HAL
open science

Performance Evaluation of Post-quantum Public-Key Cryptography in Smart Mobile Devices

Noureddine Chikouche, Abderrahmen Ghadbane

To cite this version:

Noureddine Chikouche, Abderrahmen Ghadbane. Performance Evaluation of Post-quantum Public-Key Cryptography in Smart Mobile Devices. 17th Conference on e-Business, e-Services and e-Society (I3E), Oct 2018, Kuwait City, Kuwait. pp.67-80, 10.1007/978-3-030-02131-3_9 . hal-02274171

HAL Id: hal-02274171

https://inria.hal.science/hal-02274171v1

Submitted on 29 Aug 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Distributed under a Creative Commons Attribution 4.0 International License

Performance Evaluation of Post-Quantum Public-Key Cryptography in Smart Mobile Devices

Noureddine Chikouche[1,2][0000−0001−9653−6608] and Abderrahmen Ghadbane[2]

[1] Laboratory of Pure and Applied Mathematics, University of M'sila, Algeria
[2] Computer Science Department, University of M'sila, Algeria

Abstract. The classical public-key schemes are based on number theory, such as integer factorization and discrete logarithm. In 1994, P.W. Shor proposed an algorithm to solve these problems in polynomial time using quantum computers. Recent advancements in quantum computing open the door to the possibility of developing quantum computers sophisticated enough to solve these problems. Post-quantum cryptography (PQC) is resistant against quantum attacks. The aim of this paper is to evaluate the performance of different post-quantum public-key schemes for constrained-resources smart mobile devices; and to give a comparison between the studied post-quantum schemes in terms of computational time, required memory, and power consumption.

Keywords: Post-quantum cryptography · public-key encryption · public-key signature · performance · mobile devices.

1 Introduction

The classical public-key algorithms used today to secure user data and networking communications (e.g. Internet, mobile, etc.) are based on number theory. For example, the RSA cryptosystem is based on integer factorization problem, and the Diffie-Hellman scheme is based on discrete logarithm problem. In 1994, P.W. Shor [31] proposed an algorithm to solve these problems in polynomial time using quantum computers.

In 2015, the National Security Agency (NSA) [27] announced that it is working with several partners to develop quantum-resistant encryption algorithms. In 2016, the NIST (National Institute of Standards and Technology) [29] has started the process of developing, evaluating, and standardizing one or more public-key post-quantum cryptographic algorithms. It's crucial to re-evaluate the existing cryptographic schemes which are used to protect information, and to improve quantum-safe cryptography.

The quantum computing is not a future project but it already exists in the real-world. The D-Wave Systems Inc. [11] was the first company that commercialises products based on quantum computing principles that are being used by some of the important advanced organizations, including Google and NASA

Ames. In 2017, this company developed the D-Wave 2000Q system which is a quantum annealer that has up to 2048 qubits and 5600 couplers. It can solve larger problems in various areas, such as machine learning, financial analysis, optimization and security. In security, this quantum computer can carry out factoring integers, detects computer viruses and network intrusion.

Nowadays, there are various classes of post-quantum public-key cryptography, including:

- **Code-based cryptography:** The classic example is McEliece cryptosystem based on Goppa codes [25]. It uses error correcting codes to generate public-key from private matrices with intentionally added errors. It is employed in the construction of diverse cryptographic schemes and it does not need any cryptographic processor.
- **Lattice-based cryptography:** The most basic lattice problem is the shortest vector problem (SVP), given an arbitrary basis of lattice, the goal is to find the shortest nonzero vector in it. NTRU scheme [19] is one of the most interesting lattice-based variants.
- **Multivariate-based cryptography:** It uses a set of multivariate polynomial equations that are based on the multivariate quadratic (MQ) Problem. One of its many interesting schemes is Rainbow public-key signature scheme which was proposed by J. Ding and D. Schmidt [13] in 2005.
- **Hash-based cryptography:** It is based on so-named one-time signature (OTS), a single key pair must only be used once. it requires a cryptographic hash function to create a public-key signature. For example, Winternitz one-time signature (W-OTS) [14] that relies on collision resistance, which means that using the same private-key to sign multiple documents will not yield a similar signature.
- **Isogeny-based cryptography:** This category was introduced as a solution to breaking elliptic curve cryptography by Shor's algorithm. Isogeny problem is to find the isogeny mapping between two elliptic curves with the same number of points. NIST [6] stated that not enough analysis has been done prove the claimed security.

