Grover's Algorithm Against AES-128:

Infeasibility of Quantum Brute-Force Search

Ji Won*and Naveed A.A.

PakCrypt NPO, Islamabad, Pakistan

April 22, 2026

Abstract

Grover's algorithm is frequently cited as the principal quantum threat to symmetric ciphers such as the Advanced Encryption Standard with a 128-bit key (AES-128). The commonly quoted claim that Grover "halves the effective key length" is mathematically correct but is often decoupled from the engineering realities of building, running, and error-correcting the required quantum circuit. This white paper develops, from first principles, the structure of Grover's algorithm, the precise nature of what it means to implement AES-128 as a quantum oracle, the role of entanglement and coherence throughout the computation, and the concrete resource estimates (logical qubits, physical qubits, circuit depth, gate count, and wall-clock time) that a realistic attack would demand. The paper concludes that while the $\sqrt{N}$ speedup is mathematically real and provably optimal for unstructured search, the cost of chaining together 2^{64} coherent AES evaluations vastly exceeds what any foreseeable physical quantum computer could accomplish. AES-128 therefore retains substantial practical security against Grover-based attacks, even under optimistic projections of quantum hardware progress, although long-term data confidentiality considerations still justify migration to AES-256.

---

1 Introduction and Motivation

The emergence of quantum computing has provoked considerable concern about the long-term viability of classical cryptographic primitives. The best-known quantum threats divide cleanly into two categories. Shor's algorithm poses an existential risk to asymmetric cryptosystems based on integer factorization and discrete logarithms, because it reduces those problems from super-polynomial to polynomial complexity. Grover's algorithm, by contrast, attacks unstructured search problems and is the canonical quantum technique for accelerating brute-force key recovery against symmetric ciphers. Because Grover provides only a quadratic (as opposed to exponential) speedup, its practical impact on symmetric cryptography is far more modest than Shor's impact on asymmetric cryptography, yet it is frequently misrepresented in both directions: sometimes as an imminent catastrophe and sometimes as an entirely negligible concern.

The purpose of this paper is to give a careful, self-contained technical account of what Grover's algorithm actually does when turned against AES-128, how its internal components must be assembled, what resources it requires, and why the engineering problem is far more severe than the abstract operation count suggests. The paper aims to equip a technically literate reader with a defensible intuition about both the theoretical power and the practical infeasibility of this attack.

*Corresponding author: won@pakcrypt.org

2 Grover's Algorithm: Structure and Main Steps

2.1 The Search Problem

Grover's algorithm solves the following abstract problem. Let $f : \{0,1\}^n \to \{0,1\}$ be a Boolean function, given as a black-box oracle, such that exactly one input $\omega \in \{0,1\}^n$ satisfies $f(\omega) = 1$. The task is to find ω. Classically, in the worst case one must evaluate f on roughly $N = 2^n$ inputs; on average, $N/2$ evaluations suffice. Grover's algorithm solves the problem using approximately $\frac{\pi}{4}\sqrt{N}$ oracle evaluations, and this bound is provably optimal: no quantum algorithm can solve unstructured search in fewer queries, up to lower-order terms.

For the brute-force attack on AES-128, $n = 128$, and the oracle f is defined so that $f(k) = 1$ if and only if k is the unknown encryption key, typically verified by checking whether $\mathrm{AES}_k(p) = c$ for one or more known plaintext/ciphertext pairs (p, c).

2.2 The Two Operators of Grover

Grover's algorithm operates on an n-qubit register prepared in the uniform superposition

$$|s\rangle = \frac{1}{\sqrt{N}} \sum_{x \in \{0,1\}^n} |x\rangle ,$$

which is produced by applying a Hadamard transform $H^{\otimes n}$ to the all-zeros state. The algorithm then alternates two unitary operators.

The first is the *oracle operator* O_f, which marks the correct key by flipping the phase of its amplitude:

$$O_f |x\rangle = (-1)^{f(x)} |x\rangle .$$

Concretely, O_f leaves every state alone except $|\omega\rangle$, whose amplitude is multiplied by -1. Crucially, this phase flip is not something that can be "observed" on its own; it becomes useful only when combined with the second operator.

The second operator is the *diffusion operator* D, also called inversion about the mean:

$$D = 2 |s\rangle \langle s| - I.$$

Applied to an arbitrary superposition, D reflects every amplitude about the mean amplitude of the state. States whose amplitudes lie below the mean are raised, and states whose amplitudes lie above the mean are lowered.

