IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received June 28, 2020, accepted July 16, 2020, date of publication July 31, 2020, date of current version August 14, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3013250

A Comparison of Security and Its Performance for Key Agreements in Post-Quantum Cryptography

FÁBIO BORGES, (Senior Member, IEEE), PAULO RICARDO REIS, AND DIOGO PEREIRA
National Laboratory for Scientific Computing, Petrópolis 25651-075, Brazil

Corresponding author: Fábio Borges (borges@lncc.br)

This work was supported in part by Inmetro, in part by Petrobras, in part by the National Council for Scientific and Technological Development (CNPq), in part by the Fundação de Amparo à Pesquisa do Estado do Rio de Janeiro (FAPERJ), and in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES).

ABSTRACT Nowadays, we are surrounded by devices collecting and transmitting private information. Currently, the two main mathematical problems that guarantee security on the Internet are the Integer Factorization Problem and the Discrete Logarithm Problem. However, Shor's quantum algorithm can easily solve both problems. Therefore, research into cryptographic algorithms that run in classical computers and are resistant to quantum computers is extremely necessary. This area is known as post-quantum cryptography and usually studies asymmetric cryptography. By means of asymptotic analysis, the purpose of this paper is to provide an evaluation of security and its performance for the types of cryptographic systems considered safe against quantum attacks in the second-round NIST Post-Quantum Standardization Process, namely isogeny cryptosystems based on supersingular elliptic curves, error correction code-based encryption system, and lattice-based ring learning with errors. We performed a security comparison of Key Agreements protocols based on these three post-quantum cryptographic primitives and compared them with Discrete Logarithm Problem and Integer Factorization Problem. The comparison of security and its performance is presented by security level, the former by complexity analyses to achieve theoretical minimum key sizes, and the latter by simulation to assess a practical performance comparison. In the complexity analysis, as we increase the security level and then the size of the cryptographic keys increases, techniques based on isogeny outperform all other post-quantum algorithms in relation to key sizes at practical security level. In the performance comparison, the results show that the code-based protocol presents the best results among the others.

INDEX TERMS Post-quantum cryptography, asymmetric cryptography, key agreement, complexity, algorithms, asymptotic analysis.

I. INTRODUCTION

Recently, Google claimed having achieved Quantum Supremacy [1], using a processor with programmable superconducting qubits to create quantum states on 53 qubits. This raises an important concern in cryptography: are we safe in face of what a Quantum Computer might do? Two algorithms for quantum computers from the decade of 1990 have made cryptologists rethink what kind of mathematical problems could protect our communications. The Shor's algorithm [2] is capable of solving the Integer Factorization Problem (IFP) and the Discrete Logarithm Problem (DLP) in polynomial time, while the algorithm proposed by Grover [3] can increase the speed in the search for cryptographic keys to the order of $\sqrt{n}$, where n is the dimension of the basis.

To mitigate this problem, over the years there have been developed a series of new algorithms that claims to be resistant against attacks done with quantum computers. Even though such algorithms are designed to resist quantum computers, they are running on classical computers. Modern communications rely mostly on classical algorithms, which tells us that it is needed to research for safer alternatives in this emerging quantum-era. For these reasons, cryptographic algorithms that run in a classical computer and resist quantum attack are known as post-quantum cryptography, for more information see [4].

In this paper, we use computational complexity and performance measurements to analyze three post-quantum

The associate editor coordinating the review of this manuscript and approving it for publication was Luis Javier Garcia Villalba.

key agreements cryptosystems that appear as promising candidates to cope with information security challenges in the near future: isogeny based on supersingular elliptic curves, coding theory, and lattice-based ring learning with errors key agreements protocol. From this analysis, we show security levels based on the key used. In short, each cryptographic algorithm is based on a hard mathematical problem know as cryptographic primitives and each problem might be solved by an algorithm with a certain complexity. Considering the best-known algorithm to solve problems of primitives used in post-quantum cryptography, we compute the key-length with the same level of security used for the classical cryptographic algorithms.

Suppose two devices need to agree on a secret cryptographic key without letting a potential attacker discover it. One way to do this is to previously set a key to use and secretly store it. In a scenario with few communicators, it might not seem a bad idea, but for a scenario like the whole internet, it is no longer possible to store a secret key for every pair of communicators. This is where key agreement protocols have their utility. There is no need to store secret keys, instead one can have a public key that can, together with a specific algorithm agree at a secret key in real time, without alerting the attacker.

We provide an analysis of security and its performance for the cryptographic systems that are candidates in the Post-Quantum Standardization Process held by National Institute for Standards and Technology (NIST). The computed key lengths are compared with NIST's recommendation of key sizes for the given security level. We also provide a performance analysis for the referred cryptographic systems.

In the next section, we present a brief overview of the non-quantum-resistant cryptographic primitives used in key agreement protocols and present algorithm complexity for the best-known attacks. The third section of the paper is dedicated to a discussion of quantum-resistant primitives, presenting an overview of the protocols, their security and the complexity for the best know attacks to such mathematical problems. In the fourth section, it is traced a comparison between all primitives shown before, where we focus on the evaluation of the key-length generated for each of the practical security levels recommended by the NIST and the performance of the referred algorithms. The concluding section presents our conclusions and some directions for further research.

II. NON-QUANTUM-RESISTANT CRYPTOGRAPHIC PRIMITIVES

To base our comparisons, we first present complexity analyses for the most used classical algorithms, which use primitives developed for classical computers, i.e., non-quantum-resistant primitives. The complexity analysis for non-quantum-resistant primitives were presented previously in [5].

In particular, we have two primitives being widely used in classical-world algorithms. The first is the Discrete Logarithm Problem (DLP) introduced in the key-agreement protocol known as Diffie-Hellman (DH) [6]. The second is the Integer Factorization Problem (IFP) that introduced the algorithm known as RSA to create digital signatures [7]. Several other cryptographic algorithms use these primitives [8].

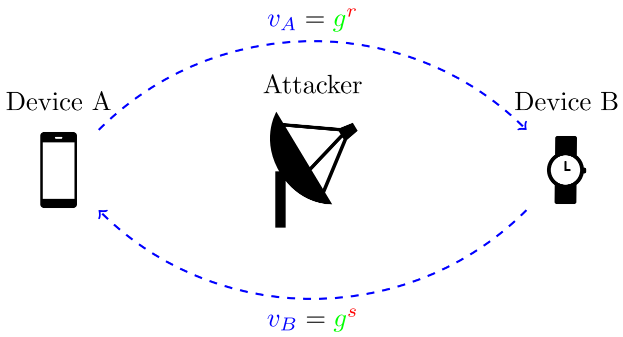

FIGURE 1. Picture illustrating the DH.

1) FOUNDATIONS

For two devices A and B agree on a secret key using the Diffie-Hellman protocol, they need to choose a large prime number p and a number g, such that $1 < g < p$ and g has a large order. These are parameters and are public. Now, device A chooses a secret integer r, computes $v_A = g^r \mod p$, and sends v_A to device B, which simultaneously does and analogous process, choosing a secret integer s, computing $v_B = g^s \mod p$ and sending v_B to device A. Hence, device A can use its secret and v_B to compute $S_k \equiv (g^s)^r \equiv g^{sr} \mod p$, and device B, using its secret and v_A, computes $S_k \equiv (g^r)^s \equiv g^{sr} \mod p$. Therefore, both have a secret key S_k that can be used to secure their communication. Figure 1 describe DH where the variables in green are parameters, in blue are public, and in red are private. Note that these key agreements consider an attack model whose attacker is passive.

