Quantum Computing and its Impact on Cryptography

Jocelin Su, Angela Zhang, Alice Zhu

May 2022

Contents

1 Introduction

The invention of computers transformed society, and computing has seen enormous improvements in efficiency over the past decades. Quantum computing seeks to take advantage of phenomena in quantum mechanics to perform tasks classical computers cannot, and may bring about sweeping changes to the cryptographic landscape. For example, quantum algorithms like Shor's for factoring and Grover's for search can break cryptographic schemes like RSA. As a result, researchers have been developing schemes that are secure in a post-quantum world, such as those that depend on the learning with errors problem. In our project, we conduct a survey of significant quantum algorithms, how they break security, and next steps in quantum computing and post-quantum security.

2 Background

Quantum mechanics is the science of describing how tiny particles, on the scale of electrons and photons, behave. Important but counter-intuitive quantum phenomena include superposition, entanglement, and uncertainty. Quantum computers use quantum particles to take advantage of these powerful concepts. We introduce qubits, the basic building block of quantum computation, and the relevant physical concepts of superposition and measurement.

Qubits

Quantum mechanics is a probabilistic theory, and a key fact is that a particle's attribute is uncertain until measured. We can describe the probability distribution of a particle's attribute, e.g. position, by a wavefunction $\psi(x)$. This is similar to a probability density function, but the probability of the particle being at position x_0 is instead $|\psi(x_0)|^2$. A useful fact is that this wavefunction can be decomposed into a combination of component functions — to be precise, a complex linear combination of eigenfunctions.

For our purposes, we consider a quantum bit, or qubit. A classical bit has two states, 0 and 1, but a qubit instead has a superposition of states 0 and 1. We explain superposition below.

Let us use bra-ket notation to describe a particle's states, where $|0\rangle$ denotes state 0 and $|1\rangle$ denotes state 1. We can also think of $|0\rangle$ as $\begin{bmatrix}1\\0\end{bmatrix}$ and $|1\rangle$ as $\begin{bmatrix}0\\1\end{bmatrix}$, vectors where the first index is the probability amplitude of being at 0 and the second index is the probability amplitude of being 1.

Superposition is the concept that a particle can have a combination of fractional states. For example, the qubit's wavefunction can be written as a linear combination of its component functions, $\alpha|0\rangle + \beta|1\rangle$, where α and β are complex numbers and are normalized so that $\alpha^2 + \beta^2 = 1$.

As a result, with superposition, a qubit can have infinitely many states "between" 0 and 1. Intuitively, this gives it more expressive power than a classical bit.

Another crucial part of computation is observing the state of the qubit. To record the state of a qubit, one has to measure or observe the qubit. Once measurement is performed, the qubit's wavefunction collapses, and it is determined to either definitely be in state 0 or definitely be in state 1. In practice, measurement may occur due to electric field or microwave resonance. Decoherence occurs when preliminary observation of qubits occurs due to its environment, which is a problem for quantum computers.

The goal in devising an algorithm for a quantum computer is for each wrong answer to have a low probability amplitude and for the correct answer to have a high probability amplitude. Then, the correct

answer will be measured with a high probability.

Quantum Computers

In real life, qubits are physical properties like the spin of an electron or the orientation of a photon. Actual quantum computers manipulate qubits through light or magnetic fields. Just like in classical computers, logical operations on qubits are represented through circuits and gates. While classical gates include operations like AND and NOT, quantum gates include operations like the Pauli gates and Hadamard gate. Mathematically, they are represented as unitary operators, which preserve inner product. Quantum algorithms may refer to quantum gates in their description.

3 Shor's Algorithm

We first discuss Shor's algorithms, one of the most famous quantum algorithms, developed in 1994 by Peter Shor. Many cryptosystems rely on the assumption that it is computationally infeasible to factor a large enough number N into two primes in polynomial time. The best known classical algorithm can do so, but only in superpolynomial time. Shor's takes advantage of the computation power of quantum machines to break this assumption. Given a sufficiently powerful quantum computer, Shor's can factor N in $O(\log(N)^3)$ time using $O(\log(N))$ space [8].

Process

Shor's algorithm has two components:

1. Reducing the factoring problem to the order-finding problem. This can be solved with a classical computer in polynomial time.
2. Using a quantum algorithm to solve the problem of order-finding in polynomial time.