Post-quantum cryptography (PQC) is resistant against quantum attacks and its computational complexity is of type NP-hard problem. Recently, there are several security protocols based on PQC have been presented, such as [24,8,9,7]. In the context of smart mobile technology, to secure user data and mobile communication, it is crucial to implement efficient cryptographic primitives. In the other hand, the most important problem is the limitation of resources, storage, processing, and power.

The aim of this paper is to survey the post-quantum public-key algorithms in regards to their efficient in smart mobiles. We evaluate their performance in terms of computational time, required memory, and power consumption. Moreover, we compare the different PQC in terms of performance and security.

The rest of this paper is structured as follows: Section 2 presents related works. Section 3 presents post-quantum encryption schemes. Section 4 presents

post-quantum signature schemes. The experimental results is detailed in Section 5. We discuss of obtained results in Section 6. Finally, conclusion has been presented in Section 7 .

2 Related works

An important number of studies have been realised in order to present efficient implementation of post-quantum cryptography in constrained resource devices. In the rest of this section, we introduce some research efforts in this area.

Eisenbarth et al. [15] investigated the efficient software implementation of McEliece scheme on embedded systems, low-cost 8-bit AVR microprocessor and a Xilinx Spartan-3AN FPGA. Hayes [17] evaluated different implementation possibilities for McEliece, Niederreiter, and their variants. In addition, He evaluated the performance of the schemes using various types od codes on smartcard class microcontrollers and a range of FPGAs. Wang et al. [34] improved the previous implementations of Niederreiter encryption scheme in terms of efficiency and security level by presenting a new implementation using binary Goppa codes in FPGA.

About the implementation of PQC on mobiles devices, Tayoub et al. [33] implemented NTRU scheme and other classical public-key schemes on Android mobiles, and evaluated their performances in terms of timing and memory occupation. In 2016, A. Boledovič and J. Varga [4] implemented McEliece encryption scheme in messenger application of Android operating system by using Bouncy castle provider.

3 Post-quantum encryption schemes

Post-quantum encryption scheme is used to safeguard the confidentiality of stored and exchanged information. It consists of three processes: key generation, encryption, and decryption. The key generation process creates a key pair consisting of a public and a private key. The public-key is used to encrypt a plaintext and the private-key, to decrypt a ciphertext. Various post-quantum encryption schemes subsequently designed are presented below.

3.1 McEliece scheme

In 1978, Robert J. McEliece [25] introduced the first public key cryptosystem (PKC) based on coding theory. The security of McEliece scheme is based on the problem of computational dual decoding syndrome. Let $\mathcal{C}[n, k, t]$ be a binary linear code, where n is length, k is dimension which stands as a generator matrix G. $\mathcal{C}$ can correct up to t errors.

The McEliece encryption scheme is defined as follows:

Key Generation:
- Generate three private matrices, a generator matrix $G' \in \mathbb{F}_2^{k\times n}$ of a binary Goppa code $\mathcal{C}$, a permutation matrix $P \in \mathbb{F}_2^{n\times n}$ and an invertible matrix $S' \in \mathbb{F}_2^{k\times k}$,
- Compute the public-key matrix $G = S'G'P$, which is another valid generator matrix,
- The private-key is $(S', G', P, \mathcal{A}(.))$, where $\mathcal{A}(.)$ is a polynomial-time decoding algorithm,
- The public-key is (G, t).

Encryption: To encrypt a message $m \in \mathbb{F}_2^k$
- Generate an error vector $e \in \mathbb{F}_2^n$ of weight $\mathsf{wt}(e) \leq t$,
- Compute the codeword $c \in \mathbb{F}_2^n$ where c is mG and the plaintext is $m \in \mathbb{F}_2^k$,
- The cryptogram $c' = c \oplus e$.

Decryption: To decrypt a cryptogram c'
- Compute $z = c'P^{-1}$,
- $y = \mathcal{A}(z)$,
- Output $m = yS'^{-1}$.