One variety of DH is the Elliptic Curve based Diffie–Hellman (ECDH), that instead of using a multiplicative group of integers modulo a prime p, uses an additive group of points in an elliptic curve E defined over a general Galois field $\mathbb{F}$ whose characteristic is a prime power instead of only a prime. In this scenario, devices A and B agree publicly on E and on a base point $P \in E/\mathbb{F}$. Devices A and B generate random secret numbers $r, s \in \{1, \ldots, n-1\}$, respectively, where n is the order of the subgroup generated by P. Afterwards, device A computes $Q_A = rP$ and sends it to device B, that at the same time sends to device A the result of $Q_B = sP$. Both are now able to compute the secret key $Q_{AB} = (r+s)P = (s+r)P$ and use it as their security parameter. Figure 2 describes the ECDH. The variables in green are parameters, in blue are public, and in red are private.

These protocols are used in most of the communication systems. The RSA is still widely used in the internet, for instance, we can find the RSA in 99.7% of the digital certificates for Hypertext Transfer Protocol Secure (HTTPS) in

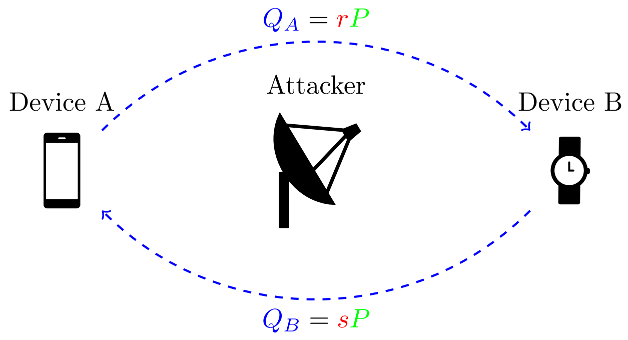

FIGURE 2. **Picture illustrating the ECDH.**

the Mediterranean [9]. The other 0.3% use a DLP based on elliptic curve cryptography, which was introduced simultaneously by Miller [10] and Koblitz [11]. This data confirms our dependency onto those classical primitives to safeguard our communication systems. In the following section, we analyze the complexity of attacking algorithms to those primitives in both classical and quantum computers.

2) COMPLEXITY

To solve the IFP in a classical computer, the best-known algorithm is the General Number Field Sieve (GNFS), which has a sub-exponential complexity, given by

$$\exp\left(\left(\left(\frac{64}{9}\right)^{1/3} + O(1)\right)(\ln n)^{1/3}(\ln \ln n)^{2/3}\right), \tag{1}$$

where n is the integer number being factored into primes.

In quantum computers, the Shor's algorithm can compute [12] in

$$(\log n)^{2+e}, \tag{2}$$

where e is the exponent of RSA. We can also use the GNFS to compute the factorization of $p-1$, which can be used to solve the DLP. In this case, e is the prime of the DH. However, Pollard's Rho algorithm for logarithms is the best-known algorithm that can be used to solve the DLP and Elliptic Curve DLP (ECDLP) in a classical computer. Its complexity is exponential and given by

$$\sqrt{\frac{\pi o}{2}}, \tag{3}$$

where o is the order of the group.

Proos and Zalk proposed a quantum algorithm [13] that can solve the ECDLP in

$$n^3, \tag{4}$$

where n is the input length in bits.

III. QUANTUM-RESISTANT CRYPTOGRAPHIC PRIMITIVES

This section presents the post-quantum algorithms for key agreement currently in the NIST Post-Quantum Standardization Process.

A. SUPERSINGULAR ISOGENY

The first proposal of using the search for isogenies between ordinary elliptic curves as a primitive was held by Rostovtsev and Stolbunov in 2006 [14]. They proposed a method of public-key construction. In 2011, Jao and Feo proposed a Diffie-Hellman protocol based on isogenies between supersingular elliptic curves (SIDH) [15]. Téllez and Borges [16] trace a security and performance comparison for this kind of cryptosystems.

The security of SIDH is based on the problem of finding isogenies between supersingular elliptic curves (SSI). Galbraith and Stolbunov [17] show that this problem is said to be hard and requires exponential time to be solved.

1) FOUNDATIONS

Let $E(\mathbb{F})$ be an elliptic curve in the Weirstraßform

$$E(\mathbb{F}) : y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6, \tag{5}$$

with all coefficients being elements of $\mathbb{F}$. If

$$\begin{cases} b_2 = a_1^2 + 4a_2 \\ b_4 = 2a_4 + a_1a_3 \\ b_6 = a_3^2 + 4a_6 \\ b_8 = a_1^2a_6 + 4a_2a_6 - a_1a_3a_4 + a_2a_3^2 - a_4^2 \\ c_4 = b_2^2 - 24b_4 \\ c_6 = -b_2^3 + 36b_2b_4 - 216b_6, \end{cases}$$

then the discriminant of $E(\mathbb{F})$ is defined by $\Delta = -b_2^2b_8 - 8b_4^3 - 27b_6^2 + 9b_2b_4b_6$ and the j-invariant of $E(\mathbb{F})$ as

$$j(E) = c_4^3/\Delta. \tag{6}$$

For fields with characteristic bigger than 3,

$$j(E) = 1728\frac{c_4^3}{c_4^3 - c_6^2}.$$

An isogeny is a rational map $\phi : E_A \to E_B$ between elliptic curves such that $\phi(O_{E_A}) = O_{E_B}$, where O_{E_A} and O_{E_B} are points at infinity, and ϕ is not trivial, i.e., there exists $P \in E_A$ such that $\phi(P) \neq O_{E_B}$.

The Supersingular Isogeny Problem consists of given a finite field $\mathbb{F}$ and two supersingular elliptic curves E_A and E_B defined over $\mathbb{F}$ such that $|E_A| = |E_B|$, compute an isogeny $\phi : E_A \to E_B$.

To devices A and B agree on a secret cryptographic key, both start with a public supersingular elliptic curve and after walking distinct paths onto the isogeny graph, end up at the same isogenous curve.

Small distinct primes l_A and l_B are fixed together with integers e_A and e_B, such that we can choose a number f that satisfies $p = l_A^{e_A}l_B^{e_B}f \pm 1$ is a prime. Fixing a supersingular curve E over $\mathbb{F}$, devices A and B are now able to proceed.

First, devices A and B pick bases $\{P_A, Q_A\}$ of $E[l_A^{e_A}]$ and $\{P_B, Q_B\}$ of $E[l_B^{e_B}]$. Hence, device A picks random integers $m_A, n_A \in \mathbb{Z}/l_A^{e_A}\mathbb{Z}$, not both divisible by l_A, and uses $\phi_A : E \to E_A$, where $E_A = E_0/\langle m_AP_A + n_AQ_A\rangle$, calculating $(\mathfrak{P}_A, \mathfrak{Q}_A) = (\phi_A(P_B), \phi_A(Q_B))$. Device A then sends $\mathfrak{P}_A, \mathfrak{Q}_A$ and E_A to device B.

