ka kvantarvutite vastu. Käesolev lõputöö annab ülevaate postkvantkrüptograafia erinevatest valdkondadest kasutades näiteid NISTi standardiseerimisprogrammist. Lõputöö eesmärk on koostada ülevaatlik materjal, mis annaks informaatika või matemaatika taustaga tudengile laiahaardelised algteadmised postkvantkrüptograafia valdkonnast.

Võtmesõnad:

Postkvantkrüptograafia, postkvantkrüptograafia standardiseerimine, võrepõhine krüptograafia, koodipõhine krüptograafia, räsipõhine krüptograafia

CERCS: P170 - Arvutiteadus, arvutusmeetodid, süsteemid, juhtimine

Table of Contents

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, it is widely believed that a fully functional quantum computer will be built and ready for use in a wide variety of fields in the coming 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. [1]

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 in trivial time. 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-secure 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, which in return 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. [2]

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, which will be referred to as NIST's Post-Quantum Competition in this paper, 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 submissions 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 between 2022-2024. [3]

This thesis 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. As a result, a concise paper is composed, which should give a person with previous knowledge in the sphere of computer science or mathematics an overview of the basic principles of post-quantum cryptography.

2. Submissions to NIST’s Post-Quantum Competition

The current chapter presents an overview of the submissions to the NIST’s post-quantum competition and is based on the official information from NIST’s webpage. [4]

2.1 Round 1

Due the submission deadline in late 2017 in total 69 ideas were submitted and accepted by NIST, 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 together, namely DME and Post-quantum RSA.

It is also important to note, that even though post-quantum RSA scheme was accepted as a submission by NIST, it is considered 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.

Before the start of the second round, five submissions were withdrawn. In addition, two submissions – HILA5 and ROUND2 – were merged into a new submission called ROUND5.

With five submission withdrawn and two merged together, 63 proposals remained under consideration at the end of the first round.

Proposed algorithms fall into four main categories based on the type of the algorithm: lattice-based, hash-based, code-based and multivariate. The most popular algorithm type in the first round submissions was based on lattice-based cryptography with a total of 25 submissions using lattice-based cryptography, including five digital signature algorithms and twenty public key encryption or key encapsulation algorithms. Nineteen submissions were using code-based cryptography, out of which only two were digital signature algorithms. To the contrary, hash-based algorithms were only used in digital signature algorithms. A small part of the algorithms did not belong to any of the aforementioned families. The number of submissions of each type is portrayed in Figure 1 below.

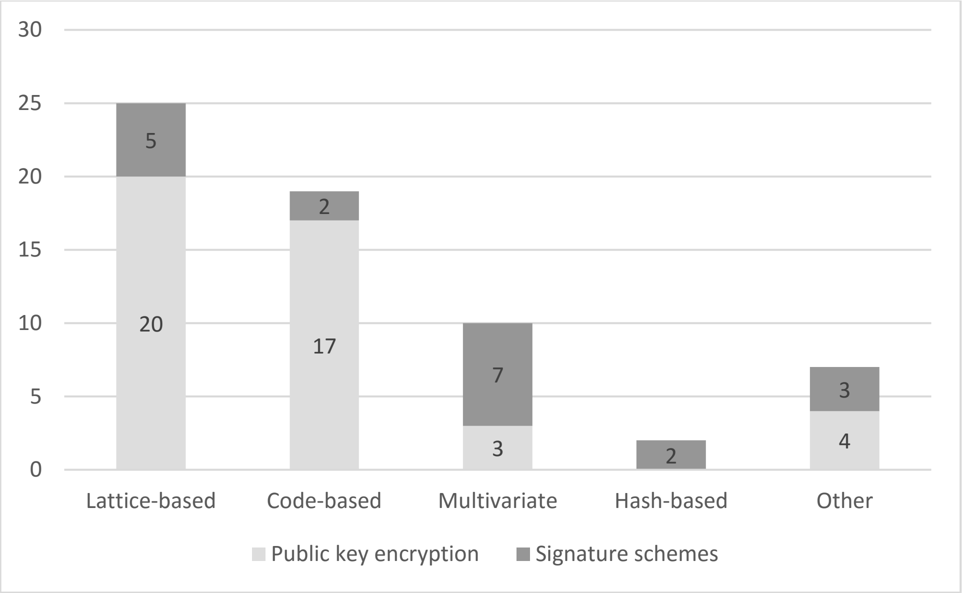

Figure 1. Round 1 submissions by type.

Based on the submissions to the first round of NIST's post-quantum competition, lattice-based and code-based algorithms are the preferred candidates for future cryptostandards.