3.2 Niederreiter scheme

Niederreiter encryption scheme [28] introduced the dual version of McEliece encryption scheme. This variant is based on the syndrome decoding $SD-$ problem using the parity check matrix. The important advantage of this scheme compared to McEliece is reduction of the public-key size from $k \times n$ to $n \times (n-k)$.

The Niederreiter encryption scheme is defined as follows:

Key generation:
- **Parameters:** $n, t \in \mathbb{N}$, where $t < n$
- Generate a parity check matrix $H' \in \mathbb{F}_2^{(n-k)\times n}$ of a binary linear $\mathcal{C}$,
- Generate a permutation matrix $P \in \mathbb{F}_2^{n\times n}$,
- Generate an invertible matrix $Q \in \mathbb{F}_2^{(n-k)\times(n-k)}$,
- **private-key:** $(Q, H', P, \mathcal{A}(.))$ with $\mathcal{A}(.)$ a decoding algorithm until $\frac{d}{2}$ errors,
- **public-key:** $H \in \mathbb{F}_2^{(n-k)\times n} := QH'P$ and t integer $< \frac{d}{2}$.

Encryption: To encrypt message m
- decode m to error vector $e \in \mathbb{F}_2^n$ with $\mathsf{wt}(e) = t$,
- $c' := H^T e$,
- output ciphertext c'.

Decryption: To decrypt cryptogram c'
- $Q^{-1}c' := Q^{-1}QH'(Pe)$,
- compute $P^{-1}(Pe)$,
- encode e into message m.

3.3 McE Kobara-Imai scheme

Kobara and Imai [23] proposed modified versions of McEliece scheme that can be proven to be semantically secure against adaptive chosen-ciphertext attacks (CCA2). In addition, their conversion γ (Algorithm below) uses the entropy in the error vector, to decrease the overhead of data further.

The McE Kobara-Imai scheme (Kobara-Imai conversion γ for the McEliece scheme)is defined as follows:

Encryption: To encrypt a message m
- $r := Rand$,
- $y1 := Gen(r) \oplus (m \parallel Const)$ where $Gen(.)$ is a random number generator and $Const$ is a public constant,
- $y2 := r \oplus Hash(y1)$,
- $(y5 \parallel y4 \parallel y3) := (y2 \parallel y1)$,
- $z \leftarrow Conv(y4)$ where $Conv(.)$ a constant weight encoding function,
- Output the cryptogram $c := y5 \parallel y2G \oplus z$.

Decryption: To decrypt a cryptogram c
- $c := y5 \| c'$,
- $y3 := Decrypt^{Mceleice}(c')$,
- $y3G \oplus y0$,
- $y4 := Conv^{-1}(z)$,
- $(y2 \parallel y1) := (y5 \parallel y4 \parallel y3)$,
- $r := y2 \oplus Hash(y1)$,
- $(\tilde{x} \| Const') := y1 \oplus Gen(r)$,
- if $Const' == Const$ return $x := \tilde{x}$,
- else return $\perp$.

3.4 NTRU encryption scheme

The NTRU cryptosystem was published by Jeffrey Hoffstein and Jill Pipher and Joseph Silverman in 1998 [19] and it was standardised by IEEE in 2008. NTRU uses a public parameter N to specifies the size of the polynomials used, a large modulus q and a smaller modulus p. The sender creates a pair of a public and a private key by generating two polynomials f and g,where f is invertible modulo both p and q.

The NTRU encryption scheme is defined as follows:

Key Generation:
- **Parameters:** N,q,p, where N and p prime, $\gcd(p, q) = \gcd(N, q) = 1$
- randomly choose two private polynomials f and g in the ring
- **Private key:** consists of the polynomials f and $f_q := f^{-1}\ mod\ p$
- **Public key:** $h := p * f_q * g\ mod\ q$ where $f_q := f^{-1}\ mod\ q$

Encryption: To encrypt a message m
- generate a random polynomial r
- $c := p * r * h + m\ mod\ q$
- The cryptogram is the polynomial c

Decryption: To decrypt a cryptogram c
- $a := f * c$ where the coefficients of a lie between $q/2$ and $q/2$
- $b := a \ mod \ p$
- $m := f_p * b \ mod \ p$
- output m

4 Post-quantum signature schemes

The digital signature scheme is a cryptographic primitive that provides public-key message authentication. It consists of three processes, key generation, signature generation, and signature verification. The key generation process creates a key pair consisting of a public and a private key. The private-key is used to sign a document and to generate a signature when the public-key is used to verify this signature. In this section, we present four important post-quantum digital signatures: Niederreiter CFS, NTRUSign, Rainbow, and XMSS.