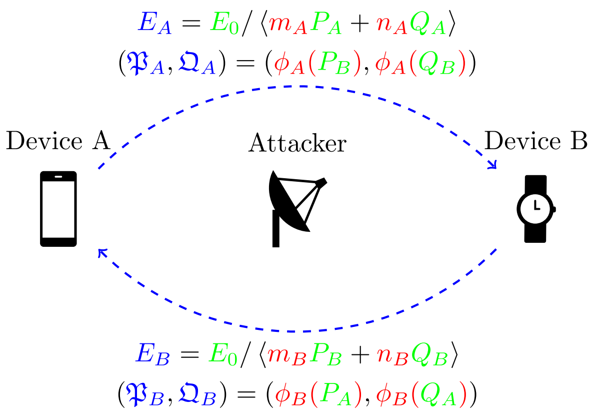
$E_A = E_0 / \langle m_A P_A + n_A Q_A \rangle$
$(\mathfrak{P}_A, \mathfrak{Q}_A) = (\phi_A(P_B), \phi_A(Q_B))$
Device A
Attacker
Device B
$E_B = E_0 / \langle m_B P_B + n_B Q_B \rangle$
$(\mathfrak{P}_B, \mathfrak{Q}_B) = (\phi_B(P_A), \phi_B(Q_A))$

FIGURE 3. Picture illustrating the SIDH.

The other device acts analogously picking random integers $m_B, n_B \in \mathbb{Z}/l_B^{e_B}\mathbb{Z}$, not both divisible by l_B, and uses $\phi_B : E \to E_B$, where $E_B = E_0/\langle m_B P_B + n_B Q_B\rangle$, calculating $(\mathfrak{P}_B, \mathfrak{Q}_B) = (\phi_B(P_A), \phi_B(Q_A))$. Device B then sends $\mathfrak{P}_B$, $\mathfrak{Q}_B$ and E_B to device A.

Now device A computes

$$E_{AB} = E_B / \langle m_A \phi_B(P_A) + n_A \phi_B(Q_A)\rangle$$

and $k = j(E_{AB})$; while device B computes

$$E_{BA} = E_A / \langle m_B \phi_A(P_B) + n_B \phi_A(Q_B)\rangle$$

and $k = j(E_{BA})$. This process leads to

$$ker\ \phi_{AB} = \langle m_A P_A + n_A Q_A, m_B P_B + n_B Q_B\rangle = ker\ \phi_{BA},$$

and $E_{AB} \simeq E_{BA}$. This way they find the shared secret key $j(E_{AB}) = j(E_{BA})$ computing the same j-invariant.

Figure 3 depicts the SIDH. The variables in green are parameters, in blue are public, and in red are private.

2) COMPLEXITY

Isogeny for regular curves can be computed in feasible time. The problem is to compute isogeny for supersingular elliptic curves.

Galbraith and Stolbunov [17] presented the best-known algorithm for classical computers to solve the isogeny problem [18]. Its classical-world supersingular isogeny (C-SSI) has complexity given by

$$p^{1/4}, \tag{7}$$

where p is the characteristic of the field.

Similarly, Tani's algorithm [19] is the best-known quantum algorithm to solve the isogeny problem [20]. Its quantum-world supersingular isogeny (Q-SSI) has complexity given by

$$p^{1/6}. \tag{8}$$

B. CODING

The first code-based cryptosystem scheme was the McEliece cryptosystem (MEC) [21], which is based on the NP-hard problem of decoding a general linear code, in fact, the general decoding problem for linear codes as well as the general problem of finding weights of a linear code are both NP-complete, as shown by Berlekamp *et al.* [22]). However, the MEC suffers from some disadvantages such as data expansion and size of the key. To circumvent the disadvantages of MEC, Misoczki *et al.* [23] proposed in 2013 a variation of MEC called MDPC-McEliece, based on Moderate Density Parity-Check codes. Also in 2013, Gaborit *et al.* [24] proposed a variation of MEC based on Low Rank Parity-Check codes.

The previous two schemes share the same security weakness, i.e., their security does not reduce to a well-known problem but to a specific problem. Thus, in this work, we will follow the approach proposed in [25], which benefits the interesting features of previous schemes and have a security reduction to decoding random quasi-cyclic code.

In short, the security of the scheme proposed by Deneuville *et al.* [25] comes from the Syndrome Decoding Problem (SDP), proven NP-hard [22]. However, the authors use quasi-cyclic codes, and the complexity of the Quasi-Cyclic Syndrome Decoding (QCSD) Problem is still unknown, although believed by the academic community to be also NP-hard.

1) FOUNDATIONS

Let $\mathbb{F}_q$ be a finite field, with $q = 2$, and let $\omega(\cdot)$ denote the Hamming weight of the vector, i.e., the number of non-zero coordinates of a vector. We define the set

$$\mathcal{S}_w^n(\mathbb{F}_2) = \{x \in \mathbb{F}_2^n \mid \omega(x) = w\}$$

as set of words $x \in \mathbb{F}_2^n$ of weight w. Let $\mathcal{V}$ a n-dimensional vector space over $\mathbb{F}_2$, for a positive integer n. The elements of $\mathcal{V}$ can be regarded as row vectors of polynomials in the ring $\mathcal{R} = \mathbb{F}_2[X]/(X^n - 1)$. Given two vectors $x, y \in \mathcal{V}$, their product is defined as $xy = c \in \mathcal{V}$ with

$$c_k = \sum_{i+j \equiv k \mod n} x_i y_j,$$

for $k \in \{0, 1, \dots, n-1\}$. The scheme [25] uses cyclic (circulant) matrices as in [26] and, given $h \in \mathcal{V}$, $rot(h)$ corresponds to the circulant matrix with $hX^i \mod X^n - 1$ as its i-th column.

Given positive integers s, n, k, and a linear code $\mathcal{C}$ with $[sn, k]$, if for every $c = (c_1, \dots, c_s) \in \mathcal{C}$, $(c_1 X, \dots, c_s X) \in \mathcal{C}$, then $\mathcal{C}$ is said to be Quasi-Cyclic (QC) of order s. In addition, if a QC code $[sn, n]$ of order s admits a parity-check matrix of the form

$$H = \begin{pmatrix} I_n & 0 & \cdots & 0 & A_1 \\ 0 & I_n & & & A_2 \\ & & \ddots & & \vdots \\ 0 & 0 & \dots & I_n & A_{s-1} \end{pmatrix},$$

the QC code $[sn, n]$ is said a systematic QC code.

Now, let n, k, w, and $s \in \mathbb{N}^*$. The s-QCSD distribution samples $H \xleftarrow{\$} \mathbb{F}_2^{(sn-k)\times sn}$, the parity-check matrix of a QC code of order s and $x = (x_1, \dots, x_s) \xleftarrow{\$} \mathbb{F}_2^{sn}$ such that $\omega(x_i) = w$, and produces (H, Hx^T) as output.

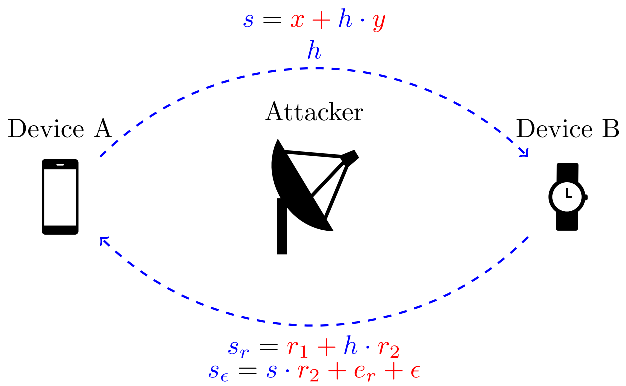
$s = x + h \cdot y$
h
Device A
Attacker
Device B
$s_r = r_1 + h \cdot r_2$
$s_\epsilon = s \cdot r_2 + e_r + \epsilon$

FIGURE 4. Picture illustrating the CODH.