2.2 Round 2

The Round 2 submissions were announced in the end of January 2019. 26 candidates were selected out of the 63 candidates from Round 1, including 17 public-key encryption algorithms and 9 digital signature schemes.

Most of the remaining candidates fall under the category of lattice-based cryptography, making up 12 out of 26 Round 2 submissions. Just as in the first round, the second most common algorithm type is based on coding theory. Multivariate polynomial based systems were most favoured for digital signature schemes with four schemes. Other families, such as hash-based, elliptic curve and zero knowledge were represented once each. Figure 2 below depicts Round 2 candidates by algorithm type.

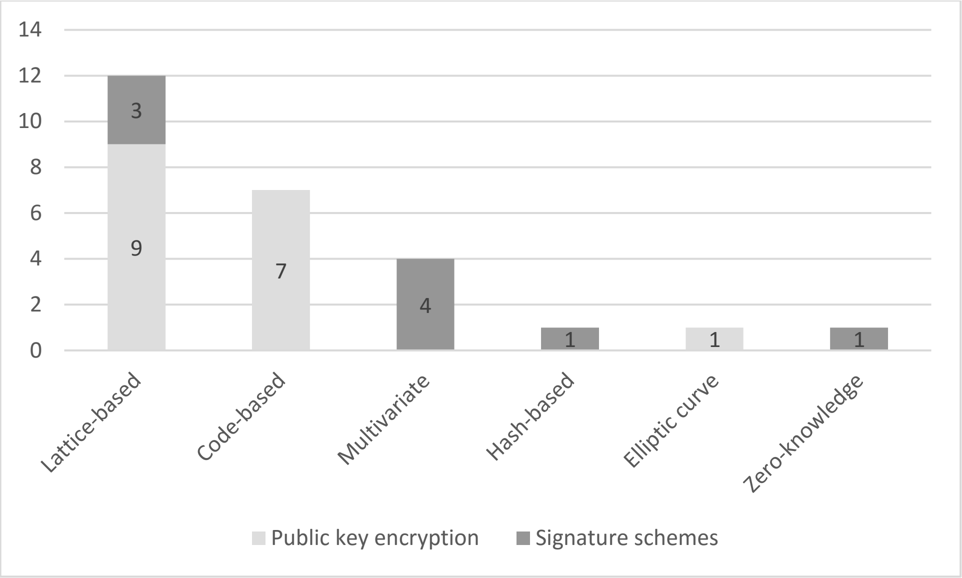

Figure 2. Round 2 submissions by type.

In the coming upchapters, the main families of quantum-resistant algorithms are introduced in more detail, specifically lattice-, code-, hash- and multivariate polynomial based. Examples from NIST’s post-quantum competition are used where appropriate.

3. Lattice-based algorithms

More than a third of the algorithms proposed to NIST's post-quantum cryptography standardization program were built on lattice-based cryptography. In total, twenty-five lattice-based schemes were proposed during Round 1. Lattice-based cryptosystems are favoured due to their proof of worst-case hardness security and quantum-resistance. In addition, lattice-based systems are often more efficient, because they do not require any difficult computations. [5]

One of the main drawbacks of lattice-based algorithms is their newness, hence the security parameters like key length are not well established and understood. 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. [6]

The following chapters are based on multiple different works on lattice-based cryptography, such as a Master's thesis by F. Bergami [7], multiple papers by O. Regev [8, 9] and other works on the topic [10-12]. In the end, an example from NIST's post-quantum standardization program is described briefly, solely based on its documentation [13].

3.1 Preliminaries

A **basis** of lattice $\mathcal{L}$ is an arbitrary set of linearly independent vectors $B = \{\overrightarrow{b_i}\}$ such that $\mathcal{L} = \{\sum a_i \overrightarrow{b_i} : a_i \in \mathbb{Z}\}$. In other words, basis is a set of vectors that can be used to reproduce any point in the lattice. 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 $\overrightarrow{b_i}$.

The **lattice** $\mathcal{L}$ generated by basis B is the set of all the integer linear combinations of the vectors in B. Intuitively, a lattice can be thought of as a regularly spaced infinite n-dimensional grid of points.

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

Figure 3 below depicts a 2-dimensional lattice with basis $B = \{\overrightarrow{b_1}, \overrightarrow{b_2}\}$. In practice, the dimension n has to be rather large to provide reasonable security.