4.1 Niederreiter-CFS signature scheme

A signature scheme based on the Niederreiter encryption scheme was introduced by Courtois, Finiasz and Sendrier in [10]. The idea of the Niederreiter-CFS scheme is to frequently hash the message, randomized by a counter of bit-length i, until the output is a ciphertext that can be decrypted. To determine the error-vector, the signer uses his corresponding private key, and with the current value of the counter, the error vector will then serve as a signature. This signature scheme has the following parts:

The Niederreiter-CFS signature scheme is defined as follows:

Signature: To sign the document d
1. $i \leftarrow i + 1$,
2. $s' := \mathcal{A}(Q^{-1}h(h(d)\|i))$,
3. if no x' was found go to 1 else output $(i, x'P)$.

Verification: To verify the $(i, x'P)$
- $s' = Hx'T$,
- $s = h(h(d)\|i)$,
- If s' and s equals, then the signature is valid.

4.2 NTRU signature scheme

The NTRU Signature scheme, also known as NTRUSign was presented in [18], it's based on the GGH signature scheme. NTRUSign includes mapping a message to a random point in 2N-dimensional space, with N being a parameter, and solving the closest vector problem (CVP) in a lattice which is related to the NTRUEncrypt lattice problem.

The NTRU signature scheme is defined as follows:

Key Generation:
- pick two short polynomials h and f in ring R,
- find (F, G) with $f * g - g * F = q$,
- **Private key:** (f, g, F, G),
- **Public key:** $h := g * f^{-1}$ mod q with f is invertible in R_q.

Signature: To sing the document d
- $t := 0$
- Repeat
  1. $t := t + 1$,
  2. $\mu := Hash(d\|t) \in R_q$,
  3. $(x, y) := (0, \mu) \left(\begin{smallmatrix} G & F \\ -G & f \end{smallmatrix}\right)$,
  4. $s := -x * f - y * g$,
- until $\|(s, s * h - d)\| \leq N$,
- return (s, t).

Verification: To verify (d, s, t)
- $\mu := Hash(d\|t)$,
- if $\|(s, s * h - d)\| \leq N$ then the signature is valid.

4.3 Rainbow signature scheme

In 2005, J. Ding and D. Schmidt [13] proposed a public-key signature scheme named Rainbow, which is based on the idea of Oil and Vinegar variables. The idea of Oil and Vinegar variables is one way to create easily invertible multivariate quadratic systems.

The Rainbow signature scheme is defined as follows:

Key Generation:
- **Private key:** Two invertible affine maps L_1 and L_2 and the map $F = (f_{v1+1}(x), ..., f_n(x))$. The number of components of F is $m = n - v1$,
- **Public key:** The composed map $P(x) = L_1 \circ F \circ L_2$.

Signature: To sign a document d
- $h := hash(d)$,
- $x := L_1^{-1}(h)$,
- $y := F^{-1}(x)$ where $F^{-1}(x)$ means finding one pre-image of x,
- $z := L_2^{-1}(y)$,
- output: the signature z.

Verification: To verify z
- $h' := P(z)$,
- $h := hash(d)$,
- if $h' = h$ then the signature is valid.

4.4 XMSS Scheme

The eXtended Merkle Signature Scheme (XMSS) was proposed by Buchmann et al. [5] in 2011. It is a hash-based digital signature system that is a variant of the Merkle tree scheme and it is forward secure. Recently, IETF has published XMSS as informational RFC (RFC 8391).

The XMSS signature scheme is summarized as follows:

Key Generation:
- **Private key:** It consists of a cryptographic *seed* for a pseudorandom function, $PRNG$. Using $PRNG$ function, create the WOTS Key pair and the leaf index i corresponding to the next W-OTS keys to be used,
- **Public key:** It contains bitmasks and the root node value used in the transitional levels of the hash tree.