The (search) s-QCSD problem corresponds to find $x = (x_1, \ldots, x_s) \in \mathbb{F}_2^{sn}$ such that $\omega(x_i) = w$, for all $i = 1 \ldots s$ and $y = xH^T$.

The Cyclic Error Decoding (CED) problem corresponds to an instance of the SDP on $H = (rot(x)^T, rot(y^T))$ and is define as follows. Let x, y, r_1, and r_2 be random vectors of length n and weight $w = \mathcal{O}(\sqrt{n})$. Let e be a random error vector of weight $w_e = cw$ (where c is a non-negative constant). Given $(x, y) \in \mathcal{S}_w^n(\mathbb{F}_2) \times \mathcal{S}_w^n(\mathbb{F}_2)$ and $e_c \leftarrow x \cdot r_2 - y \cdot r_1 + e$ such that $\omega(r_1) = \omega(r_2) = w$, the problem asks to recover (r_1, r_2).

Suppose devices A and B want to exchange a secret for an insecure channel using the protocol proposed by Deneuville *et al.* [25]. First, the protocol which uses a hash function $f_w : \{0, 1\}^* \to \mathcal{S}_w^n(\mathbb{F}_2)$.

Given a random vector $h \in \mathbb{F}_2^n$, device A constructs a random syndrome s from its secret $(x, y) \in \mathcal{S}_w^n(\mathbb{F}_2) \times \mathcal{S}_w^n(\mathbb{F}_2)$ by calculating.

$$s = x + h \cdot y$$

and send s and h to device B, that constructs its own random syndrome

$$s_r = r_1 + h \cdot r_2$$

from random $(r_1, r_2) \in \mathcal{S}_w^n(\mathbb{F}_2) \times \mathcal{S}_w^n(\mathbb{F}_2)$. It also constructs a syndrome

$$s_\epsilon = s \cdot r_2 + e_r + \epsilon,$$

associated to r_2, $\epsilon \in \mathcal{S}_{w_\epsilon}^n(\mathbb{F}_2)$ (shared secret) and e_r (obtained from the secret r_1, r_2). Device B then, sends s_r and s_ϵ to device A, which receives s_r and s_ϵ, and computes

$$e_c = s_\epsilon - y \cdot s_r = x \cdot r_2 - y \cdot r_1 + e_r + \epsilon,$$

which is the CED problem with $e = e_r + \epsilon$. Device A resolves the CED problem to recover (r_1, r_2) and the shared secret ϵ removing e_r from e_c.

Figure 4 describes the Code-Based Diffie–Hellman (CODH). The variables in blue are public, and in red are private.

2) COMPLEXITY

In the protocol described in Section III-B1, where $w \ll n$, thus, all known attacks [25] reduce the complexity of the classical Information Set Decoding (ISD) [27]. For the quantum-world, it is enough to take the square root of complexity for the classical case, corresponding to a straightforward application of the Grover algorithm [3].

According to Téllez *et al.* [28], the best approach to the ISD is a variation of the Stern's algorithm developed by Bernstein *et al.* [29]. Thus, according to previous researches [30] the complexity of the classical-world complexity to solve ISD (C-CODE) for a code of length n and $w = \mathcal{O}(\sqrt{n})$ corresponds to

$$2^{0.05564n}, \tag{9}$$

where n corresponds to the code length. Therefore, applying the Grover algorithm on classical-world complexity, we obtain the quantum-world complexity to solve ISD (Q-CODE), which is given by

$$2^{0.02782n}. \tag{10}$$

C. RING LEARNING WITH ERRORS

Given a lattice L and its basis, the Shortest Vector Problem (SVP) consists of finding the shortest non-zero vector in the lattice. For many cryptographic applications, it is enough to find a reasonably short vector. This can be achieved using an approximation factor $\gamma > 1$. Now, the approximate SVP$_\gamma$ consists of finding a non-zero vector $v \in L$ with length at most $\gamma \cdot \lambda(L)$, where the length of the shortest non-zero vector in L is $\lambda(L) = \min_{v \in L \setminus \{0\}} ||v||$.

The Ring Learning with Errors (RLWE) problem is a ring-version from the standard Learning with Errors (LWE) problem defined by Regev [31]. The RLWE problem as well as the LWE problem has become of great interest to cryptography because of its worst-case to average-case reduction, i.e., the RLWE problem that is an average-case problem reduces to a worst-case approximate SVP.

1) FOUNDATIONS

The RLWE is defined over a ring $R = \mathbb{Z}_q[x]/\Phi(x)$ of polynomials modulo a cyclotomic polynomial $\Phi(x)$ with coefficients in the field $\mathbb{Z}_q$ of integers $\mod q$ where q is a prime. The RLWE distribution $A_{s,\chi}$ over $R \times R$ is defined as

$$(a, b = s \cdot a + e \mod q),$$

where $s \in R$, $a \in R$ is sampled uniformly at random, e is sampled from χ, for χ being an error distribution over R. With the RLWE distribution, we define two RLWE problems.

1) The Search-RLWE problem is defined by: for m samples distributed according to $A_{s,\chi}$. Recover the secret s. s is called the secret.
2) The Decision-RLWE problem is defined by: given m independent samples $(a_i, b_i) \in R \times R$, where each samples is distributed according to $A_{s,\chi}$. Distinguish between $(a_i, b_i) \leftarrow A_{s,\chi}$ pairs and (a_i, b_i) pairs sampled uniformly at random.

Lyubashevsky *et al.* [32] prove that the Decision-RLWE problem is at least as hard as the search version. Contrarily, Search-RLWE problem is at least as hard as SVP$_\gamma$.

The first Diffie-Hellman key exchange protocol based on RLWE was proposed by Peikert [33]. However, in this work, we will use a protocol similar to the protocol presented by Singh and Chopra [34]. The protocol will be described below.

Suppose devices A and B want to exchange a secret for an insecure channel using the protocol proposed by Singh and Chopra [34]. Given the system's public parameters q, n, and $a(x)$. Device A generates at random, two polynomials s_A and e_A with "small" coefficients sampled from χ. It calculates

$$p_A = s_A \cdot a + e_A$$

and sends p_A to device B, while device B generates at random, two other polynomials s_B and e_B with "small" coefficients sampled from χ, and computes

$$p_B = s_B \cdot a + e_B.$$

It also generates a small e'_B from the distribution χ and computes

$$k_B = s_B \cdot p_A + e'_B = s_A \cdot s_B \cdot a + s_B \cdot e_A + e'_B.$$

Device B uses the **RandomizedRound** function on each coefficient of k_B and finds $\bar{k_B}$ = **RandomizedRound**(k_B).

The **RandomizedRound**(v) is defined for two separate cases:

- $q \equiv 1 \mod 4$. If $v = 0$, we draw a random bit and depending on the bit, we map 0 to either itself or $q - 1$. If $v = (q-1)/4$, depending on the random bit, we map $(q-1)/4$ to either itself or $(q+3)/4$.
- $q \equiv 3 \mod 4$. If $v = 0$, we draw a random bit and depending on the bit, we map 0 to either itself or $q - 1$. If $v = (3q-1)/4$, depending on the random bit, we map $(3q-1)/4$ to either itself or $(3q+3)/4$.