One *Grover iteration* is the composite operator

$$G = D \cdot O_f.$$

2.3 The Geometric Picture

The behavior of G is most clearly understood in a two-dimensional subspace. Define $|\omega\rangle$ as the (normalized) target state and $|s'\rangle$ as the uniform superposition over all non-target states. The initial state $|s\rangle$ lies in the plane spanned by these two vectors, making a small angle θ with $|s'\rangle$ where

$$\sin\theta = \frac{1}{\sqrt{N}}, \qquad \theta \approx \frac{1}{\sqrt{N}} \text{ for large } N.$$

A single Grover iteration acts on this plane as a rotation by 2θ toward $|\omega\rangle$. After k iterations, the state makes an angle $(2k+1)\theta$ with $|s'\rangle$. The optimal number of iterations is therefore the value of k that brings this angle closest to $\pi/2$, which gives

$$k_{\text{opt}} \approx \frac{\pi}{4}\sqrt{N}.$$

For $n = 128$, this is

$$k_{\text{opt}} \approx \frac{\pi}{4} \cdot 2^{64} \approx 2^{63.65}.$$

At this point, measurement in the computational basis yields ω with probability very close to unity. Notably, performing more than k_{opt} iterations causes the state to rotate *past* the target, decreasing the success probability; Grover is not a "run it as long as you can" algorithm but one with a precise optimal stopping time.

3 The AES-128 Oracle: What Quantum Implementation Entails

The abstract description of the oracle O_f hides where nearly all of the physical cost lives. Implementing O_f requires constructing a quantum circuit that evaluates AES-128 reversibly and coherently on a superposition of all possible keys.

3.1 Reversibility and the Need for Uncomputation

Quantum computation is, except for measurement, unitary and therefore reversible. A classical AES implementation freely overwrites intermediate values and discards temporary data; a quantum implementation cannot do so, because discarding information would entangle the computational register with an inaccessible environment and destroy the interference patterns that Grover relies upon.

Every non-reversible operation in classical AES must be replaced with a reversible quantum analogue. In practice, this means that the oracle circuit computes $\text{AES}_k(p)$ into an ancilla register, compares the result to the known ciphertext c to produce a single-bit flag, uses that flag to apply a phase flip conditional on equality, and then *uncomputes* the AES evaluation and all ancillas by running the inverse circuit. Without this uncomputation, the leftover intermediate state would remain entangled with the key register and prevent the diffusion step from working.

The key register itself holds a superposition over all 2^{128} candidate keys, while auxiliary registers hold intermediate values during AES evaluation. A single oracle call therefore corresponds to (compute AES) $\rightarrow$ (compare and flag) $\rightarrow$ (phase flip) $\rightarrow$ (uncompute AES), roughly doubling the effective gate count relative to the forward computation alone.

3.2 Quantum Implementation of AES Components

AES-128 consists of ten rounds, each of which applies four transformations: SubBytes, ShiftRows, MixColumns (omitted in the final round), and AddRoundKey. Three of these four are linear over GF(2) and therefore admit inexpensive quantum implementations: they can be realized with CNOT gates and, where needed, ancilla qubits, with gate counts polynomial in the block size.

The SubBytes step is the non-linear component and is responsible for the bulk of the oracle's cost. SubBytes applies the AES S-box to each of the 16 bytes of the state. The S-box is an 8-bit-to-8-bit substitution that can be described as inversion in $\text{GF}(2^8)$ followed by an affine transformation. Implementing this reversibly requires either a lookup-table approach using Toffoli (controlled- controlled-NOT) gates or an arithmetic circuit that performs the finite-field inversion directly. Both approaches have been extensively studied, and the best known constructions use on the order of a few hundred Toffoli gates per S-box, with circuit depth in the tens to low hundreds.

AES-128 performs $16 \times 10 = 160$ S-box evaluations per encryption for the data path, plus additional S-box calls in the key schedule. A full AES-128 encryption therefore requires on the order of 10^4 to 10^5 Toffoli gates in the best published reversible constructions, with the Toffoli count dominating the resource cost because Toffolis are dramatically more expensive than Clifford gates in fault-tolerant implementations.

3.3 Qubit Budget for the Oracle

A conservative estimate of the logical qubits required to implement the AES-128 oracle includes the 128-qubit key register, a 128-qubit plaintext/state register, ancilla qubits for S-box evaluation (which depend on the chosen construction and range from a few hundred to a few thousand), and additional workspace for the key schedule and comparison logic. Published resource estimates place the total logical qubit requirement for the AES-128 Grover oracle at roughly 3,000 to 6,000 logical qubits, depending on space-time trade-offs. This is before any error correction overhead is applied.