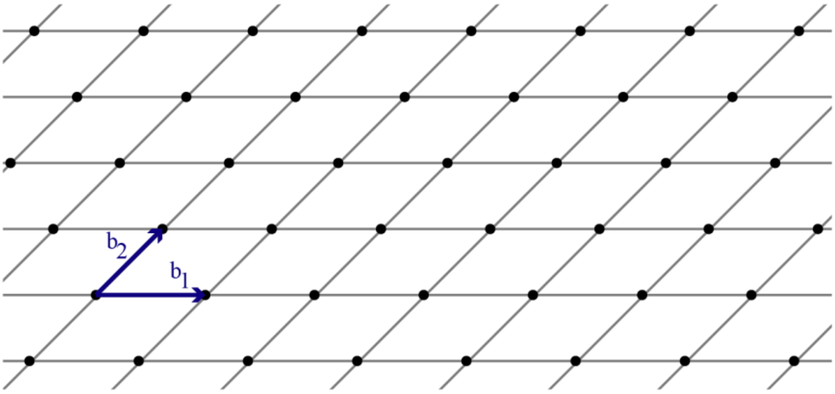

Figure 3. Two-dimensional lattice with base vectors b_1 and b_2.

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

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

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

- **Shortest vector problem** (SVP) – given a basis B, find a vector of length $\lambda_1(\mathcal{L}(B))$. 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 $\leq \gamma\lambda_n(\mathcal{L}(B))$;
- **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 to solve and despite intensive research, no efficient algorithm has been found for any of these problems.

One of the reasons why lattice-based cryptosystems are one of the most promising options for future cryptostandards, is their worst-case security guarantee. That means that breaking their security is known to be at least as hard as solving the underlying lattice problem in any of its instances including the worst one.

3.2 Learning With Errors

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. [9] 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 indicated by Regev. [9]

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.

The adaptation of LWE presented above is referred to as **search-LWE**. Another very common variation of LWE is **decision-LWE**. 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.

While the actual reduction from lattice problems such as GapSVP to LWE problem is beyond the scope of this work, a more intuitive connection between lattices and LWE is given. Having n samples of $b_i = \langle s, a_i \rangle + e_i$ from the LWE distribution, we can present the associated vector $b = (b_1, \dots, b_n)$ as $b = As + e$, where $A^T = (a_1 | \dots | a_n)$ and $e = (e_1, \dots, e_n)$ is a small noise vector. In that case, we can think of As as a point in the lattice $\mathcal{L}(a_1, \dots, a_n)$, defined by the coefficients from s. Since e is small, b must be quite close to this lattice point, thus finding the secret vector s corresponds to the closest vector problem (CVP), which in term can be translated to SVP problems.

3.3 Regev's cryptosystem

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.

Let us note, that without the error e, the system could be easily solved with Gaussian elimination and $b - \langle a, s \rangle$ would always be either 0 or $\left\lfloor \frac{q}{2} \right\rfloor$. Thus, the decryption error occurs only if the sum of the errors over chosen set is greater than $\frac{q}{4}$, which does not happen due to the chosen error distribution. In order to better understand Regev's cryptosystem, an example with small parameters is presented.

Example

Parameters used in this example are $q = 3$, $m = 4$ and $n = 3$. Let us assume that Alice wants to send Bob an encrypted message. Bob's secret key $s = (2\ 0\ 1)^T$ is chosen uniformly at random. Public key consists of the pair (A, b), where A is a 4×3 matrix

$$A = \begin{pmatrix} 1 & 2 & 2 \\ 1 & 2 & 0 \\ 0 & 1 & 0 \\ 2 & 0 & 1 \end{pmatrix}$$

and $b = (1\ 2\ 0\ 2)$ is a vector.

If Alice wants to send a message to Bob, she chooses a random subset of rows from A, in this case she has chosen rows 1 and 3. She then has to calculate the encryption of her message, based on the bit that needs to be encrypted. If the bit is 0, its encryption is $(\sum_{i\in S} a_i, \sum_{i\in S} b_i) = ((1\ 0\ 2), 1)$, while if the bit is 1, the encryption is $\left(\sum_{i\in S} a_i, \left\lfloor\frac{q}{2}\right\rfloor + \sum_{i\in S} b_i\right) = ((1\ 0\ 2), 2)$. When Bob receives the message, he needs to compute $b - \langle a, s\rangle$. In the first case (if the encrypted bit was 0), the result is $b - \langle a, s\rangle = 1 - 4 = 0$. Since it is closer to 0 than to $\left\lfloor\frac{q}{2}\right\rfloor\ mod\ q$, he can be sure that the original bit was 0. In the other case, where the encrypted bit was 1, he calculates $b - \langle a, s\rangle = 2 - 4 = 1$, which in term is closer to $\left\lfloor\frac{q}{2}\right\rfloor = 1$.

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 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 a big amount of follow-up work and multiple improvements.

One of the most researched variants of LWE is **ring-LWE** or more correctly learning with errors over rings. RLWE is 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.