Then, device B calculates his key stream

$$sk_B = \lfloor \bar{k_B} \rceil_2,$$

where

$$\lfloor v \rceil_2 = \left\lfloor \frac{2}{q} \cdot v \right\rceil \mod 2,$$

and a mask of $\bar{k_B}$

$$k'_B = \langle \bar{k_B} \rangle_2,$$

where

$$\langle v \rangle_2 = \left\lfloor \frac{4}{q} \cdot v \right\rfloor \mod 2.$$

Device B then sends p_B and k'_B to device A. Finally, device A computes

$$k_A = p_B \cdot s_A.$$

Thus, device A's key stream is

$$sk_A = \mathrm{Rec}(k_A, k'_B).$$

The Rec function is defined by

$$\mathrm{Rec}(w, b) = \begin{cases} 0 & \text{if } w \in I_b + E \mod q \\ 1 & \text{otherwise,} \end{cases}$$

where $E := [-q/8, q/8) \cap \mathbb{Z}$, $I_0 := \mathbb{Z}_q \cap [0, q/4)$ and $I_1 := \mathbb{Z}_q \cap [3q/4, q)$.

Note that $k_A \approx k_B$ and the Rec function is a reconciliation function that is used to obtain a shared key from these values.

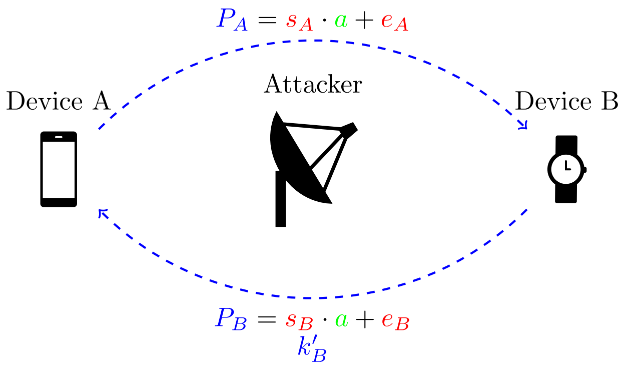

FIGURE 5. Picture illustrating the RLWEDH.

Figure 5 depicts the RLWE-Based Diffie–Hellman (RLWEDH). The variables in green are parameters, in blue are public, and in red are private.

2) COMPLEXITY

As we discussed earlier in this article, the RLWE problem is reduced to lattice SVP$_\gamma$. Thus, all attacks in SVP$_\gamma$ are based on lattice basis reduction. Lattice basis reduction is the name given to the technique that uses lattice basis and reduces its vectors to sufficiently small vectors. Obviously, lattice basis reduction is hard on lattices whereas the SVP$_\gamma$ is hard too. For a large $\gamma = 2^{\Omega(n)}$, where n is the lattice dimension, the Lenstra-Lenstra-Lovász (LLL) algorithm [35] can find a solution in polynomial time n. Until now, the Block Korkine-Zolotarev (BKZ) algorithm is the best approach to lattice reduction for small values of γ.

Since the best attacks on RLWE do not benefit from the ring structure, in this work, we consider the RLWE as a standard LWE problem. As argued in [36], we can discard the attacks like BKW, because, the RLWE-based protocol described in this work use a limited number of LWE samples ($m \approx n$). Because of the limited number of LWE samples, we consider only two BKZ-based attacks. These two attacks are called primal and dual attacks, see [36] for a description of these attacks.

The BKZ requires an algorithm that solves de SVP in smaller dimensions $b < n$, where n is the lattice dimension. This algorithm is called "SVP Oracle". Until now, the best SVP Oracle is a lattice sieve. Therefore, as in [36], we consider BKZ with lattice sieve algorithms as SVP Oracle.

According to previous researches [37], [38], the classical-world complexity (C-RLWE) is given by

$$8d \cdot 2^{0.292b+16.4}, \tag{11}$$

where $d = m + n + 1$ is the lattice dimension, b is the block size used in BKZ and m, n are the parameters of the RLWE-based protocol.

TABLE 1. Comparison between the algorithms.

	Elements	Secrets	Computations	Hard Problem
DH	Ints. g	exp. x	$g, x \mapsto g^x$	Given g, g^x, find x
ECDH	Points P in E	scalars k	$k, P \mapsto [k]P$	Given P, $[k]P$, find k
SIDH	Curves E in isogeny classes	isog. ϕ	$\phi, E \mapsto \phi(E)$	Given E, $\phi(E)$, find ϕ
RLWEDH	Polynomials $a(x) \in R$	small errors s, $e \in R$	a, s, $e \mapsto a \cdot s + e$	Given a and $a \cdot s + e$, find s
CODH	Vectors $(x, y) \in \mathcal{S}_w^n(\mathbb{F}_2) \times \mathcal{S}_w^n(\mathbb{F}_2)$	small errors vectors e, r_1, r_2	$x, y, e, r_1, r_2 \mapsto$ $e_c = x \cdot r_2 - y \cdot r_2 + e$	Given (x, y) and e_c, find (r_1, r_2)

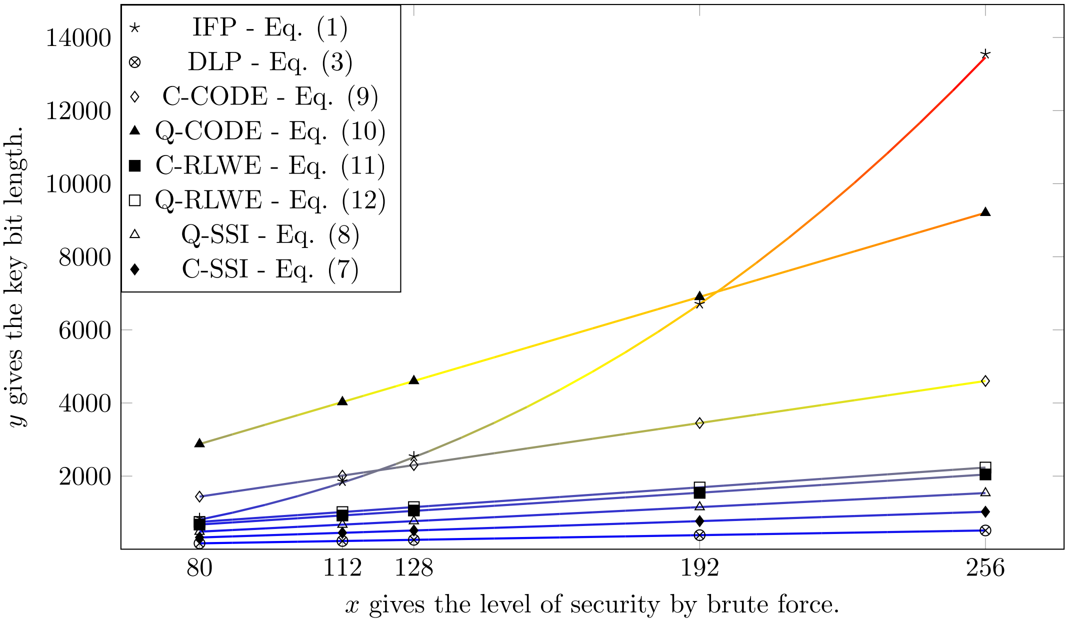

FIGURE 6. Comparison between brute force and minimum key length.