Signature:
- *input:* message M, the private-key sk and the index i,
- use the ith W-OTS key pair to sign ith message,
- The signature (σ, i, $Auth$) consists of the W-OTS signature σ, index i, and the leaf node authentication path $Auth$,
- the authentication path contains the hash values of H different nodes in the XMSS tree,
- the contained leaf index i in the XMSS private-key is updated.

Verification:
- input: a signature (σ, i, $Auth$), a message M and the XMSS public-key,
- verify the W-OTS signature σ using the corresponding W-OTS public-key,
- verify the authentication path by traversing the tree using $Auth$ to obtain P_H,
- If P_H is equal to the root node value in the public-key, the signature is valid.

5 Performance Evaluation

In this section, we present the developing environment of different post-quantum schemes in smartphone and the obtained experimental results.

5.1 Developing environment

Android smartphones are the most used mobiles. Android is an operating system based on Linux kernel and other open-source software; written in Java, C and other programming languages. It was developed by Google for mobile devices. In this work, we develop an PQC benchmark set with the studied schemes by using two cryptographic providers: FlexiProvider [16] and Spongy Castle [32] which is a repackage of Bouncy Castle for Android platform. This performance evaluation is based on the running time, required memory, and power consumption for each studied scheme.

For the evaluation of the timing and the memory usage, we used the methods of `java.lang.System` and `java.lang.Runtime` classes, respectively. To measure the energy consumption of the three processes of each scheme, we used Batterystats which is a tool included in the Android framework that collects battery data on smartphones.

We benchmarked the performance of the previously studied schemes on smartphone Samsung Galaxy A5, model SM-A500H which is equipped with Exynos 7880 Octa-core 1.9 GHz processor, an internal memory with 3072 MB capacity, and a battery with a capacity of 3000 mAh. The Android version installed was 6.0.1 Marshmallow.

5.2 Experimental results

NIST [6] recommends that all data that has to stay secure for more than 10 years should use a minimum of 128-bits security level. Table 1 describes different parameters used for each scheme with 128 security.

Table 1. Parameters for post-quantum schemes at the 128 bits security level

Scheme	Category	Parameter	Reference
McEliece McE Kobara-Imai Niederreiter Niederreiter CFS	Code-based	(n=4096,k=3604, t=41)	[2]
NTRU (Encry and Sign)	Lattice-based	APR2011_439	[22]
Rainbow	Multivariate-based	$v1$= 36, $o1$=21, $o2$=22	[30]
XMSS	Hash-based	xmss- sha2_10_256	[20]

Table 2 presents the running time (key generation, encryption and decryption) and energy consumption of different encryption schemes. We mention that the measure of consumption energy is for all parties of each scheme.

Table 2. Computation speed and consumption energy in encryption schemes

Scheme	Timing (ms)			energy (mAh)
	KeyGen	Encryption	Decryption	
McEliece	320313	11	364	18.3
McE Kobara-Imai	64478	134	463	6.3
Niederreiter	27046	157	475	0.6
NTRUEncry	4052	47	70	0.3

Table 3 presents the running time (key generation, signature generation and signature verification) and energy consumption of different digital signature schemes.

Table 3. Computation speed and consumption energy in signature schemes

Scheme	Timing (ms)			Energy (mAh)
	KeyGen	Signature	Verification	
Niederreiter-CFS	63115	>1h	>1h	-
NTRUSign	4052	262	491	6.3
Rainbow	855157	276	77	42
XMSS	352714	486	199	18

For each post-quantum scheme, the required space in terms of pair key, ciphertext (in encryption schemes) or signature (in signature schemes) and memory usage (RAM) is showed in Table 4.

Table 4. Required space for each scheme

	Scheme	pair key(kB)		ciphertext or	memory
		private key	public key	signature (Byte)	usage (MB)
Encryption	McEliece	1851.61	1802.05	512	33.57
	McE Kobara-Imai	256.29	218.264	512	9.20
	Niederreiter	40.074	216.744	62	2.76
	NTRUEncry	0.67	0.59	604	0.22
Signature	Niederreiter-CFS	40.07	216.74	-	-
	NTRUSign	0.67	0.59	604	49.01
	Rainbow	156.24	136.359	82	4.82
	XMSS	2.264	0.063	2499	5.61

