Hybrid approach to public-key algorithms in the near-quantum era

Adrian Cinal[1][0009−0001−9789−0470], Gabriel Wechta[1][0009−0009−8560−5300], and Michał Wroński[1][0000−0002−8679−9399]

NASK National Research Institute, Kolska Str. 12, Warsaw, Poland
{adrian.cinal, gabriel.wechta, michal.wronski}@nask.pl

Abstract. Application of post-quantum algorithms in newly deployed cryptosystems is necessary nowadays. In the NIST Post-Quantum Competition several algorithms that seem to be resistant against attacks mounted using quantum computers have been chosen as finalists. However, it is worth noting that one of finalists — SIKE — was catastrophically broken by a classical attack of Castryck and Decru only a month after qualifying for the final round. This shows that absolute trust cannot yet be placed in the algorithms being standardized. And so a proposition was made to use the novel, post-quantum schemes alongside the well-studied classical ones with parameters chosen appropriately to remain secure against quantum attacks at least temporarily, i.e., until a large enough quantum computer is built.
This paper analyzes which classical public-key algorithms should be used in tandem with the post-quantum instances, and studies how to ensure appropriate levels of both classical and quantum security. Projections about the development of quantum computers are reviewed in the context of selecting the parameters of the classical schemes such as to provide quantum resistance for a specified amount of time.

Keywords: post-quantum algorithms · classical algorithms · quantum computing · security level

1 Introduction

Post-quantum cryptography is a relatively new branch of modern cryptology. In 2017 NIST announced their post-quantum cryptography (NIST PQC) competition to which researchers from all around the world could submit their proposals of public-key schemes: key establishment and signature algorithms.

NIST in [7] defined five security levels for assessing the security of post-quantum cryptosystems. Instead of relying on precise estimates of the number of bits of security, these levels take into account both classical and quantum cryptanalysis. Each level is characterized by a reference primitive (AES or SHA), with *its* security forming the basis for subsequent analyses.

Table 1 presents the NIST security levels, providing their definitions and the estimated resources (quantum and classical gates) required to compromise the

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63778-0_27

reference primitive at each level. As indicated in [7], plausible values for MAXDEPTH range from 2^{40} logical gates (per year) to 2^{64} logical gates (per decade), up to a maximum of 2^{96} logical gates (per millennium).

Table 1: NIST security levels as defined in [7] with estimated complexity of breaking the scheme's security.

Security Level	Security Definition *Must require computational resources comparable to or greater than*	Quantum Gates (estimated)	Classical Gates (estimated)
1	exhaustive key search on a 128-bit key block cipher (e.g. AES-128)	2^{170}/MAXDEPTH	2^{143}
2	collision search on a 256-bit hash function (e.g. SHA3-256)	-	2^{146}
3	exhaustive key search on a 192-bit key block cipher (e.g.AES-192)	2^{233}/MAXDEPTH	2^{207}
4	collision search on a 384-bit hash function (e.g. SHA3-384)	-	2^{210}
5	exhaustive key search on a 256-bit key block cipher (e.g. AES-256)	2^{298}/MAXDEPTH	2^{272}

Once submitted, each algorithm has been analyzed both by NIST specialists and the cryptographic community. These analyses resulted in many weak algorithms being eliminated at an early stage of the competition. Identification and subsequent withdrawal of broken schemes continued, however, all the way to the end. During the third and fourth rounds, respectively, the Rainbow digital signature algorithm and the SIKE key establishment algorithm were compromised in the classical setting. Rainbow was broken completely on security level 1 and significantly weakened on other security levels, whereas SIKE is now known to be breakable in mere 2 hours on a classical CPU even at the highest (fifth) security level.

Since post-quantum algorithms have not yet been analyzed thoroughly enough, it is vital from the point of view of security to combine them with classical schemes. Thus each public-key scheme should be a hybrid consisting of at least two parts: a classical instance (secure in the classical setting) and a post-quantum instance (conjectured secure in both classical and post-quantum settings). Should the post-quantum instance prove to be breakable classically, this approach keeps the overall scheme secure against classical attacks and against quantum attacks for some time also (provided the parameters of the classical instance are chosen adequately) — until a powerful enough quantum computer is built. In the long run, if the post-quantum part of the hybrid scheme stands the test of time, the scheme shall remain secure against quantum adversaries despite the classical part having been long obsolesced.