Similarly, previous researches [38], [39] show that the best-known algorithm to solve RLWE in the quantum-world (Q-RLWE) has complexity given by

$$8d \cdot 2^{0.265b+16.4}. \tag{12}$$

Observe, the classical and the quantum complexity depends on the lattice dimension d and the size of block b used in BKZ. The value of b is chosen to optimize the attack. We refer to [36] and [38] for a better understanding of how the choice of b value is made.

IV. COMPARISON BETWEEN PRIMITIVES

Table 1 shows a summary of the main points in the key-agreement algorithms presented above for classical and post-quantum cryptography in the NIST Post-Quantum Standardization Process.

A. SECURITY COMPARISON

To compare the key length for the algorithms mentioned above, we are matching the complexity of every algorithm with the complexity for a brute force attack in a key of x bits. Since we know parameters for brute force, we can find the key length. This kind of comparison is inspired by [40] and [41], which are, to the best of our knowledge, the firsts to do this kind of analysis.

Table 2 contains the values found with the complexity equations. The curves in Table 3 were interpolated from the data in Table 2. We use Sage's `find_fit`[1] function in the Numerical Optimization module to find the curves that fit the values in Table 2. With the values in Table 2 and the curves in Table 3, we generate Figure 6, which describes a trade-off between security and key length in bits.

TABLE 2. Comparison between brute force and minimum key length.

Brute Force	80	112	128	192	256
DLP	160	224	256	384	512
IFP	851	1 853	2 538	6 707	13 547
NIST	1 024	2 048	3 072	7 680	15 360
C-SSI	320	448	512	768	1 024
Q-SSI	480	672	768	1 152	1 536
C-CODE	1 438	2 013	2 301	3 451	4 602
Q-CODE	2 876	4 026	4 602	6 902	9 203
C-RLWE	673	921	1 058	1 541	2 045
Q-RLWE	746	1 016	1 152	1 688	2 234

TABLE 3. Adjusted curves from the data in Table 2.

Algorithms	Adjusted curves
DLP	$2x - 2.26$
IFP	$0.02x^{2.42}$
C-SSI	$4x$
Q-SSI	$6x$
C-CODE	$17.98x - 0.24$
Q-CODE	$35.95x + 0.11$
C-RLWE	$7.77x + 53.42$
Q-RLWE	$8.45x + 69.91$

[1] https://www.sagemath.org

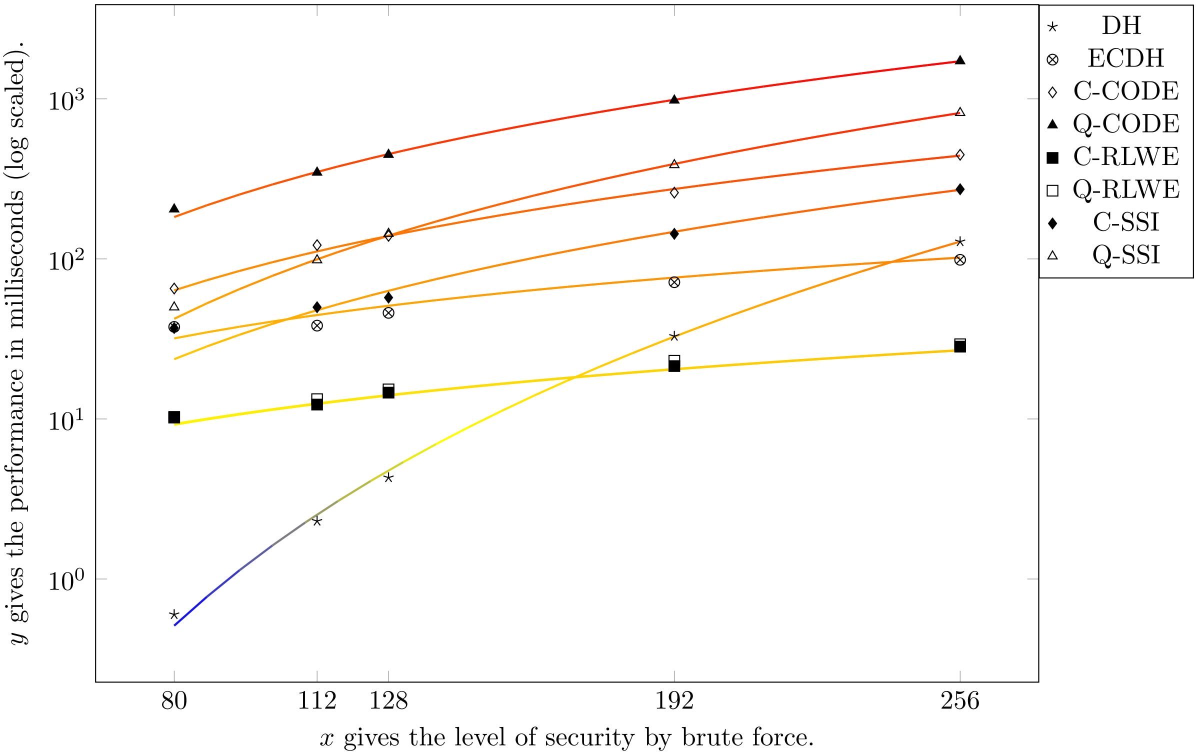

FIGURE 7. Performance Comparison.

In Figure 6, we see that DLP outperforms all other algorithms in terms of key size. However, DLP is easily solved by a quantum computer. The second best overall and the best post-quantum algorithm is the SIDH, followed by the RLWEDH. We can also see that while CODH is worse than IFP at initial security levels, however, at the slightly over 120-bit security level, classical CODH becomes better than IFP and at the nearly 196-bit security level, quantum CODH becomes better than IFP. This is because the key-size growth for IFP grows exponentially with the security level, while the CODH grows linearly. Also, another point in favor of CODH when compared to IFP is that the former is resistant to quantum computers while the latter is not. Unfortunately, it was not possible to plot the curves for quantum attacks on IFP and DLP because of the large key size required to ensure security levels. In fact, consider the complexity for the quantum DLP attack given by the equation, for the lowest security level (80-bit), we would need a key of more than a hundred million bits.

B. PERFORMANCE COMPARISON

In this section, we compare the performance of the protocols described in this work. For that, all protocols were implemented in SAGE language [42] for a 2.50GHz Intel i7 − 6500U quad-core, 16*GB* RAM and Ubuntu 20.04 64-bit operating system. Since the command `EllipticCurveIsogeny` is not efficient, we used the code from Feo.[2]

TABLE 4. Performance of key agreement protocols in milliseconds for each level of security.

Security Level	80	112	128	192	256
DH	0.6	2.3	4.3	32.9	128.2
ECDH	37.6	38.3	46.0	71.5	98.7
C-SSI	37.0	49.9	57.3	143.4	272.4
Q-SSI	49.9	98.3	143.8	386.4	818.0
C-CODE	65.3	121.9	139.4	259.4	447.5
Q-CODE	204.3	346.6	447.8	978.3	1722.1
C-RLWE	10.2	12.3	14.6	21.4	28.3
Q-RLWE	10.3	13.3	15.3	23.1	29.3

TABLE 5. Adjusted curves from the data in Table 4.