Classical Component:

1. Compute $gcd(a, N)$, where $a < N$ is a pseudo-random number.
2. If $gcd(a, N) \neq 1$, then we are done.
3. Otherwise, use quantum subroutine to find the period r of $f(x) = a^x \bmod N$.
4. If r is odd or $\frac{ar}{2} \equiv -1 \bmod N$, go back to step 1.
5. The factors of N are $gcd(\frac{ar}{2} \pm 1, N)$.

Period Finding

Consider the modular multiplication function

$$f(x) = a^x \bmod N.$$

We want to find the smallest non-zero integer r such that

$$a^r \bmod N = 1.$$

To do so, we can represent the modular multiplication function with a unitary operator

$$U\left|y\right\rangle = \left|ay \bmod N\right\rangle.$$

We can then represent the eigenstate of U in the kth state as

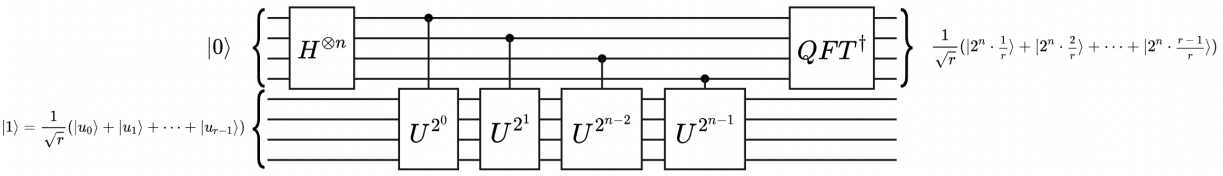
$$\left|u_1\right\rangle = \frac{1}{\sqrt{r}}\sum_{k=0}^{r-1}\exp-\frac{2\pi ik}{r}\left|a^k \bmod N\right\rangle.$$

In fact, we can further generalize this to any eigenstate u_s, where $0 \leq s \leq r-1$. By summing these eigenstates, we cancel out all computational basis states except for $|1\rangle$, meaning that $|1\rangle$ is a superposition of all these states. Then, if we estimate the phase, we can reduce the phase into the form

$$\phi = \frac{s}{r},$$

and have thus found the period. The convenience of these calculations arises from the fact that the period, r, must be in the eigenstates to ensure that the r computational basis states have equal phase differences. Thus, if we are able to to estimate the phase, we can extract r from the phase since $\phi = \frac{s}{r}$.

A high level overview of the quantum circuit[9] is shown below. The superposition of states is implemented by applying a Hadamard gate ($H^{\oplus n}$) to all qubits in the input register. Then the modular multiplication function f is implemented through repeated squaring as shown by gates $U^{2^1}, \ldots, U^{2^{n-1}}$. Finally, the input registers are passed through an inverse Quantum Fourier Transform before a measurement is performed.



Quantum Fourier Transform

The Quantum Fourier Transform plays an integral role in many popular quantum algorithms. In brief, the QFT takes a quantum state

$$\left|x\right\rangle = \sum_{j=0}^{N-1} x_j \left|j\right\rangle$$

as input and maps it to the quantum state

$$\left|y\right\rangle = \sum_{k=0}^{N-1} y_k \left|k\right\rangle.$$

The mapping is done according to the formula

$$y_k = \frac{1}{\sqrt{N}}\sum_{j=0}^{N-1} x_j \omega_N^{jk}$$

where $\omega_N^{jk} = e^{2\pi i \frac{jk}{N}}$. While Shor's uses the inverse of the transform, the only difference between the two is the reversal of all rotations, such as from π to $-\pi$.

Implications of Shor's Algorithm

Once a sufficiently powerful machine can implement Shor's, any schemes relying on the factoring attack being computationally infeasible will no longer be secure. One of the most famous of such schemes is the RSA encryption scheme. With approximately 2048 qubits, a 1024-bit key can be broken.

Shor's breaks similarly hard problems such as the Discrete Log Problem and Elliptic Curve Cryptography. This would leave many fields, such as blockchains, vulnerable to attack [6].

As Shor's breaks many public key cryptographic schemes, current research in post-quantum cryptography centers around finding post quantum secure public key algorithms.