The necessity of using hybrid public-key algorithms has been postulated years ago, even before NIST PQC competition started, but the spectacular failures of the round-three and round-four candidates (finalists) show that the need to combine (largely experimental) post-quantum cryptosystems with conservative, well-studied classical ones is very much real. Such hybrid solutions exist today and are used, for example, in TLS 1.3 [32], where classical algorithms based on elliptic curves over 256-bit prime fields, X25519 and Secp256r1, are used alongside Kyber768, so far believed to be resistant against quantum attacks. TLS 1.3 uses a simple "concatenation approach," where public keys of the two algorithms are concatenated back to back and transmitted as a single value in order to avoid changing the existing data structure and message fields. Similarly, when deriving the session key, two shared secrets are obtained by the two schemes, classical and post-quantum, and are then concatenated to obtain the master shared secret, from which the session key is derived.

This combination, however, is largely ad hoc. What we endeavour to achieve in this paper is a careful analysis of the classical instances (signature and key establishment) and their respective parameters that would match the most closely the security levels of their post-quantum counterparts and accompany them best. We shall also provide estimates about how long these classical companions shall remain secure, based on the current forecasts about the development of quantum computers.

2 Known attacks against post-quantum instances

2.1 Attack on Rainbow

Beullens in 2022 presented new key recovery attacks against Rainbow [4], one of the three finalist signature schemes in the NIST PQC competition. Previously, it was believed that breaking Rainbow at its lowest security level would take 2^{128} operations. Beullens' attack, however, utilizes differentials to efficiently recover the secret key, thus surpassing all previously known attacks for every parameter set submitted to NIST. Specifically, with a Rainbow public key for the NIST security level 1 parameters from the second-round submission, Beullens' approach can retrieve the corresponding secret key in an average of 53 hours (roughly a weekend) using a standard laptop.

2.2 Attack on SIKE

In June of 2022, SIKE advanced to the fourth round of NIST PQC competition as an alternate candidate algorithm for key establishment. Not a whole month afterwards, the algorithm was totally broken by Castryck and Decru [6] on a classical computer. A limitation to their attack is that the endomorphism ring of the base curve must be known (which, however, is already the case in SIKE). Still, not long after this attack, other attacks were devised, with Maino and Martindale [20] presenting a subexponential attack on SIKE, which does not

DOI: 10.1007/978-3-031-63778-0_27

require the knowledge of the endomorphism ring of the base curve. Robert [26] then introduced an algorithm for breaking SIKE, which is, first, polynomial-time, and, second, works no matter the choice of the starting curve.

The attack of Castryck and Decru takes only a few hours on a regular laptop to break SIKE instances at NIST security level 5. It is worth noting that by the time of the attack isogeny-based cryptography had already been studied for about 10 years and no similar attacks had been found. Also, Kani's theorem, lying at the heart of the attack, had been known for over 20 years and was never regarded to be any serious threat. This goes to show that even schemes built on top of well-studied constructions may surprisingly fail, and so extra precautions must be taken as we are entering the near-quantum era.

2.3 Recent breakthrough against LWE

In a recent article [8], Chen proposes an efficient quantum algorithm for solving the *learning with errors* (LWE) problem. Hardness of LWE is the assumption underlying the security of lattice-based schemes such as CRYSTALS-Kyber and CRYSTALS-Dilithium standardized already by NIST. As of this writing, Chen's paper is undergoing peer review and the validity of his claims or their practical impact are yet unclear.

3 Quantum computing

The purpose of this Section is to provide a brief overview of the principles of quantum computation and, above all, introduce the terminology used throughout the paper.

Exploiting quantum mechanics to obtain computational advantage was first proposed by Feynman in 1981 [14], followed by Deutsch definining a quantum Turing machine in 1985 [10]. Thus, the field of quantum computing was born and with publication in 1994 of Shor's seminal paper [31], it gained unprecedented momentum and attracted the attention of cryptographers, as in [31] Shor showed how to leverage quantum computation to break the mathematical problems underpinning contemporary asymmetric cryptography. Two years later, Grover presented an algorithm for searching an unstructured N-element set in time $O(\sqrt{N})$ [17], thus posing further threat to symmetric cryptography.

Similarly to bits in classical computing, a fundamental unit of information in quantum computing is a *qubit* which we associate with a unit vector in $\mathbb{C}^2$:

$$|q\rangle = \alpha_0 |0\rangle + \alpha_1 |1\rangle \tag{1}$$

with $|\alpha_0|^2+|\alpha_1|^2 = 1$ and $|0\rangle = (1,0)^T$, $|1\rangle = (0,1)^T$ column (standard basis) vectors in $\mathbb{C}^2$. We say that a qubit is in a *superposition* of the states $|0\rangle$ and $|1\rangle$. A qubit can be *measured* thus yielding a classical binary value 0 or 1, where 0 is measured with probability $|\alpha_0|^2$ and 1 is measured with probability $|\alpha_1|^2$ according to *Born's rule* (we refer to the numbers $\alpha_0, \alpha_1 \in \mathbb{C}$ as *amplitudes* of their

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63778-0_27

associated basis states $|0\rangle$ and $|1\rangle$, respectively). After a measurement, a qubit is said to have *collapsed* to a classical state and the superposition once present is now destroyed. Before measurement, however, qubits can be manipulated in such a way that once measured they collapse with overwhelming probability to the desired result of computation. This is the basis of quantum computing.