6 Discussion

In this work, we evaluate three variants of schemes based on coding theory, McEliece,McE Kobara-Imai, Niederreiter, and Niederreiter CFS. McEliece encryption scheme has high-speed encryption process compared to other post-quantum schemes. Moreover, it is easy to implement as it does not need any cryptographic processor. However, McEliece scheme consumes an important amount of energy (18.3 mAh). McE Kobara-Imai which is the CCA2 secure conversion of McEliece scheme, led to slower results because of the added steps to the encryption procedure. Niederreiter scheme performs better than the McEliece one when it comes to the timing of the decryption process. It generates the lowest ciphertext sizes, 8 times smaller than any other encryption scheme, which make it the best scheme when storing or transmitting the ciphertext. Moreover, McE Kobara-Imai scheme and Niederreiter scheme consume less amount of energy. The Niederreiter-CFS signature was the worst performing scheme on the signing process, it used a lot of computational resources and therefore is not suitable for mobile devices.

In the other hand, code-based schemes requires large key sizes especially the McEliece scheme with an astounding 3.8 MB for a key pair. To avoid this limitation, quasi-cyclic and quasi-dyadic variants [1,26] were designed to offer much lower key sizes. In addition, the generation of pair key is slower than NTRU-Encry. We notice that the code-based cryptography has been recommended by Post-Quantum Crypto Project of Europe because it "has been studied since 1978 and has withstood attacks very well" [12].

When we compared the code-based schemes to the NTRUEncrypt scheme we found that NTRUEncrypt offers a balanced execution speeds for encryption and decryption processes and requires the lowest amount of memory in RAM when performing the operations of encryption and decryption. Moreover, the key length of NTRUEncrypt is smaller, which is suitable with limited resources of mobiles phones. In addition, NTRUEncrypt scheme uses the least amount of energy. NTRUSign scheme is fast in signature generation process and consumes less energy compared to other post-quantum signature schemes. However, NTRUSign is slow in signature verification and it occupies the biggest space of RAM memory (49MB).

The NTRU cryptosystem has been broken by recent attacks that use special structures of the rings used in those schemes. Recently, Bernstein et al. [3] proposed a new variant of NTRU, called "NTRU Prime" to avoid the weaknesses of NTRU, and proved that it is IND-CCA2 secure. They submitted this work to NIST's "Post-Quantum Cryptography Standardization Project".

The Rainbow signature scheme provided the shortest signature, 7 times smaller than the NTRUSign and 31 times smaller than the XMSS scheme. It is fast and fairly similar to the NTRUSign scheme in signing operation. Unfortunately, it is the worst scheme in terms of key pair generation time, key pair size, and consumption energy. In security, Rainbow scheme is still uncertain.

The verification speed of XMSS was good. It has small private- and public-key and consumes less energy. However, it provides the biggest signature among post-quantum signature schemes and its signing speed is very slow. Concerning the security of XMSS, Hülsing et al. [21] presented a multi-target attack against hash-based signature schemes like SPHINCS and XMSS.

Based on our experimental results and discussion, we summarized the advantages and the limitations of each tested scheme in 5.

7 Conclusion

In this work, we have presented results of performance benchmarks of implemented different post-quantum schemes on smart mobile devices. Based on the experimental results, we have compared the studied schemes from the viewpoints of computational time, required memory, and power consumption.

In code-based cryptography, the encryption operation is faster than other PQC and it seems to have the most important security, but it uses large keys. Recently, it was improved to generate small key sizes. The optimisation of de-

Table 5. Advantages and disadvantages of each studied scheme

Scheme	Advantages	Disadvantages
McEliece	– Very fast encryption process	– Very large key sizes – High memory usage and energy consumption
McE Kobara-Imai	– More secure McEliece variant	– Large key sizes
Niederreiter	– Lowest cipher text size	– Large key sizes
Niederreiter CFS		– Resource-intensive signature generation process – Large key sizes
NTRUEncrypt	– Very fast execution – Lowest key size – Lowest memory usage and energy consumption	– Somewhat large cipher text sizes
NTRUSign	– Very fast execution (signature generation) – Lowest key size	– High memory usage and energy consumption (signature generation)
XMSS	– Small key sizes	– Somewhat large signature sizes
Rainbow	– Very fast signature generation and verification – signature size	– Long key generation process – High energy consumption

cryption operation and the minimization of the consumption energy are further studies.