Algorithms	Adjusted curves
DH	$4.6 \cdot 10^{-13} \cdot x^{4.8}$
ECDH	$4.0 \cdot 10^{-4} \cdot x^{0.9}$
C-SSI	$2.4 \cdot 10^{-6} \cdot x^{2.1}$
Q-SSI	$6.0 \cdot 10^{-7} \cdot x^{2.5}$
C-CODE	$4.3 \cdot 10^{-5} \cdot x^{1.7}$
Q-CODE	$3.9 \cdot 10^{-5} \cdot x^{1.9}$
C-RLWE	$1.0 \cdot 10^{-4} \cdot x + 1.2 \cdot 10^{-3}$
Q-RLWE	$1.0 \cdot 10^{-4} \cdot x + 1.3 \cdot 10^{-3}$

Using the parameter sizes found in Table 2, we ran a total of 1 000 simulations for each protocol and each security level. It is worth noting that, for ECDH, we use curves considered safe presented in [43]. As in [43] there is no curve for the 80-bit security level, in this work, we use the Curve P-192 recommended by NIST for the 80-bit security level. In our simulations, we do not consider the generation time of the protocol parameters. For the time counting, we only consider the calculations of the partial keys and the secret key. While the ECDH uses only values from the DLP, the classic Diffie-Hellman protocol uses values from both, IFP and DLP.

[2] https://github.com/defeo/ss-isogeny-software

Specifically, for the size of the prime p, we consider the values found using IFP; but for the secret integers r and s, we consider the values found using DLP. This setting is the same used in the HTTPS and results high performance for low security levels with rapid loss of performance when the security level increases.

Note that Table 2 provides more than parameters for performance comparison in terms of processing time. Table 2 and Figure 6 provide a comparison of data size for transferring in a network, storing in long-term memory, and allocating in short-term memory.

Table 4 shows the computation times for each of the protocols and security levels. Again, we use Sage's `find_fit` function to find the curves that fit the values in Table 4. Table 5 shows the curves found.

Figure 7 presents the growth of the computational cost of the protocols for each security level. Its horizontal axis is given in bits, while its vertical axis is given in logarithmic scale of milliseconds.

In Figure 7, we see that RLWE outperforms other post-quantum protocols. Only for the first security levels, the DH is faster than RLWE due to the DLP and IFP. In fact, the protocol based on RLWE outperforms even the classical protocol based on elliptic curves. Although ECDH uses small keys due to DLP, the point operations require several computations [10], [11], thus, ECDH performance is better than DH only for the last level of security. The protocol based on codes is the one that presents the worst performance. Although the CODH works in binary fields, the algorithm to solve the CED problem requires a considerable amount of time, thus making the protocol present the worst performance. However, we believe that the implementation of both protocols based on isogeny and code could be more optimized.

V. CONCLUSION

In this paper, we have shown a security comparison for the most promising key agreement protocols in the post-quantum era of cryptography. To do this analysis, we have compared this post-quantum algorithms with the present day most used classical algorithms. The complexity for each of the best-known attacks, in a classical and quantum way, to IFP, DLP, SSI, ISD and RLWE, was evaluated and matched to the complexity of a brute force attack. We were able to get the minimum key-length to achieve the practical security levels proposed by NIST.

The analysis performed in this work allowed us to compare the algorithms completely. The results show that SSI uses small key sizes when compared to other post-quantum algorithms. In sequence, we have the RLWE and lastly code-based algorithm. The results also show that when compared to IFP, the code-based algorithm shows worse results for initial security levels but becomes better than IFP from a certain level of security. Considering the performance of the protocol, the RLWE based presents the best result among the post-quantum algorithms followed by SSI and code-based algorithm.

SSI-based algorithm currently outperforms both RLWE and code-based algorithm in terms of key size, while RLWE protocol outperforms both SSI-based and code-based on performance. Further research is needed to be held to compare other classes of post-quantum algorithms, such as digital signature schemes. Comparing specific post-quantum algorithms, such as those competing at NIST Post-Quantum Standardization Process is also a future research direction.

REFERENCES

[1] F. Arute, "Quantum supremacy using a programmable superconducting processor," *Nature*, vol. 574, no. 7779, pp. 505–510, 2019.

[2] P. W. Shor, "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer," *SIAM J. Comput.*, vol. 26, no. 5, pp. 1484–1509, Oct. 1997.

[3] L. K. Grover, "A fast quantum mechanical algorithm for database search," in *Proc. twenty-eighth Annu. ACM Symp. Theory Comput.*, New York, NY, USA,1996, pp. 212–219, doi: 10.1145/237814.237866.

[4] D. Bernstein, J. Buchmann, and E. Dahmen, *Post-Quantum Cryptography*. Berlin, Germany: Springer, 2009, doi: 10.1007/978-3-540-88702-7.

[5] F. B. de Oliveira, *Selected Privacy-Preserving Protocols*. Cham, Switzerland: Springer, 2017, pp. 61–100.

[6] W. Diffie and M. Hellman, "New directions in cryptography," *IEEE Trans. Inf. Theory*, vol. IT-22, no. 6, pp. 644–654, Nov. 1976.

[7] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Commun. ACM*, vol. 21, no. 2, pp. 120–126, Feb. 1978.

[8] F. Borges, P. Lara, and R. Portugal, "Parallel algorithms for modular multi-exponentiation," *Appl. Math. Comput.*, vol. 292, pp. 406–416, Jan. 2017.

[9] D. Pereira, M. Aranha, and F. Borges, "HTTPS keys in the mediterranean," in *Proc. II Workshop Metrol.*, Jun. 2019, pp. 449–454.

[10] V. S. Miller, "Use of elliptic curves in cryptography," in *Advances in Cryptology*. New York, NY, USA: Springer-Verlag, 1986, pp. 417–426.

[11] N. Koblitz, "Elliptic curve cryptosystems," *Math. Comput.*, vol. 48, no. 177, pp. 203–209, 1987.

[12] S. Y. Yan, *Quantum Attacks Public-Key Cryptosystems*. Boston, MA, USA: Springer, 2013.

[13] J. Proos and C. Zalka, "Shor's discrete logarithm quantum algorithm for elliptic curves," *arXiv:quant-ph/0301141*, 2003, [Online]. Available: https://arxiv.org/abs/quant-ph/0301141.

[14] A. Rostovtsev and A. Stolbunov, "Public-key cryptosystem based on isogenies," Cryptol. ePrint Arch., Las Vegas, NV, USA, Tech. Rep. 2006/145, 2006. [Online]. Available: https://eprint.iacr.org/2006/145

[15] D. Jao and L. D. Feo, "Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies," in *Post-Quantum Cryptography*, B.-Y. Yang, Ed. Berlin, Germany: Springer-Verlag, 2011, pp. 19–34.

[16] C. Téllez and F. Borges, "Trade-off between performance and security for supersingular isogeny-based cryptosystems," in *Proc. Anais do Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais (SBSeg)*, 2018, pp. 113–126. [Online]. Available: http://portaldeconteudo.sbc.org.br/index.php/sbseg/article/view/4247

[17] S. Galbraith and A. Stolbunov, "Improved algorithm for the isogeny problem for ordinary elliptic curves," *Appl. Algebra Eng., Commun. Comput.*, vol. 24, no. 2, pp. 107–131, Jun. 2013.

[18] L. D. Feo, D. Jao, and J. Plût, "Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies," Cryptol. ePrint Arch., Las Vegas, NV, USA, Tech. Rep. 2011/506, 2011. [Online]. Available: https://eprint.iacr.org/2011/506

[19] S. Tani, "Claw finding algorithms using quantum walk," *Theor. Comput. Sci.*, vol. 410, no. 50, pp. 5285–5297, Nov. 2009.