While there are other realizations of quantum computation (cf., for example, *quantum annealing*) the most prevalent model is that of quantum *gates*. A gate represents a *unitary* (reversible and preserving the unit length of the vector $|q\rangle$) transformation to a qubit or a register of n qubits. Conceptually, we design quantum algorithms in this setting as *circuits* with wires going in and out of gates, whereas a physical realization may be completely decoupled from this image (and quantum "circuits" are more temporal than spacial in practice with gates being applied in place one after another).

We say a set of quantum gates is *universal* if any operation possible on a quantum computer (any unitary), or at least a satisfactory approximation thereof, can be expressed as finite sequence of gates from this set. The most common such universal set currently studied is the Clifford+T set and so it shall also be the focus of this paper.

Note that quantum systems are susceptible to noise which has to be accounted for in the quantum computer by implementing extensive error correction. By far the most costly of the Clifford+T gates to implement in a fault-tolerant manner is the T gate [15, 27] corresponding to the following unitary matrix:

$$T = \begin{pmatrix} 0 & 0 \\ 0 & e^{i\pi/4} \end{pmatrix}. \tag{2}$$

For this reason complexity of quantum circuits is often expressed in terms of the number of T gates (or T*-count*) or the number of T gates modulo gates which can be "run" in parallel (T*-depth*) [18]. Another useful metric which we shall refer to in this paper is the number of *logical* qubits[1] needed to run the circuit. This is also sometimes referred to as the *width* of the circuit.

Another commonly found measure of complexity is the number of *Toffoli gates* or the associated *Toffoli-depth* [19, 27]. Toffoli gates can be implemented using 7 T gates and a T-depth varying between 4 and 1 [1, 28], thus we shall translate the Toffoli-count (Toffoli-depth) estimates found in the literature to the T-counts (T-depths) and use these as a common denominator.

4 Analyses

4.1 Methodology

It must be noted that due to noise in the quantum computations, there are success probabilities typically associated with the attacks presented in the literature.

[1] As a fault-tolerance and error-correction measure a single logical qubit is typically implemented using a number of physical qubits.

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63778-0_27

For the purposes of this paper, however, we shall work on the assumption that if the quantum resources available are sufficient to mount an attack with non-negligible probability, even if it requires rerunning the computations a number of times, then the relevant cryptosystem is vulnerable.

We shall restrict our attention to RSA as the sole scheme based on integer factorization intractability,[2] and to elliptic-curve schemes (ECDSA, EdDSA, ECDH) based on the discrete logarithm problem. As for the latter, we shall further focus only on elliptic curves over prime fields $\mathbb{F}_p$. The decisions here are motivated by the relevant NIST publications, which deprecate use of DSA [21] as well as ECDSA based on binary curves [21,22]. Also, we are explicitly interested in signature and key establishment algorithms, for which standard implementations always fall into one of the two categories just delineated. While elliptic curves over extension fields (including binary fields) are used in some settings (e.g., in pairing-based cryptography), we intentionally leave them out and focus on schemes of the most fundamental utility and enjoying most standardization and prevalence. Quantum cryptanalysis of binary curves receives thorough treatment in [2].

Our methodology focuses on analyzing progress in the development of quantum computers and the scale of computations able to be run on them. In particular, it is important to determine when we expect to build a quantum computer on which one could run Shor's algorithm for a given problem (integer factorization, discrete logarithm in a finite field, discrete logarithm on an elliptic curve) with given parameters. A crucial factor for this approach is the choice of the time frame and deciding for how long the information (protected by the hybrid scheme) should remain secure (authentic, secret).

Three time periods have to be taken into consideration:

- *implementation time* — time required for the scheme to be globally deployed,
- *usage time* — time when the scheme is actively used,
- *expiration time* — period in which the scheme is being phased out (deprecated), but the information protected by it should still remain secure.

For example, suppose we want to protect a piece of information (encrypted using a key derived from a hybrid key establishment protocol) for 5 years. If we design the system now, implement it by 2025, and intend to use it until 2040, we need to ensure the security of the information until the end of the year 2045 (the last time any plaintext is encrypted using the hybrid scheme may be in late 2040).

Assuming a declassification period of 25 years[3] and intended usage time until year 2040, we arrive at the conclusion that hybrid schemes should be resistant against quantum attacks until 2065 (on all NIST security levels). As the analyses presented in Section 5.1 show, this can be achieved with practical values for security parameters.