NTRUEncrypt and NTRUSign schemes gave the best results compared to other schemes, it should be noted that NTRUSign uses some improvement for

memory consumption when generating digital signatures. The Rainbow scheme is very fast and it generates the lowest signature sizes, but it takes too long to generate its key pair which lead to great energy consumptions. The XMSS scheme has lower key sizes, and acceptable signature generation speeds, however the generated signature is too big and could use some optimizations in that regard. The most posed problem in NTRU, Rainbow, and XMSS schemes is the security, there are various detected attacks in these schemes in the last years.

Finally, we mention that the selection of the post-quantum scheme depends on its performance, its security, and the context in which it will be used.

References

1. Berger T.P., Cayrel PL., Gaborit P., Otmani A.: Reducing Key Length of the McEliece Cryptosystem. In: Preneel B. (eds) Progress in Cryptology AFRICACRYPT 2009. LNCS, vol. 5580, pp. 77–97. Springer (2009)
2. Bernstein, D.J., Chou, T., Schwabe, P.: McBits: Fast constant-time code-based cryptography. In: Bertoni, G., Coron, J.S. (eds.) Cryptographic Hardware and Embedded Systems - CHES 2013. LNCS, vol. 8086, pp. 250–272. Springer (2013)
3. Bernstein, D.J., Chuengsatiansup, C., Lange, T., van Vredendaal, C.: NTRU Prime: Reducing attack surface at low cost (2017), `https://ntruprime.cr.yp.to/papers.html`
4. Boledovič, A., Varga, J.: Practical implementation of McEliece cryptosystem on Android. In: 16th Central European Conference on Cryptology (CECC 2016) (2016)
5. Buchmann, J., Dahmen, E., Hülsing, A.: XMSS - a practical forward secure signature scheme based on minimal security assumptions. In: Yang, B.Y. (ed.) Post-Quantum Cryptography. LNCS, vol. 7071, pp. 117–129. Springer (2011)
6. Chen, L., Jordan, S., Liu, Y.K., Moody, D., Peralta, R., Perlner, R., Smith-Tone, D.: Report on post-quantum cryptography (2016)
7. Chen, R., Peng, D.: A novel NTRU-based handover authentication scheme for wireless networks. IEEE Communications Letters **22**(3), 586–589 (2018)
8. Chikouche, N., Cherif, F., Cayrel, P.L., Benmohammed, M.: RFID authentication protocols based on error-correcting codes: A survey. Wireless Personal Communications **96**(1), 509–527 (Sep 2017)
9. Cho, J.Y., Griesser, H., Rafique, D.: A mceliece-based key exchange protocol for optical communication systems. In: Baldi, M., Quaglia, E.A., Tomasin, S. (eds.) Proceedings of the 2nd Workshop on Communication Security. LNEE, vol. 447, pp. 109–123. Springer, Cham (2018)
10. Courtois, N.T., Finiasz, M., Sendrier, N.: How to achieve a McEliece-based digital signature scheme. In: Boyd, C. (ed.) Advances in Cryptology — ASIACRYPT 2001. LNCS, vol. 2248, pp. 157–174. Springer (2001)
11. D-Wave Systems Inc.: The D-Wave 2000Q Quantum Computer: Technology Overview `http://www.dwavesys.com/` (2017)
12. Daniel, A., Lejla, B., et al.: Initial recommendations of long-term secure post-quantum systems. PQCRYPTO. EU. Horizon **2020 ICT-645622** (2015)
13. Ding, J., Schmidt, D.: Rainbow, a new multivariable polynomial signature scheme. In: Ioannidis, J., Keromytis, A., Yung, M. (eds.) Applied Cryptography and Network Security. LNCS, vol. 3531, pp. 164–175. Springer (2005)