Ring-LWE offers multiple improvements over 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 FrodoKEM

One of the many lattice-based cryptosystems proposed to NIST’s post-quantum standardization program was FrodoKEM. The core of FrodoKEM is an LWE public-key encryption scheme called FrodoPKE, which is based on the original Regev’s cryptosystem. Unlike most of the algorithms proposed to NIST’s program, FrodoKEM does not use ring-LWE as its basis, but holds to the original LWE problem.

FrodoKEM does require moderately longer running times than the submissions based on ring-LWE, but in return, FrodoKEM offers simplicity and compactness, reducing the potential for errors. For example, the base code given in the specification can be used for different LWE security levels, without making major changes to the code solely by changing compile-time constants. FrodoKEM imposes very few requirements on its parameters, which makes it possible to meet almost any desired security target in an automated way.

4. Code-based algorithms

Code-based cryptography has been widely researched ever since Robert McEliece published his groundbreaking research in 1978. McEliece cryptosystem is the first fully functioning code-based cryptosystem and even though McEliece cryptosystem with original parameters has been broken, it is still very expensive to attack. Breaking it becomes nearly impossible with larger key sizes. Furthermore, decryption and encryption process is faster than, for example, in RSA. The main disadvantage of McEliece is its already large key size. [14]

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. Its quantum-security and speed encouraged multiple submissions to the NIST's post-quantum competition, that are directly based on the classic McEliece cryptosystem, providing various improvements for decreasing key sizes and improving security.

Other submissions based on code-based cryptography exploit the advantages of various codes, including different quasi–cyclic codes, Goppa codes and multiple newly introduced codes, which will not be explained in detail in this thesis. In this report, a cryptosystem based on classic McEliece system is presented.

This chapter is mostly founded on two papers: the original report by Robert McEliece [15] and the Classic McEliece submission to NIST's post-quantum standardization program [16], while also building upon ideas brought up in a study written in the University of Tartu [17].

4.1 Preliminaries

A $[n, k]$-**linear code** C is a k-dimensional linear subspace of a finite field $\mathbb{F}^n$ of size n. We say that code C has a length n and dimension k. In the McEliece cryptosystem we only work with binary linear codes over the field $\mathbb{F}_2$. Codewords will be expressed as bit vectors.

The Hamming **weight** of codeword $x \in \mathbb{F}_2^n$, denoted $wt(x)$ is defined as the number of coordinates that are not equal to zero. That is equal to its distance from the zero-vector: $wt(x) = d(x, 0)$.

The **Hamming distance** of two codewords $x = x_1, \dots, x_n$ and $y = y_1, \dots, y_n$ is defined as

$$d(x, y) = \sum_{i=1}^{n} d(x_i, y_i)$$

where $(x_i, y_i) = \begin{cases} 1, & x_i \neq y_i \\ 0, & x_i = y_i \end{cases}$.

The **distance** of code C is defined as the minimum Hamming distance of any two distinct codewords of C:

$$d(C) = \min_{\substack{x,y \in C \\ x \neq y}} d(x, y)$$

If C is an $[n, k]$-linear code with distance d, then C is called an $[n, k, d]$-linear code.

Generator matrix G of a $[n, k]$-linear code C is a $k \times n$ matrix whose rows form a basis of the code C. Let us note that the generator matrix for a linear code is generally not unique, since every basis of C gives a different, but equivalent generator matrix for C.

Let C be an $[n, k, d]$-linear code with generator matrix G. We say that code C can correct up to t errors, if there exists a decoding algorithm $Dec : \mathbb{F}^n \to C$ such that for every $u \in \mathbb{F}^k$ and every vector $e \in \mathbb{F}^n$ with weight $wt(e) \leq t$, the word $y = uG + e$ is always correctly decoded as $Dec(y) = u$. Code C is then an **error-correcting code**.

A **permutation matrix** P is a binary matrix, whose every row and every column each consist of a single 1, while all other values are 0. That means, that multiplying any matrix with a permutation matrix P results in a matrix contains all the same columns as the original matrix, but in permuted order.

4.2 Construction of McEliece Cryptosystem

Let C be a random code, such that $C = [n, k, d]$ for which there is an efficient algorithm Dec_C, that can decode any codeword with up to $t = \left\lfloor \frac{d-1}{2} \right\rfloor$ errors.

Let G be a $k \times n$ generator matrix of code C, let S be a random non-singular $k \times k$ matrix and let P be a random $n \times n$ permutation matrix.

McEliece's system is constructed as follows.