[2] The Paillier cryptosystem, found in multi-party computation, uses "de facto" RSA keys so interested parties may use our analyses to evaluate viability of Paillier's encryption for their purposes. Caution is advised here, however, as this scheme is outside the scope of our work.

[3] See, e.g., U.S. Executive Order 13526.

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63778-0_27

Remark 1. When analyzing the complexity of the attack algorithms (Shor's [31], Grover's [17]), it is customary to consider quantum resources usage, not merely time, as is the case for classical algorithms. This is due to the nature of the current state of quantum computing: since the problem of engineering large-scale multi-qubit systems has not yet been solved, these finer points about the algorithms/circuits give better insight into exactly how feasible the attacks are.

4.2 Classical schemes based on factoring

Integers of the form $N = pq$, for p, q — different primes of similar length in bits, are commonly used in the RSA cryptosystem, and thus called *RSA integers* going forward. The security of RSA is based on the assumption that factoring such integers is computationally infeasible.[4] Shor's algorithm [31] efficiently, albeit quantumly, factors RSA integers, thereby solving the underlying computational problem of the RSA cryptosystem. The best known algorithm for factoring RSA integers on classical computers is the General Number Field Sieve [5] which heuristically runs in subexponential time, while the time (or more practically - circuit depth) of Shor's algorithm is polynomial in the size of the input. This is achieved by a reduction to finding the order of an element in $\mathbb{Z}_N$ [31].

Since the introduction of Shor's algorithm, there have been numerous attempts to optimize it in terms of both the number of required logical qubits and the number of quantum gates.

In each run, Shor's factoring algorithm requires $2n$ group operations for an n bit integer. Ekerå and Håstad [13] have shown that, by replacing order finding with short discrete logarithm computation, the number of group operations can be reduced to $\frac{3}{2}n$ without making any trade-offs.

Gidney et al. in [16] have presented a quantum algorithm for factoring RSA integers which, by introducing a number of optimization techniques, has significantly reduced quantum resource costs when compared to the original Shor's algorithm and follow-up works. Additionally, they provide a detailed analysis of the quantum resource requirements of the algorithm in terms of logical and physical qubits as well as Toffoli and T gates. Gidney et al. report over 100x improvement over other top works, which use the same basic cost model as they do. Reported results are presented in Tables 2 and 3.[5]

In 2023, Regev presented an algorithm with lattice reduction post-processing that lowered the number of gates from $\tilde{O}(n^2)$ (original Shor's algorithm) to $\tilde{O}(n^{\frac{3}{2}})$ [25] at the expense of increasing the number of logical qubits from $O(n)$ (optimized Shor's algorithm) to $O(n^{\frac{3}{2}})$.[6] Soon after that, Ragavan and Vaikun-

[4] Technically, it is based on solving for roots modulo N, but this distinction is not relevant here.

[5] The metrics outlined in [16] differ from conventional standards, particularly in their nomenclature. To establish a unified basis for comparing various algorithms in both RSA and EC cryptography, we adopt a consistent set of metric names. For a detailed explanation of these metrics, we invite readers to refer to Appendix A in [16].

[6] Although, as we explain in subsection 5.1, from a practical point of view, it remains unclear whether such trade-offs lead to faster realization of the attack in practice.

tanathan [24] showed how to lower the number of necessary qubits to only $\tilde{O}(n)$, while keeping the circuit size (depth and total number of gates) $\tilde{O}(n^{\frac{3}{2}})$.

Table 2: Expected costs of factoring n-bit RSA integers according to [16].

	Factoring n-bit RSA integer		
	$n = 3072$	$n = 7680$	$n = 15360$
Logical Qubits	9287	23238	46507
T-Count	$1.25 \cdot 2^{27}$	$1.36 \cdot 2^{31}$	$1.47 \cdot 2^{34}$
Circuit Depth	$1.12 \cdot 2^{32}$	$1.76 \cdot 2^{34}$	$1.76 \cdot 2^{36}$

Table 3: Asymptotic costs of factoring n-bit RSA integers according to [16].

	Factoring n-bit RSA integer
Logical Qubits	$3n + 0.002n \lg n$
T-Count	$0.3n + 0.0005n^3 \lg n$
Circuit Depth	$500n^2 + n^2 \lg n$

4.3 Elliptic curve cryptography

Alongside the factoring algorithm, Shor also presented an efficient quantum algorithm for solving the discrete logarithm problem in a multiplicative group of a prime field $\mathbb{F}_p$ [31]. This was later made appropriate to the setting of elliptic curves by Proos and Zalka in [23]. It is this latter setting which is relevant to contemporary cryptography. Adapting Shor's algorithm to elliptic curves (or any abelian group) is straightforward provided the group operation can be implemented efficiently. As pointed out in [18], it is the reversible implementation of the group operation which contributes the most to the overall cost (in terms of resources) of the quantum circuit.

