MSRDG International Journal of Computer Scientific Technology & Electronics Engineering ISSN: 3107 - 426X
Volume 2 Issue 5, 01-05, October 2026
©2026 Multi-Sense Research Developing Group
https://doi.org/10.66037/MSRDG-IJCSTEE/V2I5P101

An Evaluation of Computational Complexity of Shor's Algorithm on Surface Codes for Factorization

M. Pradeep[1], C. Ganesh[2]
[1,2] *Department of CSE, Stain Martin College, India.*
[1]*pradeep541@gmail.com*

Received: 05.09.2026 Revised:16.10.20256 Accepted: 25.10.2026 Published:31.10.2026

Abstract - *Shor's algorithm factors an n-bit integer in time polynomial in n, which would break RSA. Its asymptotic gate complexity, however, says little about what a real machine must provide, because the dominant cost is the overhead of quantum error correction. This report evaluates the computational complexity of Shor's algorithm in three layers: the logical algorithm (gate count, depth and qubits of order finding and modular exponentiation), the surface-code layer (code distance, logical error suppression, lattice surgery and magic-state production) and the physical resource layer (qubits and wall-clock time). We present a compact analytic resource model, calibrated to the published Gidney-Ekerå estimate for RSA-2048, and use it to study scaling with key size, physical error rate and reaction time. For RSA-2048 the model gives about 2.6 billion Toffoli gates, roughly 6,200 logical qubits, a code distance near 29 and about 20 million physical qubits for a runtime of about seven hours at a 0.1% physical error rate. Resource requirements grow as n3 in gates and runtime, while physical-qubit overhead is dominated by the code distance, which rises steeply as the physical error rate approaches the threshold. We also summarize how algorithmic and code improvements through 2026 have reduced published estimates by orders of magnitude.*

Keywords - Shor's algorithm, integer factorization, surface code, fault tolerance, lattice surgery, magic states, resource estimation, computational complexity.

1. Introduction

The security of RSA rests on the belief that factoring a large semiprime N = pq is intractable. The best known classical method, the general number field sieve (GNFS), runs in sub-exponential time, roughly exp((64/9)1/3 (ln N)1/3 (ln ln N)2/3) [22]. Its practical reach is illustrated by the factorization of the 829-bit RSA-250 in 2020, which took on the order of 2,700 core-years [21]; a 2048-bit modulus is out of reach by many orders of magnitude. Shor's 1994 algorithm places factoring in BQP, the class of problems solvable in polynomial time by a quantum computer [1]. Polynomial time is not the same as cheap. A fault-tolerant quantum computer must spend many physical qubits and many error-correction rounds to protect each logical operation, and the algorithm needs billions of non-Clifford gates. This report therefore asks a concrete question: how does the complexity of Shor's algorithm translate into physical resources when it is run on surface codes?

1.1 Contributions and approach

- a derivation of the gate, depth and qubit complexity of Shor's algorithm and its major variants;
- a self-contained description of surface-code fault tolerance, focusing on the parameters that determine cost;
- a calibrated analytic resource model, with explicit assumptions, evaluated for key sizes from 1024 to 4096 bits and across hardware parameters;
- a comparison with published estimates from 2012 to 2026, including recent architectures beyond the planar surface code

This is an open access article is licensed under CC BY-NC-ND 4.0 (https://creativecommons.org/licenses/by-nc-nd/4.0/)

2. Shor's Algorithm and its Logical Complexity

2.1 From factoring to order finding

To factor an odd composite N, pick a random a coprime to N and find the multiplicative order r, the smallest positive integer with ar = 1 (mod N). If r is even and ar/2 is not congruent to -1 (mod N), then gcd(ar/2 ± 1, N) yields a nontrivial factor. For N with at least two distinct odd prime factors, this succeeds for at least half of the random choices of a, so a constant number of repetitions suffices. Everything except order finding is classical and polynomial: gcd computations, and continued-fraction expansion of the measured phase.

2.2 Quantum order finding

Order finding is quantum phase estimation applied to the unitary U|x> = |ax mod N> [28]. A register of t » 2n qubits is placed in superposition, controlled operations compute ax mod N into a second n-qubit register, and an inverse quantum Fourier transform (QFT) on the first register concentrates amplitude near multiples of 2t /r. Measuring and applying continued fractions recovers r with constant probability per run (repeating O(log log N) times in the worst case gives confidence). The two costly pieces are:

• **Modular exponentiation:** computing ax mod N for an n-bit N and a ~2n-bit exponent. Done by repeated squaring, it consists of ~2n controlled modular multiplications by classically known constants, each built from O(n) modular additions of O(n) gates (schoolbook), so the total is O(n3) gates.