Key generation:

- Pick a random $k \times n$ generator matrix G of code C. Generate a random non-singular $k \times k$ matrix S and a random $n \times n$ permutation matrix P.
- Public key: $SGP = G'$
- Private key: (S, G, P)

Encryption:

- Let m be a k-bit message
- Let e be an random n-bit vector such that $wt(e) = t$
- Then $c = m \cdot G' + e$ is the ciphertext

Decryption:

- The recipient uses his private key to compute P^{-1}, the inverse of P
- The recipient computes $c' = cP^{-1}$
- The recipient uses the decoding algorithm Dec_C to decode c' to m'
- Then $m = m$'S^{-1} is the original message

Let us note that $c' = cP^{-1} = mG'P^{-1} + eP^{-1} = mSG + eP^{-1}$. Since P is a permutation matrix, the weight of eP^{-1} is equal to t. Seeing that Dec_C can correct up to t errors and mSG can be at a distance up to t from cP^{-1}, the correct codeword m' $= mS$ will be obtained. Now the original message can be obtained easily by multiplying the codeword with the inverse of S: $m = m'S^{-1} = mSS^{-1}$.

4.3 Security of McEliece Cryptosystem

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.

If an attacker got hold of an encrypted message c, he would have two possibilities in order to retrieve the original message m:

1. Find out G knowing G';
2. Decode c without knowing an efficient decoding algorithm.

Attacks of the first type are called structural attacks, while attacks of the second type are called decoding attacks. The security of the McEliece cryptosystem is suggested by the NP-hard general decoding problem.

General decoding problem of linear codes. Give an $[n, k]$-linear code C and a codeword $y \in \mathbb{F}^n$, find a codeword $c \in C$ such that the distance $d(y, c)$ is minimal.

Information-set decoding attack proposed by McEliece in his original paper solves the NP-hard general decoding problem, but the attack runs in exponential time. However, the same basic idea is used to construct attacks that are more efficient. Classic McEliece submission to NIST's post-quantum program has addressed this threat as well as other possible attacks by choosing proper parameters and adding other various improvements.

5. Hash-based algorithms

Hash-based algorithms are quite different from the 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. [2]

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. Hash-based algorithms are also very fast, because the only calculations required are the computations of the underlying hash function. [18]

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. [2]

In order for a hash function $H(X)$ to be suitable for creating secure signature schemes, it must possess certain properties. For example, cryptographic hash function must be a one-way function – given a random output $Y = H(X)$ it should be hard to find an input X, which would satisfy $H(X) = Y$. This property is also called **pre-image resistance**. A similar property required of cryptographic hash functions is **second-preimage resistance** – given a random input X, it should be difficult to find a different input X' that would produce the same hash (such that $H(X) = H(X')$). [19]

One of the most important properties of cryptographic hash functions is **collision resistance.** It is quite similar to second-preimage resistance, in fact second-preimage resistance is often called weak collision resistance. Collision resistance insures that it is computationally infeasible to find two different inputs that produce the same output. Collision resistance is important because collisions pose a serious security risk. For example, if an attacker found a message that produces the same hash as another authentic digitally signed message (*a collision*), they could easily exchange the original message with the fake one, while still keeping the same signature value. It would be impossible to distinguish them when verifying the signature. [20]

The following chapters are based on multiple works by Daniel J. Bernstein *et al.* [21, 22] and an article by Matthew Green [19].

5.1 Hash-based signatures

First hash-based signature scheme was introduced by Leslie Lamport in 1979.

Given a 256-bit cryptographic hash function and a secure random bit generator, the Lamport signature scheme can be used as described below.

- **Key pair generation.** In order to create a private key, Alice needs to generate two sets of 256 random 256-bit bit strings (*i.e.* random numbers). These 512 values are her private key. To create her public key, she needs to hash all 512 values from her secret key. These two sets of 256 hashes (512 hashes in total) form her public key.
- **Signing a message.** If Alice wants to sign a message, she first needs to hash the message to a 256-bit hash. Then, for each bit in the hash, Alice picks one number from the corresponding set of random numbers that make up her private key, based on the value of the bit. For example, if the bit is 0, she chooses the corresponding random number from the first set, and if the bit is 1, she needs to choose the corresponding number from the second set.
- **Verifying the signature.** When Bob needs to verify Alice's signature on a message he received, he also first has to hash the message to a 256-bit hash sum. Then he picks 256 hashes from Alice's public key based on the hash sum, exactly in the same manner that Alice picked numbers for her signature – if the bit from the message is 0, he needs to pick the hash from the first set of the public key, and if the bit is 1, he picks the hash from the second set of the public key.

  In order to verify Alice's signature, Bob hashes all of the numbers in Alice's signature. If every hash out of these 256 hashes match all the 256 hashes he picked from the public key earlier, then he can be sure that the signature is valid.