As per the scope defined in Section 4.1, works cited here focus only on elliptic curves over prime fields [11, 12, 18, 23], while neglecting binary fields. Given that curves over binary fields have been deprecated by NIST [21], they shall not receive treatment in this paper either. Interested readers may look to [2] to learn more. Thus we shall henceforth be considering an elliptic curve E over a field $\mathbb{F}_p$ with p prime and n denoting the bit-length of p. Also, without loss of generality, we may assume that E is a Weierstrass curve despite Montgomery and Edwards curves being commonly used. That follows from the fact that there exist birational equivalence relations between (twisted) Edwards curves and Montgomery curves, with every instance of the latter being equivalent to some Weierstrass curve further still [3,9]. We can thus restrict our attention to Weierstrass curves wherever this level of detail is necessary.

Roetteler et al. in [27] have estimated that Shor's algorithm for breaking ECDLP on E would require:

$$(448\lg(n) + 4090)n^3. \tag{3}$$

Toffoli gates (recall that each Toffoli gate corresponds to 7 T gates). Häner et al. [18] improve on the results of Roetteler et al. by reducing the number of logical qubits and Toffoli gates and providing an asymptotic estimate of the number of T gates:

$$436n^3 + o(n^3). \tag{4}$$

Häner et al. also present various trade-offs possible when implementing Shor's algorithm for ECDLP, optimizing, e.g., for circuit depth (see Table 4) or its T-depth.

Table 4: Expected costs of solving ECDLP according to [18].

	Solving DLP on an n-bit elliptic curve		
	$n = 256$	$n = 384$	$n = 512$
Logical Qubits *(optimized for width)*	2124	3151	4258
T-Count *(optimized for width)*	$1.72 \cdot 2^{32}$	$1.51 \cdot 2^{34}$	$1.82 \cdot 2^{35}$
Circuit Depth *(optimized for width)*	$1.89 \cdot 2^{32}$	$1.77 \cdot 2^{34}$	$1.09 \cdot 2^{36}$
Logical Qubits *(optimized for T-count)*	2619	3901	5273
T-Count *(optimized for T-count)*	$1.08 \cdot 2^{31}$	$1.74 \cdot 2^{32}$	$1.00 \cdot 2^{34}$
Circuit Depth *(optimized for T-count)*	$1.85 \cdot 2^{31}$	$1.31 \cdot 2^{33}$	$1.54 \cdot 2^{34}$
Logical Qubits *(optimized for depth)*	2871	4278	5789
T-Count *(optimized for depth)*	$1.34 \cdot 2^{32}$	$1.13 \cdot 2^{34}$	$1.43 \cdot 2^{35}$
Circuit Depth *(optimized for depth)*	$1.40 \cdot 2^{27}$	$1.48 \cdot 2^{28}$	$1.27 \cdot 2^{29}$

5 Forecasting evolution of quantum computers: when practical attacks will be possible

Quantum computing technology, possesses a limited historical track record, and predictions concerning its future development largely rely on quantum experts' educated guesses, occasionally supported by more substantiated arguments. In literature, the performance of quantum computers is frequently monitored via the following quantities:

- average two-qubit-gate error rate,
- number of physical qubits in a system,

Table 5: Asymptotic costs of solving ECDLP according to [18].

	Solving DLP on an n-bit elliptic curve
Logical Qubits *(optimized for width)*	$8n + 10.2 \cdot \lfloor \lg n \rfloor - 1$
T-Count *(optimized for width)*	$436n^3 - 1.05 \cdot 2^{26}$
T-Depth *(optimized for width)*	$120n^3 - 1.67 \cdot 2^{22}$
Logical Qubits *(optimized for T-count)*	$10n + 7.4 \cdot \lfloor \lg n \rfloor + 1.3$
T-Count *(optimized for T-count)*	$1115n^3/\lg n - 1.08 \cdot 2^{24}$
T-Depth *(optimized for T-count)*	$389n^3/\lg n - 1.70 \cdot 2^{22}$
Logical Qubits *(optimized for depth)*	$11n + 3.9 \cdot \lfloor \lg n \rfloor + 16.5$
T-Count *(optimized for depth)*	–
T-Depth *(optimized for depth)*	$285n^2 - 1.54 \cdot 2^{17}$

- number of logical qubits in a system.