4 The Role of Entanglement

Entanglement is not an optional accelerant in Grover's algorithm; it is the structural mechanism by which the computation works at all. Three aspects deserve emphasis.

First, the initial uniform superposition $|s\rangle$ is itself a product state, but as soon as the oracle begins computing AES on the key register and writing results into ancilla registers, entanglement is created between the key qubits and the ancilla qubits. If the ancillas were then measured, or if they were discarded without uncomputation, this entanglement would cause decoherence of the key register and destroy the quadratic speedup. The oracle must therefore uncompute all ancillas so that, at the end of each oracle call, the key register is left in a state that depends on the oracle's output only through the phase of each basis state.

Second, within the AES computation itself, each round spreads information from each byte of the state into every other byte, producing a highly entangled intermediate state. This intra-oracle entanglement is precisely the reversible analogue of the avalanche effect that gives AES its classical security. The quantum computer must maintain coherence across all of these entangled qubits for the duration of the oracle call.

Third, and most consequentially, the entire Grover iteration must be executed coherently, and 2^{64} iterations must be chained end-to-end without any measurement or decoherence until the final read-out. A single decoherence event at any point in those 2^{64} iterations will, with high probability, collapse the key register into a random state and force the computation to restart from the beginning. The demand on coherence time is therefore not the duration of one oracle call but the duration of 2^{64} such calls executed sequentially.

5 Amplitude Amplification in Detail

The mechanism by which Grover increases the amplitude of the target state deserves careful attention because it is the step that converts the phase flip produced by the oracle into an observable outcome.

After preparation of the uniform superposition, every basis state carries amplitude $1/\sqrt{N}$. The oracle flips the sign of the target amplitude, producing a state in which the target has amplitude $-1/\sqrt{N}$ and every other state has amplitude $+1/\sqrt{N}$. The mean amplitude across all N basis states is now

$$\mu = \frac{1}{N}\left[(N-1)\cdot\frac{1}{\sqrt{N}} + \left(-\frac{1}{\sqrt{N}}\right)\right] = \frac{N-2}{N\sqrt{N}} \approx \frac{1}{\sqrt{N}}.$$

The diffusion operator reflects each amplitude about this mean. Non-target states, whose amplitude slightly exceeds the mean, are lowered slightly. The target state, whose amplitude is $-1/\sqrt{N}$, is reflected to approximately $3/\sqrt{N}$ – roughly triple its starting magnitude with the correct sign.

Each iteration repeats this process. Because the amplitudes live on a two- dimensional circle parameterized by the angle θ, the process is a rotation, not a runaway amplification. The target

amplitude grows, passes through a maximum near k_{opt}, and then decreases again as the rotation continues past $\pi/2$. The success probability after k iterations is exactly

$$P_{\text{success}}(k) = \sin^2\big((2k+1)\theta\big).$$

For $N = 2^{128}$ and $k = \lfloor \frac{\pi}{4}\sqrt{N} \rfloor$, this probability is essentially indistinguishable from 1.

6 Circuit Size, Depth, and the Cost of a Single Oracle Call

To translate the 2^{64} iteration count into a physical resource estimate, one must multiply by the cost of a single oracle call, then apply the overhead of fault tolerance.

6.1 Logical-Level Cost

The leading published resource estimates for a reversible AES-128 circuit place the Toffoli count per oracle call in the range of 10^4 to a few times 10^5, with circuit depth (longest path of sequential gates) on the order of 10^3 to 10^4. The diffusion operator adds a comparatively negligible overhead.

A full Grover attack therefore entails approximately

$$N_{\text{Toffoli}}^{\text{total}} \;\approx\; 2^{64} \times 10^5 \;\approx\; 2^{81}$$

logical Toffoli gates, executed in a circuit of total depth on the order of $2^{64} \times 10^4 \approx 2^{77}$.

These numbers are at the logical level, meaning they assume ideal, error-free qubits with perfect gate fidelity. No such device has ever been built, and none can be built without error correction.

6.2 Fault-Tolerance Overhead

On any physically realizable quantum computer, gate errors and decoherence necessitate quantum error correction. The leading candidate architecture is the surface code, which encodes one logical qubit into a two-dimensional patch of physical qubits. To suppress logical error rates sufficiently for a computation of depth 2^{77}, one needs a logical error rate per gate of approximately $2^{-77} \approx 10^{-23}$, which in turn requires surface-code distances on the order of $d \approx 30$ to 50 and on the order of $d^2 \approx 10^3$ physical qubits per logical qubit, assuming physical error rates around 10^{-3} (which is at or beyond the current state of the art).