[20] G. Adj, D. Cervantes-Vázquez, J.-J. Chi-Domínguez, A. Menezes, and F. Rodríguez-Henríquez, "On the cost of computing isogenies between supersingular elliptic curves," in *Proc. IACR*, 2018, p. 313.

[21] R. J. McEliece, "A public-key cryptosystem based on algebraic," *Coding Thv*, vol. 4244, pp. 114–116, Apr. 1978.

[22] E. Berlekamp, R. McEliece, and H. van Tilborg, "On the inherent intractability of certain coding problems (Corresp.)," *IEEE Trans. Inf. Theory*, vol. IT-24, no. 3, pp. 384–386, May 1978, doi: 10.1109/TIT.1978.1055873.

[23] R. Misoczki, J. Tillich, N. Sendrier, and P. S. L. M. Barreto, "Mdpc-mceliece: New mceliece variants from moderate density parity-check codes," in *Proc. IEEE Int. Symp. Inf. Theory*, Apr. 013, pp. 2069–2073.

[24] P. Gaborit, G. Murat, O. Ruatta, and G. Zémor, "Low rank parity check codes and their application to cryptography," in *Proc. Workshop Coding Cryptography WCC*, 2013, pp. 1–7.

[25] J.-C. Deneuville, P. Gaborit, and G. Zémor, "Ouroboros: A simple, secure and efficient key exchange protocol based on coding theory," in *Proc. PQCrypto*, 2017, pp. 1–10.

[26] J. Hoffstein, J. Pipher, and J. H. Silverman, "Ntru: A ring-based public key cryptosystem," in *Proc. Int. Algorithmic Number Theory Symp.*, 1998, pp. 267–288.

[27] R. Canto Torres and N. Sendrier, "Analysis of information set decoding for a sub-linear error weight," in *Post-Quantum Cryptography*, T. Takagi, Ed. Cham, Switzerland: Springer, 2016, pp. 144–161.

[28] C. Tãllez, D. Pereira, and F. Borges, "Trade-off between performance and security for coding and ring learning with errors-based diffie-hellman cryptosystems," in *Proc. Workshop Metrol.*, Jun. 2019, pp. 460–465.

[29] D. J. Bernstein, T. Lange, and C. Peters, "Smaller decoding exponents: Ball-collision decoding," in *Advances in Cryptology*, P. Rogaway, Ed. Berlin, Germany: Springer, 2011, pp. 743–760.

[30] A. Becker, A. Joux, A. May, and A. Meurer, "Decoding random binary linear codes in 2n/20: How 10 improves information set decoding," in *Advances in Cryptology*, D. Pointcheval and T. Johansson, Eds. Berlin, Germany: Springer, 2012, pp. 520–536.

[31] O. Regev, "On lattices, learning with errors, random linear codes, and cryptography," *J. ACM*, vol. 56, no. 6, pp. 34-1–34-40, 2009.

[32] V. Lyubashevsky, C. Peikert, and O. Regev, "On ideal lattices and learning with errors over rings," in *Proc. Annu. Int. Conf. Theory Appl. Cryptograph. Techn.*, 2010, pp. 1–23.

[33] C. Peikert, "Lattice cryptography for the Internet," in *Post-Quantum Cryptography*, M. Mosca, Ed. Cham, Switzerland: Springer, 2014, pp. 197–219.

[34] V. Singh and A. Chopra, "Even more practical key exchanges for the Internet using lattice cryptography," Cryptol. ePrint Arch., Las Vegas, NV, USA, Tech. Rep. 2015/1120, 2015. [Online]. Available: https://eprint.iacr.org/2015/1120

[35] A. K. Lenstra, H. W. Lenstra, and L. Lovász, "Factoring polynomials with rational coefficients," *Math. Ann.*, vol. 261, no. 4, pp. 515–534, Dec. 1982.

[36] E. Alkim, L. Ducas, T. Pöppelmann, and P. Schwabe, "Post-quantum key exchange—A new hope," in *Proc. 25th Secur. Symp.*, 2016, pp. 327–343.

[37] A. Becker, L. Ducas, N. Gama, and T. Laarhoven, "New directions in nearest neighbor searching with applications to lattice sieving," in *Proc. 27th Annu. ACM-SIAM Symp. Discrete Algorithms*, Jan. 2016, pp. 10–24.

[38] M. Albrecht, M. Chase, H. Chen, J. Ding, S. Goldwasser, S. Gorbunov, S. Halevi, J. Hoffstein, K. Laine, K. Lauter, S. Lokam, D. Micciancio, D. Moody, T. Morrison, A. Sahai, and V. Vaikuntanathan, "Homomorphic encryption security standard," HomomorphicEncryption.org, Toronto, ON, Canada, Tech. Rep., Nov. 2018.

[39] T. Laarhoven, "Search problems in cryptography," Ph.D. dissertation, Dept. Math. Comput. Sci., Eindhoven Univ. Technol., Eindhoven, The Netherlands, 2015. [Online]. Available: http://www.thijs.com

[40] F. Borges and M. Muhlhauser, "EPPP4SMS: Efficient privacy-preserving protocol for smart metering systems and its simulation using real-world data," *IEEE Trans. Smart Grid*, vol. 5, no. 6, pp. 2701–2708, Nov. 2014.

[41] F. B. de Oliveira, *On Privacy-Preserving Protocols for Smart Metering Systems: Secur. Privacy Smart Grids*. Cham, Switzerland: Springer, 2017, doi: 10.1007/978-3-319-40718-0.

[42] The Sage Developers. (2019). *SageMath, the Sage Mathematics Software System (Version 8.7)*. [Online]. Available: https://www.sagemath.org

[43] D. F. Aranha, P. S. Barreto, C. P. Geovandro, and J. E. Ricardini, "A note on high-security general-purpose elliptic curves," in *Proc. IACR*, 2013, p. 647.

FÁBIO BORGES (Senior Member, IEEE) received the bachelor's degree in mathematics from Londrina State University (UEL), the master's degree in computational modeling from the National Laboratory for Scientific Computing (LNCC), and the Ph.D. degree in Doctor of Engineering (Dr.-Ing.) from the Department of Computer Science, TU Darmstadt. He is currently a Professor with LNCC where he gives lectures for Ph.D. students in computational modeling. He works in the areas of security and privacy, artificial intelligence, smart grids, high performance computing, and algorithms. He is author of the book *On Privacy-Preserving Protocols for Smart Metering Systems: Security and Privacy in Smart Grids*. He is the coauthor, with C. R. Rao and B. B. Pereira, of the book *Statistical Learning Using Neural Networks: A Guide for Statisticians and Data Scientists with Python*. He received the Latin America Distinguished Service Award by the IEEE Communications Society.

PAULO RICARDO REIS graduated in physics from the Federal Center of Technological Education Celso Suckow da Fonseca (CEFET/RJ). He is currently pursuing the LNCC Postgraduate Program. He is also researching on security and privacy under the supervision of Prof. Fábio Borges with the National Laboratory for Scientific Computing (LNCC). He has a technical background in information technology. He was researching with the Quantum Computation and Cryptography Group.

DIOGO PEREIRA received the B.S. degree in computational mathematics from the Federal University of Paraíba (UFPB), and the M.Sc. degree in computational modeling from the National Laboratory for Scientific Computing (LNCC). He is currently pursuing the Ph.D. degree with the LNCC Postgraduate Program. He is also researching on security and privacy under the supervision of Prof. Fábio Borges with LNCC.