4 Grover's Algorithm

We next discuss Grover's algorithm, developed by Lov Grover in 1996. Grover's algorithm is a quantum search algorithm that finds the unique input that gives a specific output i to a black box function with high probability in $O(\sqrt{N})$ function calls, where N is the input size. On the other hand, the classical computation would require $O(N)$, as on average we would have to check $N/2$ of the input and at worst all N inputs. As with most quantum algorithms, Grover's does so with high probability; Grover's algorithm is able to find the input with at least probability $1/2$. This does not change the runtime as the algorithm will only be needed to run twice on expectation. Furthermore, Grover's algorithm requires $\log_2 N$ qubits in its computation.

As a result, Grover's algorithm can be used not only for searching a database, but also for inverting a function; given a function $y = f(x)$, Grover's can calculate x when given an output y. Hence, for symmetric key cryptography, Grover's can be used in brute-force attacks to find the key.

Algorithm

Grover's algorithm is as follows [4]:

1. Initialize system to uniform superposition over all states $|s\rangle = \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} |x\rangle$, as all choices are equally likely.

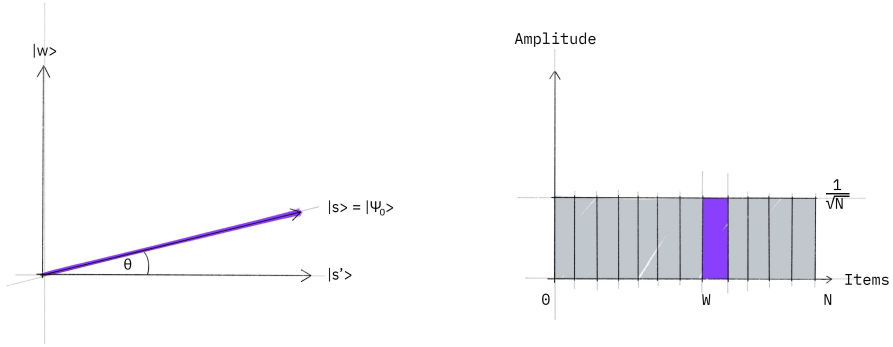
2. Perform amplitude amplification $O(\sqrt{n})$ times. This step amplifies the amplitude of the winner vector w, which is the unique input that we are trying to find, so that we are eventually able to pick out the winner from the group.

2a. Apply the reflection operator U_f.

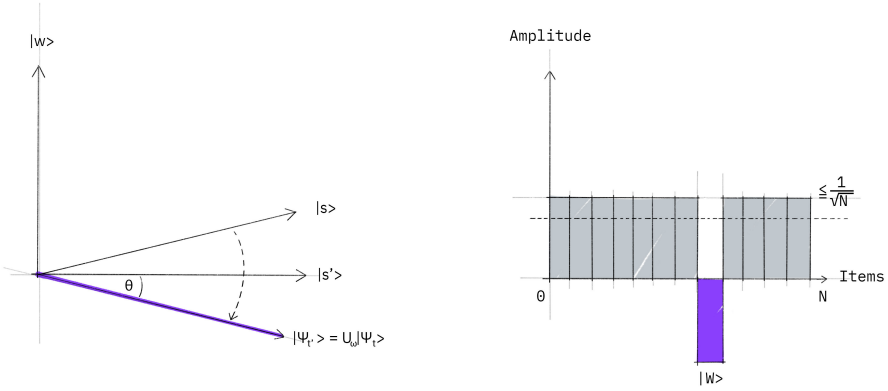
2b. Apply the Grover diffusion operator $U_s = 2|s\rangle\langle s| - I$.

3. Measure the quantum state for the answer.

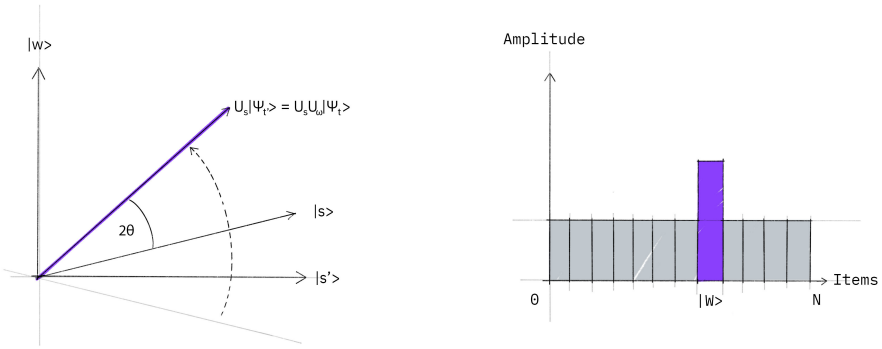
The following gives a visual representation of Grover's algorithm [4]. Here, w is the winner vector, s is the uniform superposition, and s' is the span of these two vectors. The leftmost diagrams depict the vectors and rightmost diagrams depict their amplitude. The first diagram is the system at uniform superposition.



