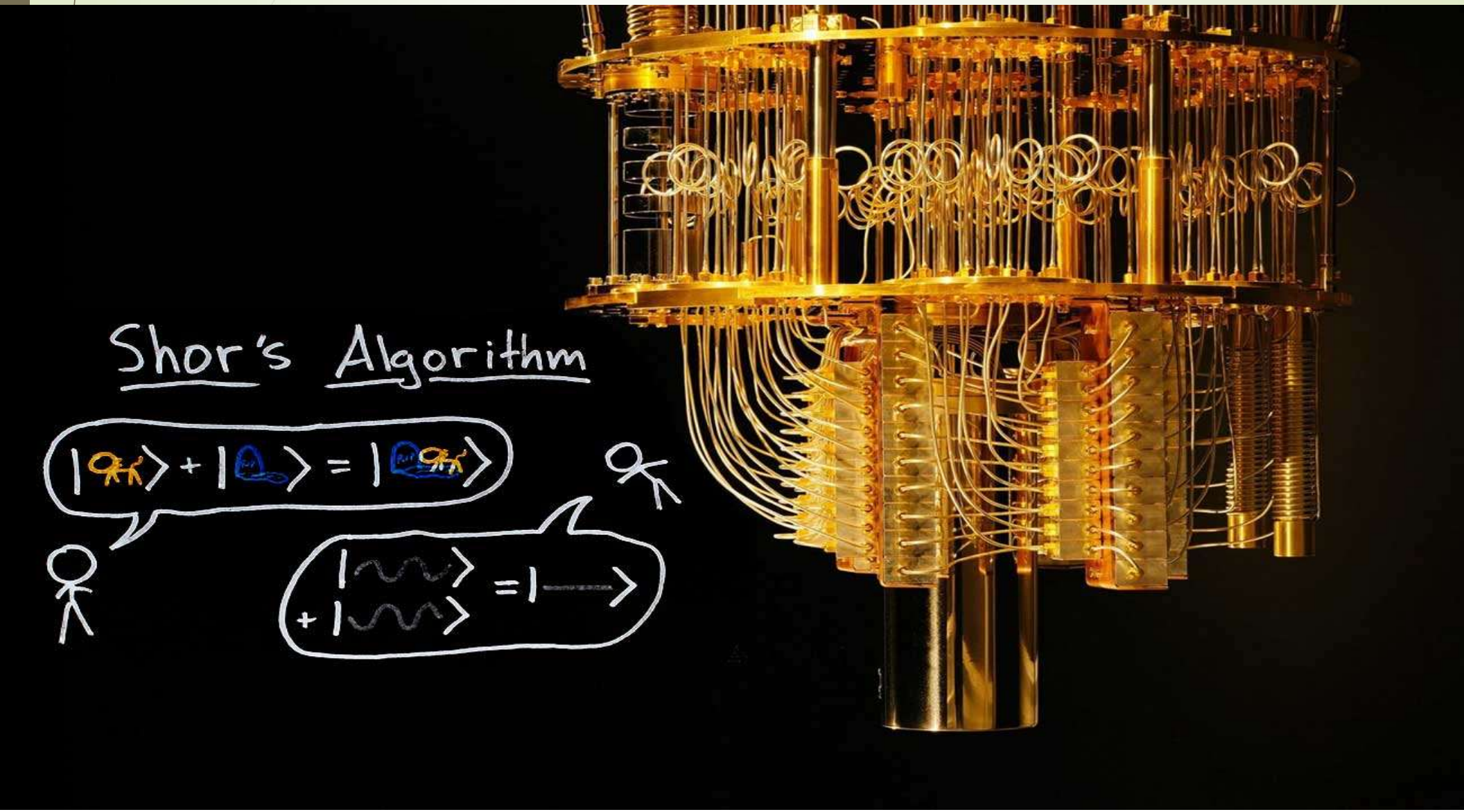
An Efficient Quantum Factoring Algorithm

By: M.Khavari



Abstract

- Shor's Algorithm (in English: Shor's Algorithm) is a quantum algorithm for decomposing numbers into prime factors in polynomial time. Named after Peter Sher, this algorithm was formulated in 1994.
- The algorithm turns random guessing into better guessing.
- Shore's algorithm accelerates the factorization algorithm by using the quantum Fourier transform and it is shown that the factorization operation is performed in polynomial time.
- Important The most common application of this algorithm is breaking the RSA public key cipher system and ciphers based on bending curves,It is an ellipse that depends on the discrete logarithm problem.

- This algorithm has the potential to break many internet security systems that rely on the difficulty of factoring large numbers when run on a quantum computer.
- Soonwon Choi, a physicist at the Massachusetts Institute of Technology, says:"We don't want a quantum computer to just do one specific task. We need to see what else we can do with a quantum computer besides Shor's algorithm."
- In the past, researchers thought many times that they had achieved a quantum advantage and discovered a quantum algorithm that could be solved faster on a quantum computer than a classical computer. But after some time, other researchers, including Ewin Tang, presented intelligent classical algorithms that could outperform quantum algorithms and question the advantage of quantum.