• **QFT:** exactly O(n2) gates, or O(n log n) when small-angle rotations are dropped [29]. This is negligible next to exponentiation

2.3 Experimental context

The first experimental factoring of 15 used a 7-qubit NMR device in 2001 [19]. Later demonstrations on photonic, ion and superconducting devices reached 15 and 21, but many used circuits that were simplified using prior knowledge of the answer, which is not scalable [20]. No demonstration to date involves error-corrected, compiled Shor circuits at cryptographic scale; the gap between those experiments and RSA-2048 is the fault-tolerance overhead analysed below.

3. Surface-Code Fault Tolerance

3.1 The code

The surface code stores one logical qubit in a 2D grid of physical qubits using only nearest-neighbor interactions [7][8][9]. In the common rotated layout with code distance d, a patch has d2 data qubits and d2 - 1 measurement qubits, so about 2d2 physical qubits per logical qubit (Figure 1). Measurement qubits repeatedly check parity (stabilizer) conditions; the pattern of violated checks (the syndrome) is decoded to infer and correct errors. A patch of distance d corrects up to (d - 1)/2 errors per round, and a logical error requires a chain of at least d physical errors. Page

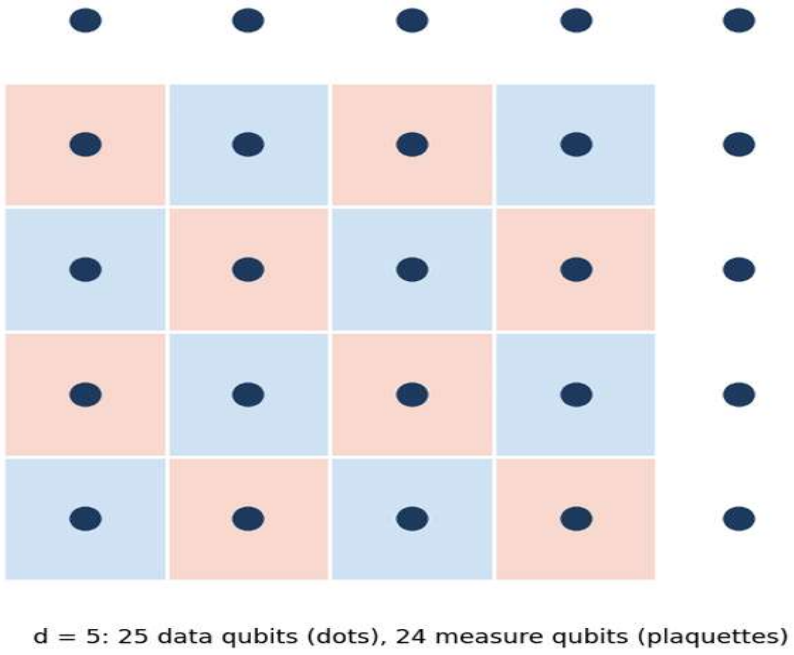


Fig. 1 A distance-5 rotated surface-code patch: 25 data qubits (dots) and 24 measurement qubits (one per shaded plaquette)

3.2 Error suppression

If the physical error rate p is below the threshold p th (about 0.5 to 1% for circuit-level noise, depending on the decoder), the logical error probability per patch per round falls exponentially in d [6]: e(d, p) » A · (p / p th) (d+1)/2 , A » 0.1, p th » 1%. Each increase of d by 2 divides the logical error rate by p th /p, for example by 10 at p = 0.1%. Below-threshold behavior has now been shown experimentally: Google Quantum AI reported logical error suppression by a factor of about 2 per distance step from distance 3 to 7 on a superconducting processor [15]. That experiment demonstrates the mechanism, but at logical error rates around 10-3 per cycle, far above the ~10-15 needed for RSA-2048 (Section 5)

3.3 Logical operations: lattice surgery

Clifford operations between patches are performed by lattice surgery: merging and splitting neighboring patches through boundary measurements [10][11][25]. A logical CNOT takes about d rounds and requires routing space between patches, so the chip is divided into data patches, routing lanes and magic-state factories. Each logical timestep costs d syndrome-extraction rounds, which is the reason code distance enters runtime as well as qubit count when operations are not reaction-limited.