14. Dods, C., Smart, N.P., Stam, M.: Hash based digital signature schemes. In: Smart, N.P. (ed.) Cryptography and Coding. LNCS, vol. 3796, pp. 96–115. Springer (2005)
15. Eisenbarth, T., Güneysu, T., Heyse, S., Paar, C.: MicroEliece: McEliece for embedded devices. In: Clavier, C., Gaj, K. (eds.) Cryptographic Hardware and Embedded Systems - CHES 2009. pp. 49–64. LNCS, Springer (2009)
16. FlexiProvider Homepage, https://www.flexiprovider.de/. Last accessed 2 March 2018
17. Heyse, S.: Post Quantum Cryptography: Implementing Alternative Public Key Schemes On Embedded Devices. Ph.D. thesis, Ruhr-University Bochum, Germany (2013)
18. Hoffstein, J., Howgrave-Graham, N., Pipher, J., Silverman, J.H., Whyte, W.: NTRUSign: digital signatures using the ntru lattice. In: Joye, M. (ed.) Topics in Cryptology — CT-RSA 2003. LNCS, vol. 1612, pp. 122–140. Springer (2003)
19. Hoffstein, J., Pipher, J., Silverman, J.H.: NTRU: a ring-based public key cryptosystem. In: Buhler, J.P. (ed.) Algorithmic Number Theory. LNCS, vol. 1423. Springer (1998)
20. Huelsing, A., Butin, D., Gazdag, S., Rijneveld, J., Mohaisen, A.: XMSS: eXtended Merkle Signature Scheme. RFC 8391 (May 2018), https://www.rfc-editor.org/rfc/rfc8391.txt
21. Hülsing, A., Rijneveld, J., Song, F.: Mitigating multi-target attacks in hash-based signatures. In: Cheng, C.M., Chung, K.M., Persiano, G., Yang, B.Y. (eds.) Public-Key Cryptography – PKC 2016. pp. 387–416. LNCS, Springer (2016)
22. Jarvis, K., Nevins, M.: ETRU: NTRU over the eisenstein integers. Designs, Codes and Cryptography **74**(1), 219–242 (Jan 2015)
23. Kobara, K., Imai, H.: Semantically secure McEliece public-key cryptosystems - conversions for mceliece pkc -. In: Kim, K. (ed.) Public Key Cryptography. LNCS, vol. 1992, pp. 19–35. Springer (2001)
24. Li, D., Chen, H., Zhong, C., Li, T., Wang, F.: A new self-certified signature scheme based on NTRUSing for smart mobile communications. Wireless Personal Communications **96**(3), 4263–4278 (Oct 2017)
25. McEliece, R.J.: A public-key system based on algebraic coding theory. Tech. Rep. DSN Progress Report 44, Jet Propulsion Lab (1978)
26. Misoczki R., Barreto P.S.L.M.: Compact McEliece Keys from Goppa Codes. In: Jacobson M.J., Rijmen V., Safavi-Naini R. (eds) Selected Areas in Cryptography. SAC 2009. LNCS, vol. 5867, pp. 37-6-392. Springer (2009)
27. National Security Agency: Cryptography Today (August 2015), https://www.nsa.gov/ia/programs/suitebcryptography/
28. Niederreiter, H.: Knapsack-type cryptosystems and algebraic coding theory. Problems Control Inform. Theory **15**(2), 159–166 (1986)
29. NIST: Post-Quantum Cryptography Standardization, https://csrc.nist.gov/projects/post-quantum-cryptography (2016)
30. Petzoldt, A., Bulygin, S., Buchmann, J.: Selecting parameters for the Rainbow signature scheme. In: Sendrier, N. (ed.) Post-Quantum Cryptography. LNCS, vol. 6061, pp. 218–240. Springer (2010)
31. Shor, P.: Polynomial-time algorithm for prime factorization and discrete logarithms on a quantum computer. In: Proceedings of the 35th Annual Symposium on Foundations of Computer Science. vol. 124 (1994)
32. Spongy Castle Homepage, https://rtyley.github.io/spongycastle/. Last accessed 2 March 2018

33. Tayoub, W., Somia, L., Chikouche, N.: Implementation of public-key cryptographic systems on embedded devices (case: Computation speed). In: The First International Symposium on Informatics and its Applications (ISIA'2014) (2014)
34. Wang, W., Szefer, J., Niederhagen, R.: FPGA-based Niederreiter cryptosystem using binary goppa codes. In: Lange, T., Steinwandt, R. (eds.) Post-Quantum Cryptography. LNCS, vol. 10786, pp. 77–98. Springer (2018)