These are the main principles of Lamport signature scheme. Evidently, the main downside of this scheme is that one can only use the generated key pair once, which is why it is called a one-time signature scheme (OTS). If we were to use the same private key twice, we would reveal both secret key values for some of the positions. An attacker could use this knowledge to forge our signature. In addition, the keys and signatures in Lamport signature scheme are quite large in size.

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 (called a Merkle tree) to authenticate 2^h one-time signature key pairs. He places the public keys of the Lamport signature schemes into the leaves of the tree. All the other non-leaf nodes are hash values of the concatenation of its children, thus the root of the tree becomes the public key for all of the signatures and one-time signature secret keys become the secret key of the new scheme. Using this scheme allows to sign 2^h messages. The signer has to retain all of the Lamport secret keys for signing.

To sign a message, the signer selects an unused public key from the tree and signs the message using the corresponding Lamport secret key. In addition to the Lamport signature, the signature in Merkle's scheme also holds the corresponding Lamport public key and something called a Merkle proof. Merkle proof ensures, that the specific Lamport public key belongs to the tree identified by the root.

Merkle proof is a way of making sure that the given data belongs to a Merkle tree, without having to provide the full tree. Since all the non-leaf nodes are just hashes of their children nodes, we only need to provide the siblings of the nodes that belong to the path from the chosen leaf to the root.

This results in a few-time signature scheme (FTS), which can be used to generate a small amount of signatures. A few-time signature schemes can be extended to increase the number of signatures, creating many-time signature schemes (MTS).

Merkle's idea has since been used in many signature schemes. After more than 40 years of research, eXtended Merkle Signature Scheme (XMSS) was introduced. 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. As an important addition, SPHINCS uses randomized index selection – the index of the Merkle tree leaf containing an OTS key pair is chosen randomly, instead of applying a hash function to the message to determine the index.

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+.

5.2 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.

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.

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

A more detailed description of SPHINCS and SPHINCS+ is beyond the scope of this work.

6. Multivariate polynomial based 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, four of which were selected for further examination and proceeded to Round 2.

The main downside of multivariate polynomial cryptosystem is their newness. Most of the research during the inception of multivariate cryptography was conducted in Japan and thus most of the earlier publications are only available in Japanese. The amount of research of multivariate cryptography has grown since, but much more time is needed to prove its security. [23]

This chapter is based on a paper by Jintai Ding and Bo-Yin Yang [23] and multiple submissions to NIST’s post-quantum standardization problem, such as HIMQ-3 [24] and Rainbow [25].

6.1 Multivariate quadratic polynomials

Multivariate polynomial cryptography relies on the difficulty of solving systems of multivariate polynomials over finite fields. Most of the multivariate cryptosystems use quadratic polynomials and rely on the NP-hard $\mathcal{MQ}$ problem. $\mathcal{MQ}$ problem consist of solving a multivariate quadratic equation system over a finite field – given coefficients y_k, $a_{ij}^{(k)}$, $b_i^{(k)}$ and $c^{(k)}$ find a solution $(x_1, \ldots, x_n)$ for

$$f_1(x_1, \ldots, x_n) = y_1 = \sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij}^{(1)} x_i x_j + \sum_{i=1}^{n} b_i^{(1)} x_i + c^{(1)}$$

$$\vdots$$

$$f_1(x_1, \ldots, x_n) = y_m = \sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij}^{(m)} x_i x_j + \sum_{i=1}^{n} b_i^{(m)} x_i + c^{(m)}$$

where n is the number of variables, m is the number of equations, k is in range $1 \ldots m$ and coefficients are all elements in finite field $\mathbb{F}$.

6.2 Construction of the Rainbow scheme

In this paper, the multivariate quadratic (MQ) signature scheme Rainbow is introduced. The general structure of Rainbow over $\mathbb{F}_\mathrm{q}$ is as follows.

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

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

For $k = 1, \ldots, m$ and $p_{ij}^{(k)}, p_i^{(k)}, p_0^{(k)} \in_R \mathbb{F}_\mathrm{q}$.

The main idea for key generation in a MQ-signature scheme is to choose a central map $\mathcal{F} = \left(\mathcal{F}^{(1)}, \ldots, \mathcal{F}^{(m)}\right) : \mathbb{F}_\mathrm{q}^n \to \mathbb{F}_\mathrm{q}^m$ of multivariate quadratic polynomials, which can be easily inverted. After that two affine or linear invertible maps $S : \mathbb{F}_\mathrm{q}^m \to \mathbb{F}_\mathrm{q}^m$ and $T : \mathbb{F}_\mathrm{q}^n \to \mathbb{F}_\mathrm{q}^n$ 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 $\mathcal{P}$.