Toffoli and T gates are not directly available in the surface code; they must be synthesized via magic-state distillation, a procedure that itself consumes many additional physical qubits and gate operations per logical non-Clifford gate. Each logical T or Toffoli gate requires a distilled magic state, and magic-state distillation factories are among the largest single consumers of physical resources in any fault-tolerant architecture.

Combining these overheads produces physical resource estimates of:

- millions of physical qubits, often cited in the range 10^6 to 10^7 for the AES-128 Grover attack with realistic error rates;
- a physical gate count inflated to roughly 2^{87} to 2^{94} after accounting for error correction and magic-state distillation;
- a wall-clock time dominated by the sequential nature of the 2^{64} iterations, which at a microsecond per iteration would take hundreds of thousands of years, and at a nanosecond per iteration would still require hundreds of years of continuous, uninterrupted coherent operation.

7 Why Grover Is Impractical in Practice

Beyond the raw resource numbers, several structural properties of Grover's algorithm make it substantially less threatening than a naive reading of its asymptotic complexity would suggest.

7.1 The Inherent Serial Nature of Grover Iterations

Classical brute-force key search parallelizes trivially: M machines each test N/M keys and collectively complete the search M times faster. Grover's algorithm does not parallelize this way. Running M independent Grover instances, each searching a separate partition of the key space of size N/M, yields a total runtime of $\sqrt{N/M}$ per machine rather than $\sqrt{N}/M$. The parallel speedup is only $\sqrt{M}$, not M. Doubling the number of quantum computers reduces the attack time only by a factor of $\sqrt{2}$, not a factor of 2. This property dramatically weakens the economic attractiveness of Grover at scale and is one of the most underappreciated facts about the algorithm.

7.2 Extreme Coherence Requirements

Because the 2^{64} iterations must be performed coherently, the required coherence time grows not with the AES circuit depth but with the total depth of the chained iterations. Any decoherence event, any uncorrected gate error, and any inadvertent measurement anywhere in the computation destroys the quantum state and forces a restart. The surface code can in principle suppress errors to arbitrary levels, but the cost (in physical qubits and in magic-state distillation factories) grows with the target logical error rate.

7.3 Optimality Bound for Unstructured Search

A tempting hope is that improved quantum algorithms might reduce the $\sqrt{N}$ cost further. This hope is foreclosed by a provable lower bound: any quantum algorithm for unstructured search over N elements requires at least $\Omega(\sqrt{N})$ oracle queries. Grover is therefore not a starting point for future improvement against structureless problems; it is already the ceiling. Further speedups can arise only by exploiting structure in the cipher that AES was specifically engineered to lack.

7.4 Comparison with Shor's Algorithm

It is instructive to contrast Grover with Shor. Shor's algorithm attacks RSA by exploiting the algebraic structure of modular arithmetic. Its quantum circuit is of polynomial depth and must be run only a small, constant number of times. Coherence is required only across a single polynomial-depth circuit. For RSA-2048, Shor requires roughly 10^{10} to 10^{11} logical gate operations and a few thousand logical qubits – still a serious engineering undertaking, but polynomial in the input size.

Grover against AES-128, by contrast, requires 2^{81} logical gate operations that must all be executed in a single coherent run (or a small number of such runs). Remarkably, this means that on a future fault-tolerant quantum computer, RSA-2048 is *easier* to break than AES-128, reversing the classical ordering of their difficulty. This inversion is the reason post-quantum cryptography has focused almost exclusively on replacing public-key primitives while leaving symmetric primitives essentially intact, typically with a doubling of key length as a precaution.

8 Complete Resource Picture for Brute-Forcing AES-128 with Grover

Consolidating the previous sections, a realistic end-to-end attack on AES-128 using Grover's algorithm demands the following resources.

The logical qubit count lies in the range of a few thousand qubits, with specific published estimates typically falling between 3,000 and 6,000 logical qubits. These qubits must remain coherent, under active error correction, for the entire duration of the attack.

The physical qubit count, once surface-code encoding and magic-state distillation are included, rises to the range of 10^6 to 10^8 physical qubits depending on the assumed physical error rate and the desired attack time. Lower error rates and shorter attack times both push this number upward.

The total gate count is on the order of 2^{81} logical Toffoli or T gates, corresponding to something in the range of 2^{87} to 2^{94} physical gate operations after fault-tolerance overhead.