3.4 Non-Clifford gates and magic states

Clifford gates alone are classically simulable, so Shor's algorithm needs a non-Clifford resource. The surface code has no transversal non-Clifford gate; instead, noisy |T> or |CCZ> states are distilled to high fidelity in dedicated factories [12] and consumed through gate teleportation. A Toffoli costs one CCZ state (equivalently four T states). Distillation factories occupy a large share of the machine, and the consumption rate of magic states sets the speed. Magic state cultivation [13] reduces this cost for target error rates by growing high-fidelity states inside a small surface-code patch, and is one of the main reasons for the 2025 reduction in qubit estimates [4]

3.5 Decoding and reaction time

A classical decoder must process each round's syndrome in real time to avoid a growing backlog. Matching-based decoders

now run fast enough for large patches [26]. Shor-type circuits are slowed further by the reaction limit: a Toffoli's teleportation correction depends on a measurement outcome, so consecutive Toffolis in a dependent chain cannot run faster than the measure-decode-feed-forward latency, taken as about 10 μs in the Gidney-Ekerå model [3]. With enough factories, runtime is therefore approximately (Toffoli count) × (reaction time).

4. Evaluation

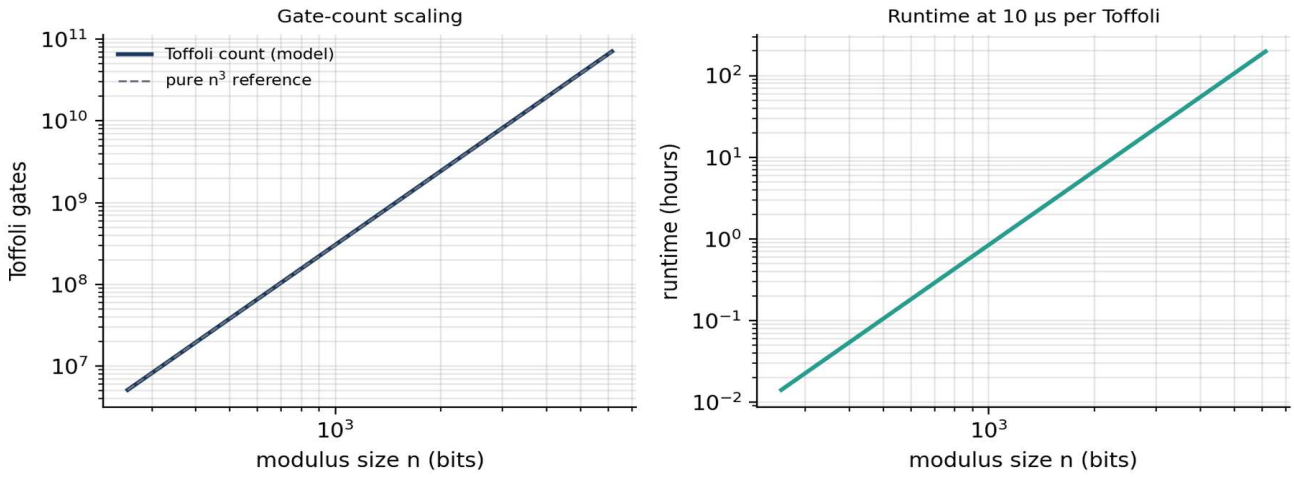


Fig.2 Toffoli count and runtime versus modulus size. The curve tracks n3 (dashed) closely; the lg n term gives a slight upward bend.

Doubling the key size from 1024 to 2048 bits multiplies the Toffoli count and runtime by about eight, matching the cubic scaling, while physical qubits grow by about 2.3 times: logical qubits double and the code distance increases only slightly (the distance depends on the logarithm of the spacetime volume). This logarithmic dependence is why surface codes handle long computations well: a factor of 8 more operations costs only about two more distance steps. In practice, moving from RSA-2048 to RSA-4096 is not a security gain against quantum attack in the way it is against classical attack. The model gives about 2.3 times the qubits and 8 times the runtime, which is a modest polynomial cost, whereas against GNFS the same doubling raises the cost by many orders of magnitude.