Generating a signature

In order to sign a document d a hash function $\mathcal{H}: \{0,1\} \to \mathbb{F}^m$ is used to compute the hash value $h = \mathcal{H}(d) \in \mathbb{F}^m$. The signature z is generated as follows.

- $x = S^{-1}(h) \in \mathbb{F}^m$ is computed.
- A pre-image of x is computed under the central map $\mathcal{F}$, resulting in y. This pre-image is computed using a special algorithm, which takes x and the central map $\mathcal{F}$ as arguments and returns a vector $y \in \mathbb{F}^n$ which satisfies $\mathcal{F}(y) = x$.
- Signature $z \in \mathbb{F}^n$ is then computed: $z = T^{-1}(y)$

Verifying a signature

Given a document d and signature z, in order to verify the signature, the hash value of the document has to be computed first: $h = \mathcal{H}(d) \in \mathbb{F}^m$. Then $h' = \mathcal{P}(z) \in \mathbb{F}^m$ is computed. If $h' = h$ holds, the signature z is valid.

The figure 4 below illustrates the process of generating and verifying a signature.

Signature generation

$$\begin{tikzcd}
h \in \mathbb{F}^m \arrow[r, "\mathcal{S}^{-1}"] & x \in \mathbb{F}^m \arrow[r, "\mathcal{F}^{-1}"] & y \in \mathbb{F}^n \arrow[r, "\mathcal{T}^{-1}"] & z \in \mathbb{F}^n \arrow[lll, bend left, "\mathcal{P}"]
\end{tikzcd}$$

Signature verification

Figure 4. The process of generating and verifying a signature using the MQ-signature scheme.

The Rainbow scheme includes some improvements to the basic algorithm introduced above, in order to increase security and speed. For example, to facilitate computations, some of the $p_{ij}^{(k)}$ coefficients chosen at random can be set to zero.

6.3 Security of the Rainbow scheme

Security analysis of multivariate schemes such as Rainbow is rather difficult, since no direct reduction from a NP-hard problem exists. Since there is no proof of the practical security for Rainbow, the parameter choice is crucial. In the Rainbow documentation, parameters are chosen in a way that the complexities of known attacks are beyond the levels of security required by NIST.

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, multivariate cryptosystems remain as very strong competitors for the potential quantum-resistant digital signature standards.

7. Conclusion

All of the four main families have their merits and deficits. 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 been subject to 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. Based on the number of submissions, lattice-based algorithms seem to be favoured the most.

The NIST's post-quantum standardization program gives a good 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.

8. References

[1] Greenemeier L. How Close Are We Really to Building a Quantum Computer? Scientific American, 2018. https://www.scientificamerican.com/article/how-close-are-we-really-to-building-a-quantum-computer/ (14.01.2019)

[2] Chen L, Jordan S, Liu Y K, Moody D, Peralta R, Perlner R, Smith-Tone D. NISTIR 8105 Report on Post-Quantum Cryptography. National Institute of Standards and Technology, 2016. https://doi.org/10.6028/NIST.IR.8105 (19.12.2018)

[3] Schwartz M J. Post-Quantum Crypto: Don't Do Anything. Bank Info Security, 2017. https://www.bankinfosecurity.com/quantum-crypto-dont-do-anything-a-9737 (13.01.2019)

[4] NIST. Post-Quantum Cryptography. https://csrc.nist.gov/Projects/Post-Quantum-Cryptography (27.03.2019)

[5] Alwen J. What is Lattice-based cryptography & why should you care, 2018. Medium. https://medium.com/cryptoblog/what-is-lattice-based-cryptography-why-should-you-care-dbf9957ab717 (27.03.2019)

[6] Lipmaa H. Lecture in Cryptographic Protocols: Lattice-Based Cryptography, 2013. University of Tartu. https://courses.cs.ut.ee/MTAT.07.014/2013_fall/uploads/Main/CP2013.L15.lattices.pdf (27.03.2019)

[7] Bergami F. Lattice-Based Cryptography, 2016. Universite de Bordeaux. https://www.math.u-bordeaux.fr/~ybilu/algant/documents/theses/BERGAMI.pdf (27.03.2019)

[8] Regev O. Introduction to Lattices, 2012. Winter School on Lattice-Based Cryptography and Applications. Bar-Ilan University. https://cyber.biu.ac.il/wp-content/uploads/2017/01/slides-barilan1-2.pdf (27.03.2019)

[9] Regev O. On Lattices, Learning with Errors, Random Linear Codes, and Cryptography, 2009. Tel-Aviv University. https://cims.nyu.edu/~regev/papers/qcrypto.pdf (27.03.2019)

[10] Chi D P, Choi J W, Kim J S, Kim T. Lattice Based Cryptography for Beginners, 2015. IACR Cryptology ePrint Archive. https://eprint.iacr.org/2015/938.pdf (27.03.2019)