Remark 2. In our current understanding of quantum computing, the primary bottleneck for executing any of the aforementioned attacks is likely to be the maximal circuit depth. Unfortunately, no methodology has been proposed to forecast the evolution of quantum computers that adequately takes into account the circuit depth. Moreover, researchers working on quantum computers rarely share information concerning this topic. Therefore, we base our analyses on a more freely accessible metric.

5.1 Forecasting based on a statistical model

To the best of our knowledge, the most comprehensive assessment of future quantum computing progress based on statistical modeling was presented in a 2020 article by Sevilla and Riedel [29]. They gathered all available information (scientific articles and enterprises' marketing alike) on quantum computers from 2003 to 2020 [30]. It is vital to note that such data is subject to significant noise and bias primarily because the decision to report findings or failures lies with the researchers.

In light of this, Sevilla et al. devised the *generalized logical qubits* (GLQ) index, which estimates the number of logical qubits that will be available after accounting for the error-correction overheads [29].

The GLQ is expressed as follows:

$$N_{GLQ} = N_{PQ} \left[4 \cdot \frac{\log\left(\sqrt{10}\frac{e_P}{e_L}\right)}{\log\left(\frac{e_{th}}{e_P}\right)} + 1 \right]^{-2} \tag{5}$$

where N_{PQ} represents the number of physical qubits, e_P is the two-qubit gate error rate, $e_{th} = 10^{-2}$ denotes the approximate threshold error under which fault-tolerance becomes viable for the surface code, and $e_L = 10^{-18}$ represents the acceptable logical error rate. To put it simply, this formula utilizes the number of physical qubits and the two-qubit gate error rate to estimate the number of logical qubits available after factoring in the error-correction overhead. In essence, N_{GLQ} models the development of FTQC over time and should represent the quantum computer's "real-world" capability to perform computations. The authors of [29] claim that formula 5 was coined to fit data well for devices that have achieved two-qubit gate error rates below the fault-tolerance threshold mentioned above. They propose a *multivariate log-linear* model that takes a date as input and outputs a distribution for the combination of metrics that quantum computers around that date are likely to represent. The model assumes a linear relation between time and the logarithms of N_{PQ} and e_P, thus giving an exponential relation between time and N_{GLQ}.

Our estimation. We extended the dataset used in [29] by 25% through the addition of 16 of the most recently published pieces of information regarding quantum processors. By doing so we allowed predictions to represent real-world data more accurately. It is noteworthy that many of the state-of-the-art models currently incorporate technologies other than superconductors, therefore we trained the model on all types of physical realizations. For the bootstrapping process, we used the top 15% of samples with the highest N_{GLQ}. Despite introducing certain refinements to the parameter assumptions of the original model, the resultant curve still exhibits an overestimation of N_{GLQ} for all data points, see Figure 1. Prediction of N_{GLQ} for the coming years is presented in Table 6.

Table 6: Expected N_{GLQ} for the coming years.

Year	2035	2040	2045	2050	2055	2060	2065	2070
Predicted N_{GLQ}	2	6	24	96	376	1450	5495	20542

6 Recommendations and closing remarks

According to the estimations presented in Tables 4 and 5, as well as the results summarized in Table 6, we claim that many classical algorithms currently in use may continue to be used as the well-studied, conservative fallbacks alongside (relatively experimental) post-quantum instances. Such hybrids are then expected to withstand quantum attacks until the year 2065, even if their post-quantum components have been broken classically by then. The following RSA parameters are recommended:

- at NIST Security Level 1: the RSA modulus N being a 3072-bit integer,
- at NIST Security Level 3: the RSA modulus N being a 7680-bit integer,
- at NIST Security Level 5: the RSA modulus N being a 15360-bit integer.