The circuit depth, which sets the wall-clock time, is on the order of 2^{77} sequential logical operations. Even at an aggressive logical gate rate of 10^{-9} seconds per gate, this corresponds to roughly 10^{16} seconds, or hundreds of millions of years. At a more realistic rate of 10^{-6} seconds per logical gate (which is closer to what error-corrected systems are expected to achieve), the time extends into the trillions of years.

It is possible to shorten the wall-clock time by using the parallel Grover strategy described above, but only at a square-root-inefficient scaling: dividing the problem across M machines reduces the time by $\sqrt{M}$ while multiplying the physical qubit cost by M. Reducing the attack to a practical duration of, say, one year would require on the order of 10^{16} to 10^{20} parallel machines, each with millions of physical qubits – a total hardware requirement so far beyond plausibility that it ceases to be meaningfully discussed in the engineering literature.

9 Additional Requirements Often Overlooked

A complete attack entails requirements beyond qubit counts and gate operations. Among the most significant are the need for a high-fidelity physical platform achieving physical gate error rates at or below the surface-code threshold of approximately 10^{-3}, sustained across billions of qubits for the full attack duration; a classical control system capable of decoding surface-code syndromes in real time at the rate at which they are generated (a non-trivial classical computing task in itself); a supply of magic states produced at a rate sufficient to feed the Toffoli and T gates consumed by the oracle, which typically requires dedicated distillation regions occupying a substantial fraction of the total qubit budget; and, finally, a stable cryogenic or otherwise controlled environment capable of hosting this machinery continuously for the duration of the attack without interruption.

Each of these requirements is itself a major open engineering challenge, and the absence of any one of them invalidates the attack.

10 Current State of Quantum Hardware

Against the resource estimates above, it is useful to set the current state of quantum computing hardware. Contemporary superconducting, trapped-ion, neutral- atom, and photonic platforms have demonstrated systems with roughly 100 to 1,000 physical qubits, with physical two-qubit gate error rates in the range of 10^{-2} to 10^{-3} depending on the platform. Demonstrations of fault-tolerant logical qubits have shown the ability to construct a handful of logical qubits with logical error rates modestly lower than the underlying physical error rate, representing important

proof-of-principle progress but leaving many orders of magnitude between current capability and what Grover against AES-128 would require.

The gap is not merely quantitative. Moving from 10^3 noisy physical qubits to 10^6 or 10^7 high-fidelity physical qubits supporting thousands of logical qubits entails solving problems in qubit fabrication, control electronics, cryogenic engineering, classical co-processing, and systems integration that are each substantial research programs in their own right. Even under optimistic projections of continued exponential growth in qubit counts and fidelity, reaching the scale required for a credible Grover attack on AES-128 is generally considered a multi-decade undertaking, and many credible researchers regard it as essentially uncertain whether the required scale will ever be reached.

11 Conclusions

Grover's algorithm is a mathematically elegant and provably optimal quantum algorithm for unstructured search. Applied to AES-128, it reduces the nominal brute-force cost from 2^{127} to approximately 2^{64} oracle evaluations, and this reduction is genuine and unavoidable in the quantum setting. In this narrow sense, the often-repeated claim that "Grover halves the effective key length of symmetric ciphers" is correct.

However, each of those 2^{64} oracle evaluations entails a full reversible implementation of AES-128, executed coherently and chained end-to-end without interruption, all under a regime of quantum error correction that itself inflates the physical resource requirements by several orders of magnitude. The inability to parallelize Grover efficiently, the enormous coherence demands, the physical qubit counts reaching into the millions or tens of millions, and the wall-clock time stretching into geological scales at realistic logical gate speeds together conspire to place the attack well outside the envelope of what any physically plausible quantum computer could accomplish in the foreseeable future.

The writer who argues that Grover poses no practical threat to AES-128 is, in this engineering sense, correct: the attack is a theoretical possibility but not a near-term or medium-term practical concern. At the same time, a fully rigorous assessment acknowledges that the quadratic speedup is real, that long-lived data encrypted today could in principle be attacked by future machines, and that the conservative migration to AES-256 – which restores a full 128-bit post-quantum security margin – is a prudent defense for data whose confidentiality must be preserved across very long horizons.

The correct posture, therefore, is neither alarm nor dismissal. Grover's algorithm sharpens our understanding of the true quantum security of symmetric ciphers; it informs the design choices embedded in current post-quantum cryptographic standardization; but it does not, under any realistic projection of quantum hardware, render AES-128 insecure in the practical sense. The cipher's strength against brute-force quantum search rests not on the absence of a quantum algorithm, but on the extraordinary cost of executing the one algorithm that exists.

End of White Paper