The next diagram demonstrates step 2a, where the reflection operator U_f reflects s about s', making the amplitude of w negative.



The final diagram demonstrates step 2b, where the Grover operator reflects it about s, making the winner vector noticeable. Repeating this algorithm, we can measure the winner vector.



Implications of Grover's Algorithm

Grover's algorithm provides quadratic speedup from $O(n)$ to $O(\sqrt{n})$ time for function inversion problems. One consequence is that it allows for brute force attacks on symmetric encryption algorithms such as AES or SHA using a known plaintext attack, which is where given a known ciphertext and plaintext, we run over all the possible keys until the correct key is found. For instance, 128 bit keys could be broken in 2^{64} trials with 128 qubits and 256 bit keys could be broken in 2^{128} trials with 256 qubits. A simple prevention mechanism is to simply double key lengths, such as from 128 to 256 and 256 to 512, in order to prevent such attacks [3]. Hence, the implications of Grover's algorithms are not as far-reaching as Shor's and are ones that can be easily mitigated.

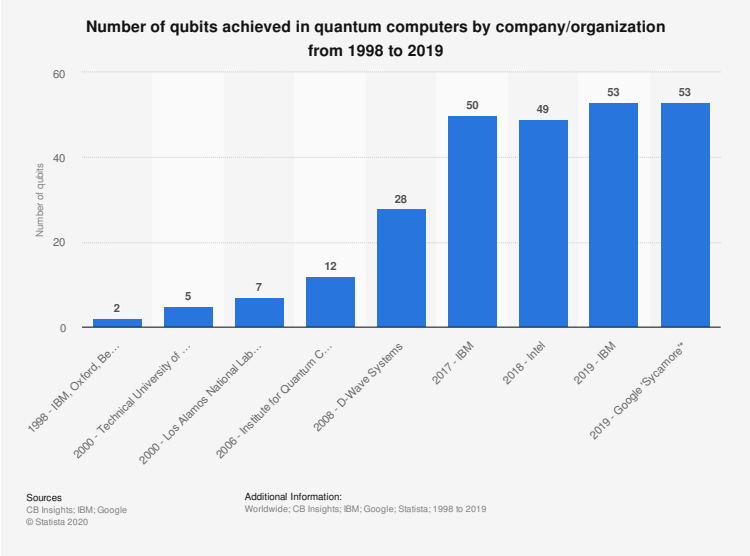
5 Future Developments

Because of the enormous power of quantum algorithms, it is important to estimate when quantum computing will become applicable in the real world. In addition, cryptographers are developing protocols that are secure against both quantum computers and classical computers.

Quantum Computing

The ultimate goal in quantum computing is to achieve a quantum advantage, performing a practical application like breaking cryptographic codes and simulating chemistry faster than a classical computer. Though quantum computers have been able to perform Gaussian boson sampling, a task that would take classical computers 2.5 billion years [5], experts predict that quantum computers being practically applicable is "likely still a long way off" [1]. Some compare compare the current stage of quantum computing to working with machine code, with software and programming languages yet to be developed.

As a heuristic for how the field has progressed, we look at number of qubits in quantum computers since the first quantum computer was built.



The graph above shows programmable quantum computer qubit records from 1998, where the first quantum computer was created at IBM with 2 qubits, to 2019. Over the past few years, there have been new records. In 2020, Chinese researchers developed Zuchongzhi 2 with 66 qubits. In 2021, IBM developed the "Eagle" chip with 127 qubits. They aim to create a 433-qubit quantum processor in 2022 and one with 1121 qubits by 2023. Google has its own plan to build a million-qubit quantum computer within the next 10 years. [2] As such, the rate of qubit usage has increased rapidly.

We also look at factoring records with quantum computers to see how large of a number quantum computers can handle.