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63778-0_27

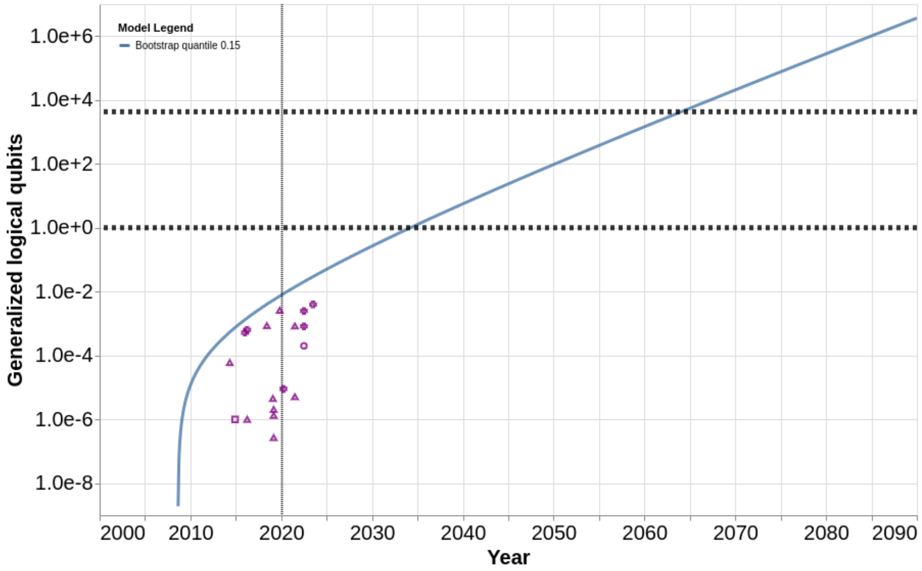

Fig. 1: N_{GLQ} prediction based on a statistical model. Note that the y-axis uses a logarithmic scale. Horizontal dotted lines are, from the bottom, $y = 1$ (one generalized logical qubit) and $y = 4258$ ("breaking" a 512-bit elliptic curve). Different markers denote different physical qubit realizations; for details, see [29].

As far as elliptic curve-based cryptosystems are concerned, forecasts in 1 show that 512-bit curves should be used to ensure security until year 2065. Smaller elliptic curve groups (e.g., 256-bit) are predicted to withstand quantum attacks until the year 2060. (Note that RSA is more secure against quantum adversaries than elliptic curve-based schemes at a similar classical security level, as already pointed out in [18].) After year 2065 it is nearly impossible to estimate the speed of evolution of quantum computers and so no further predictions are given.

Note that from a number of possible methodologies for studying the pairing of post-quantum schemes with classical ones, we have chosen the one which gives the most practical results (cf. Section 4.1), in the sense that it is not so pessimistic as to require exceedingly large group orders (elliptic curve or RSA). We believe, however, that this approach (albeit seemingly best-effort) is the most pertinent to the problem at hand since, as pointed out already, hybridization and parallel use of classical and post-quantum cryptography is to be thought of as a temporary measure for the transition period into the full-blown quantum era. Still, due to much uncertainty in the projections concerning the development of large-scale quantum computers, following the trends regularly and adapting accordingly is imperative. Caution is advised.

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63778-0_27

References

1. Amy, M., Maslov, D., Mosca, M., Roetteler, M.: A meet-in-the-middle algorithm for fast synthesis of depth-optimal quantum circuits. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems **32**(6), 818–830 (2013)
2. Banegas, G., Bernstein, D.J., van Hoof, I., Lange, T.: Concrete quantum cryptanalysis of binary elliptic curves. IACR Transactions on Cryptographic Hardware and Embedded Systems **2021**(1), 451–472 (Dec 2020). https://doi.org/10.46586/tches.v2021.i1.451-472, https://tches.iacr.org/index.php/TCHES/article/view/8741
3. Bernstein, D.J., Birkner, P., Joye, M., Lange, T., Peters, C.: Twisted Edwards curves. Cryptology ePrint Archive, Paper 2008/013 (2008), https://eprint.iacr.org/2008/013, https://eprint.iacr.org/2008/013
4. Beullens, W.: Breaking Rainbow takes a weekend on a laptop. Cryptology ePrint Archive, Paper 2022/214 (2022), https://eprint.iacr.org/2022/214, https://eprint.iacr.org/2022/214
5. Boudot, F., Gaudry, P., Guillevic, A., Heninger, N., Thomé, E., Zimmermann, P.: The state of the art in integer factoring and breaking public-key cryptography. IEEE Security Privacy **20**(2), 80–86 (2022). https://doi.org/10.1109/MSEC.2022.3141918
6. Castryck, W., Decru, T.: An efficient key recovery attack on SIDH. In: Hazay, C., Stam, M. (eds.) Advances in Cryptology – EUROCRYPT 2023. pp. 423–447. Springer Nature Switzerland, Cham (2023)
7. Chen, L., Moody, D., Liu, Y.: NIST post-quantum cryptography standardization. Transition **800**(131A), 164 (2017)
8. Chen, Y.: Quantum algorithms for lattice problems. Cryptology ePrint Archive, Paper 2024/555 (2024), https://eprint.iacr.org/2024/555, https://eprint.iacr.org/2024/555
9. Costello, C., Smith, B.: Montgomery curves and their arithmetic: The case of large characteristic fields. Cryptology ePrint Archive, Paper 2017/212 (2017), https://eprint.iacr.org/2017/212, https://eprint.iacr.org/2017/212
10. Deutsch, D.: Quantum theory, the Church-Turing principle and the universal quantum computer. Proceedings of the Royal Society of London A **400**, 97–117 (1985)
11. Ekerå, M.: Quantum algorithms for computing general discrete logarithms and orders with tradeoffs. Journal of Mathematical Cryptology **15**(1), 359–407 (2021). https://doi.org/doi:10.1515/jmc-2020-0006, https://doi.org/10.1515/jmc-2020-0006
12. Ekerå, M.: Revisiting Shor's quantum algorithm for computing general discrete logarithms (2023)
13. Ekerå, M., Håstad, J.: Quantum algorithms for computing short discrete logarithms and factoring RSA integers. pp. 347–363 (02 2017). https://doi.org/10.1007/978-3-319-59879-6_20
14. Feynman, R.P.: Simulating physics with computers. International Journal of Theoretical Physics **21**(6), 467–488 (1982)
15. Fowler, A.G., Mariantoni, M., Martinis, J.M., Cleland, A.N.: Surface codes: Towards practical large-scale quantum computation. Physical Review A **86**(3) (Sep 2012). https://doi.org/10.1103/physreva.86.032324, http://dx.doi.org/10.1103/PhysRevA.86.032324
16. Gidney, C., Ekerå, M.: How to factor 2048 bit RSA integers in 8 hours using 20 million noisy qubits. Quantum **5**, 433 (Apr 2021). https://doi.org/10.22331/q-2021-04-15-433, https://doi.org/10.22331/q-2021-04-15-433