4.1 Sensitivity to the physical error rate

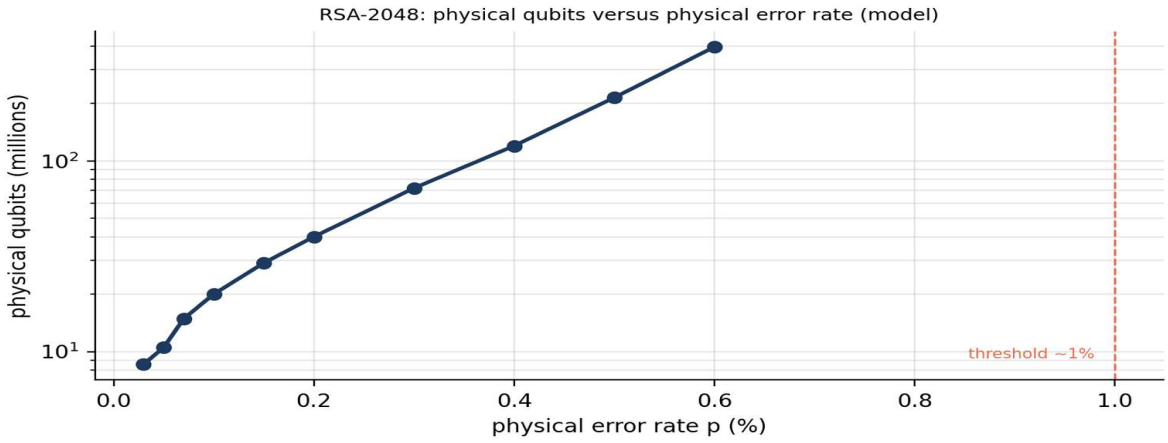


Fig. 3 Physical qubits versus physical error rate. Cost rises steeply as p approaches the threshold, because the required distance diverges

Halving the error rate from 0.1% to 0.05% roughly halves the physical qubits, since the distance drops from 29 to 21. Doubling it to 0.2% roughly doubles the cost, and at 0.5%, close to the threshold, the model needs a distance near 95 and over 200 million qubits. Physical fidelity is therefore the most powerful lever, which is why hardware roadmaps focus on two-qubit gate errors..

5. Conclusion

Shor's algorithm has cubic gate complexity in the key size, and on surface codes this becomes billions of magic-state-consuming Toffoli gates, thousands of logical qubits and tens of millions of physical qubits in the 2019 baseline. The evaluation highlights three structural facts: runtime scales as n3 and is set by classical reaction time; the code distance, and hence qubit overhead, depends only logarithmically on computation size but strongly on physical error rate; and magic states and routing make up roughly half of the machine.

At the same time, estimates are falling quickly through algorithmic and architectural advances, from about a billion qubits in 2012 to under a million in 2025 on the same surface-code hardware model, with 2026 preprints claiming further order-of-magnitude reductions using different codes. For risk planning the message is that the physical-qubit requirement is a moving target, that the underlying n3 scaling means increasing RSA key length offers little protection, and that migration to post-quantum cryptography should be planned on the assumption that estimates will keep improving.

References

[1] P. W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. SIAM J. Comput., 26(5):1484-1509, 1997.
[2] S. Beauregard. Circuit for Shor's algorithm using 2n+3 qubits. Quantum Information and Computation, 3(2):175-185, 2003.
[3] C. Gidney and M. Ekerå. How to factor 2048 bit RSA integers in 8 hours using 20 million noisy qubits. Quantum, 5:433, 2021.
[4] C. Gidney. How to factor 2048 bit RSA integers with less than a million noisy qubits. arXiv:2505.15917, 2025.
[5] C. Gidney. Windowed quantum arithmetic. arXiv:1905.07682, 2019.
[6] A. G. Fowler, M. Mariantoni, J. M. Martinis, and A. N. Cleland. Surface codes: towards practical large-scale quantum computation. Phys. Rev. A, 86:032324, 2012.
[7] A. Yu. Kitaev. Fault-tolerant quantum computation by anyons. Annals of Physics, 303(1):2-30, 2003.
[8] E. Dennis, A. Kitaev, A. Landahl, and J. Preskill. Topological quantum memory. J. Math. Phys., 43:4452-4505, 2002.
[9] S. B. Bravyi and A. Yu. Kitaev. Quantum codes on a lattice with boundary. arXiv:quant-ph/9811052, 1998.
[10] D. Horsman, A. G. Fowler, S. Devitt, and R. Van Meter. Surface code quantum computing by lattice surgery. New J. Phys., 14:123011, 2012.
[11] D. Litinski. A game of surface codes: large-scale quantum computing with lattice surgery. Quantum, 3:128, 2019.
[12] S. Bravyi and A. Kitaev. Universal quantum computation with ideal Clifford gates and noisy ancillas. Phys. Rev. A, 71:022316, 2005.
[13] C. Gidney, N. Shutty, and C. Jones. Magic state cultivation: growing T states as cheap as CNOT gates. arXiv:2409.17595, 2024.
[14] C. Gidney, M. Newman, P. Brooks, and C. Jones. Yoked surface codes. arXiv:2312.04522, 2023.
[15] Google Quantum AI. Quantum error correction below the surface code threshold. Nature, 638:920-926, 2025.
[16] M. Ekerå and J. Håstad. Quantum algorithms for computing short discrete logarithms and factoring RSA integers.