[11] Regev, O. The Learning with Errors Problem, 2010. https://cims.nyu.edu/~regev/papers/lwesurvey.pdf (27.03.2019)

[12] Lyubashevsky V, Peikert C, Regev O. On Ideal Lattices and Learning with Errors Over Rings, 2010. EUROCRYPT 2010. https://eprint.iacr.org/2012/230.pdf (27.03.2019)

[13] Alkim E, Bos J W, Ducas L, Patrick L, Mironov I, Naehrigg M, Nikolaenko V, Peikert C, Raghunathan A, Stebila D. FrodoKEM. Learning With Errors Key Encapsulation, 2019. https://frodokem.org/files/FrodoKEM-specification-20190330.pdf (24.04.2019)

[14] Sendrier, N. Code-Based Cryptography: State of the Art and Perspectives, 2017. https://ieeexplore.ieee.org/document/8012331 (27.03.2019)

[15] McEliece R J. A Public-Key Cryptosystem Based on Algebraic Coding theory, 1978. The Deep Space Network Progress Report. https://tmo.jpl.nasa.gov/progress_report2/42-44/44N.PDF (27.03.2019)

[16] Bernstein D J, Chou T, Lange T, von Maurich I, Misoczki R, Niederhagen R, Persichetti E, Peters C, Schwabe P, Sendrier N, Szefer J, Wang W. Classic McEliece: conservative code-based cryptography, 2017. https://classic.mceliece.org/nist/mceliece-20171129.pdf (27.03.2019)

[17] Siim S. Study of McEliece cryptosystem, 2015. University of Tartu. https://courses.cs.ut.ee/MTAT.07.022/2015_spring/uploads/Main/sander-report-s15.pdf (18.03.2019)

[18] Trail of Bits. A Guide to Post-Quantum Cryptography, 2018. https://blog.trailofbits.com/2018/10/22/a-guide-to-post-quantum-cryptography/ (10.04.2019)

[19] Green M. Hash-based Signatures: An illustrated Primer, 2018. https://blog.cryptographyengineering.com/2018/04/07/hash-based-signatures-an-illustrated-primer/ (10.04.2019)

[20] Pigeon D. Cryptographic Hash Functions Explained: A Beginner’s Guide, 2018. https://komodoplatform.com/cryptographic-hash-function/ (10.04.2019)

[21] Bernstein D J, Hopwood D, Hülsing A, Lange T, Niederhagen R, Papachristodoulou L, Schwabe P, O'Hearn Z W. SPHINCS: practical stateless hash-based signatures, 2015. https://sphincs.cr.yp.to/sphincs-20150202.pdf (01.05.2019)

[22] Bernstein D J, Dobraunig C, Eichlseder M, Fluhrer S, Gazdag S-L, Hülsing A, Kampanakis P, Kölbl S, Lange T, Lauridsen M M, Mendel F, Niederhagen R, Rechberger C, Rijneveld J, Schwabe P. SPHINCS+ Submission to the NIST post-quantum project, 2017. https://cryptojedi.org/papers/sphincsnist-20171130.pdf (01.05.2019)

[23] Ding J, Yang B-Y. Multivariate Public Key Cryptography, 2009. https://link.springer.com/content/pdf/10.1007/978-3-540-88702-7_6.pdf (01.05.2019)

[24] Shim K-A, Park C-M, Kim A. HiMQ-3: A High Speed Signature Scheme based on Multivariate Quadratic Equations, 2017. NIST's post quantum project.

[25] Ding J. Rainbow Documentation, 2017. NIST's post quantum project.

Appendix

I. Licence

Non-exclusive licence to reproduce thesis and make thesis public

I, **Helen Tera**,

1. herewith grant the University of Tartu a free permit (non-exclusive licence) to reproduce, for the purpose of preservation, including for adding to the DSpace digital archives until the expiry of the term of copyright,

   Introduction to Post-Quantum Cryptography in scope of NIST's Post-Quantum Competition,

   supervised by Dominique Unruh .

2. I grant the University of Tartu a permit to make the work specified in p. 1 available to the public via the web environment of the University of Tartu, including via the DSpace digital archives, under the Creative Commons licence CC BY NC ND 3.0, which allows, by giving appropriate credit to the author, to reproduce, distribute the work and communicate it to the public, and prohibits the creation of derivative works and any commercial use of the work until the expiry of the term of copyright.
3. I am aware of the fact that the author retains the rights specified in p. 1 and 2.
4. I certify that granting the non-exclusive licence does not infringe other persons' intellectual property rights or rights arising from the personal data protection legislation.

Tartu, **09.05.2019**