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63778-0_27

17. Grover, L.K.: A fast quantum mechanical algorithm for database search (1996)
18. Häner, T., Jaques, S., Naehrig, M., Roetteler, M., Soeken, M.: Improved quantum circuits for elliptic curve discrete logarithms. In: Ding, J., Tillich, J.P. (eds.) Post-Quantum Cryptography. pp. 425–444. Springer International Publishing, Cham (2020)
19. Häner, T., Roetteler, M., Svore, K.M.: Factoring using 2n+2 qubits with Toffoli based modular multiplication (2017)
20. Maino, L., Martindale, C.: An attack on sidh with arbitrary starting curve. Cryptology ePrint Archive, Paper 2022/1026 (2022), https://eprint.iacr.org/2022/1026, https://eprint.iacr.org/2022/1026
21. National Institute of Standards and Technology: Digital signature standard (DSS). https://csrc.nist.gov/pubs/fips/186-5/final (2023)
22. National Institute of Standards and Technology: Recommendations for discrete logarithm-based cryptography: Elliptic curve domain parameters. https://csrc.nist.gov/pubs/sp/800/186/final (2023)
23. Proos, J., Zalka, C.: Shor's discrete logarithm quantum algorithm for elliptic curves (2004)
24. Ragavan, S., Vaikuntanathan, V.: Optimizing space in Regev's factoring algorithm. Cryptology ePrint Archive, Paper 2023/1501 (2023), https://eprint.iacr.org/2023/1501, https://eprint.iacr.org/2023/1501
25. Regev, O.: An efficient quantum factoring algorithm (2023). https://doi.org/10.48550/ARXIV.2308.06572, https://arxiv.org/abs/2308.06572
26. Robert, D.: Breaking SIDH in polynomial time. In: Hazay, C., Stam, M. (eds.) Advances in Cryptology – EUROCRYPT 2023. pp. 472–503. Springer Nature Switzerland, Cham (2023)
27. Roetteler, M., Naehrig, M., Svore, K.M., Lauter, K.: Quantum resource estimates for computing elliptic curve discrete logarithms. In: Takagi, T., Peyrin, T. (eds.) Advances in Cryptology – ASIACRYPT 2017. pp. 241–270. Springer International Publishing, Cham (2017)
28. Selinger, P.: Quantum circuits of t-depth one. Physical Review A **87**(4), 042302 (2013)
29. Sevilla, J., Riedel, C.J.: Forecasting timelines of quantum computing (2020). https://doi.org/10.48550/ARXIV.2009.05045, https://arxiv.org/abs/2009.05045
30. Sevilla, J., Riedel, C.J.: Quantum computing progress - data. https://docs.google.com/spreadsheets/d/1pwb4gf0FxlxgfVhtXTaqEGS9b7FwssstsJ0v7Zb1naQ0/edit#gid=0 (2020), 2023
31. Shor, P.: Algorithms for quantum computation: discrete logarithms and factoring. In: Proceedings 35th Annual Symposium on Foundations of Computer Science. pp. 124–134 (1994). https://doi.org/10.1109/SFCS.1994.365700
32. Stebila, D., Fluhrer, S., Gueron, S.: Hybrid key exchange in TLS 1.3. Tech. rep., Internet Engineering Task Force (Sep 2023), https://datatracker.ietf.org/doc/draft-ietf-tls-hybrid-design/09/