- 2001: the number 15 was factored using 8 qubits
- 2012: the number 21 was factored using 10 qubits; later, 143 was factored using 4 qubits, by exploiting special structure in the number
- 2014: larger numbers like 56153 with a similar structure to 143 were factored
- 2020: at Q2B conference, announced that 1099551473989 factored by pre-compiling into 3 qubits

A number's magnitude does not determine its complexity in factoring, as some factoring algorithms take advantage of the special structure of certain numbers. In fact, in 2019, IBM attempted to factor 35 using

Shor's algorithm, but failed due to accumulating error. Thus, implementing algorithms for large input is still difficult.

A practical next step for building quantum computers is quantum error correction. To protect against decoherence, the premature observation of qubits due to environmental interference, researchers can use dozens to hundreds of actual qubits to encode a theoretical qubit state. Most major experiments so far do not use error correction, but this is starting to be implemented in the real world.

Post-Quantum Cryptography

Current approaches in post-quantum secure algorithms include lattice based algorithms, multivariate algorithms, code-based algorithms, and hash-based algorithms. Lattice based algorithms are short and fast, and some constructions build on top of the learning with errors (LWE) problem [3].

The LWE problem seeks to recover a secret from a set of approximate random linear equations on the secret [7]. Without noise, it would take polynomial time using Gaussian elimination to solve the system of equations, but introducing noise seems to make the problem exponentially more difficult. The current best-known solution to LWE is the Blum algorithm, and quantum computers do not seem to offer improvements. Hence, constructions based on the LWE problem could be secure in a post-quantum world.

Government organizations are also looking into post-quantum secure algorithms. In 2016, the National Institute of Standards and Technology (NIST) called for candidates for quantum-resistant public-key algorithms, in encryption, digital signatures, and key distribution [10]. These standards should be capable of protecting sensitive data in the future, even after quantum computing becomes practical. NIST released finalists in 2020, and aim to publish standards in 2024. As a result, in a few years, we hope to be better prepared for a post-quantum world.

6 Conclusion

Quantum computing has seen a surge of development in the past decade. Because quantum algorithms are more powerful than classical algorithms, the cryptographic landscape demands change in a post-quantum world. We discuss Shor's and Grover's algorithms as well as their implications on cryptographic schemes. While Grover's applications in function inversion affects symmetric algorithms, general mitigation techniques involve simply doubling the key length. On the other hand, Shor's integer factoring capability has devastating effects on algorithms such as RSA and elliptic curve-based schemes. Consequently, most post-quantum cryptography research centers on finding secure public key algorithms. As both fields are areas of active research, it is exciting to see the next innovations in quantum computing and cryptography.

References

[1] Scott Aaronson. *What makes quantum computing so hard to explain? - quanta magazine*. June 2021. URL: https://www.quantamagazine.org/why-is-quantum-computing-so-hard-to-explain-20210608/.

[2] Philip Ball. *First quantum computer to pack 100 qubits enters crowded race*. Nov. 2021. URL: https://www.nature.com/articles/d41586-021-03476-5.

[3] Daniel J. Bernstein and Tanja Lange. *Post quantum cryptography*. Sept. 2017. URL: https://www.nature.com/articles/nature23461.pdf.

[4] *Grover's algorithm*. Apr. 2022. URL: https://qiskit.org/textbook/ch-algorithms/grover.html.

[5] Hamish Johnston. *Gaussian Boson Sampling*. Dec. 2020. URL: https://physicsworld.com/a/quantum-advantage-demonstrated-using-gaussian-boson-sampling/.

[6] Joseph J. Kearney and Carlos A. Perez-Delgado. *Vulnerability of blockchain technologies to quantum attacks*. Apr. 2021. URL: https://www.sciencedirect.com/science/article/pii/S2590005621000138.

[7] Oded Regev. *Learning with Errors*. URL: https://cims.nyu.edu/~regev/papers/lwesurvey.pdf.

[8] *Shor's Algorithm*. 2015. URL: https://www.quantiki.org/wiki/shors-factoring-algorithm.

[9] *Shor's algorithm*. Apr. 2022. URL: https://qiskit.org/textbook/ch-algorithms/shor.html#3.-Qiskit-Implementation.

[10] Dawn Turner. *NIST Standards*. Feb. 2022. URL: https://www.cryptomathic.com/news-events/blog/understanding-nists-process-on-post-quantum-cryptography-pqc-standardization.