RSA encryption algorithm

- It is asymmetric type or public key method.
- This encryption method is used when the user who encrypted the data does not have a safe and secure way to share the key with the second user.
- Another feature of RSA encryption is the use of the operation of multiplying two prime numbers (N = a.b), the result of which (N) is the encrypted data.

- Since calculating the relationship between these numbers is easy on one hand and extremely difficult on the other, it's called a trapdoor function. In RSA encryption, very large numbers are used to make guessing these two factors even harder.
- The difficulty of breaking RSA encryption comes down to two main points:

  1- The numbers used in this algorithm are so large that finding the private key through trial and error isn't feasible.

  2- Current computers would need millions or even billions of years of calculations to find the prime factors of these huge numbers.

How the Shor algorithm works in simple terms

$$gcd(N, m) \neq 1$$

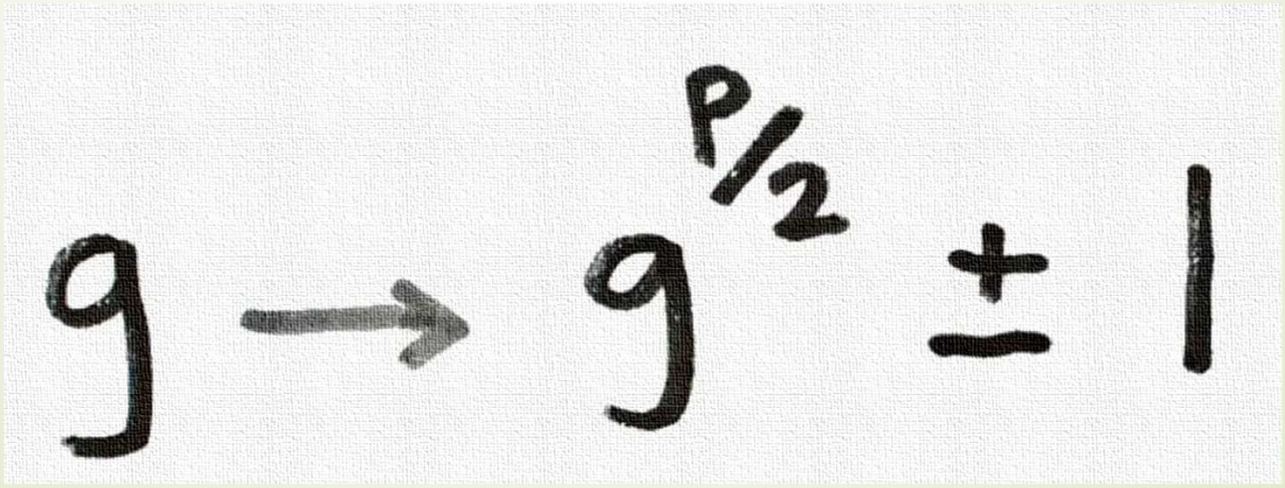
$$f_N : N \longmapsto N$$
$$a \longmapsto m^a mod N$$

$$(m^{(p/2)} - 1)(m^{(p/2)} + 1) = (m^p - 1) = 0 mod N$$

$$(m^{(p/2)} + 1) = 0 mod N$$

- So, in Shore's algorithm, we start with a random numerical guess, which may be a multiple of N (but not likely), and then this algorithm turns this random guess into a much better guess, which is probably a multiple of N.

- According to this algorithm, any random and baseless guess (g) of a number that could be a multiple of N, when raised to the power of p/2 and adjusted by adding or subtracting one, gets much closer to the correct answer.
- Everything starts with the big number N, which we need to find its multiples in order to break the encrypted information lock.
- The important thing to note is that the number we guess doesn't necessarily have to be a multiple of the prime factor N.



- This algorithm uses a trick that turns an unlikely guess into one that has a high chance of sharing a common multiple with N.

$$g^p = m \cdot N + 1$$

- So for a large number N and an unlikely guess g, we can be sure that a power of g equals a multiple of N plus one.

Example

$$7^2 = 3 \cdot 15 + 4$$

$$7^3 = 22 \cdot 15 + 13$$

$$7^4 = 160 \cdot 15 + 1$$

- By cleverly rearranging this equation, it can be rewritten like this:

$$g^p - 1 = m \cdot N$$

$$\left(g^{\frac{p}{2}} + 1\right) \cdot \left(g^{\frac{p}{2}} - 1\right) = m \cdot N$$

Example

$$7^{\frac{4}{2}} + 1 = 50$$

$$7^{\frac{4}{2}} - 1 = 48$$

- Unlike classical computations that provide only one answer for each input, quantum computations can simultaneously calculate multiple possible answers for each input thanks to the phenomenon of superposition; however, in the end, only one answer is calculated, and it's entirely random.
- The key to fast quantum computations lies in achieving a type of quantum superposition that calculates all possible answers at once while intelligently placing all incorrect answers in a destructive interference state to neutralize each other. In this case, when the answer to the problem is computed, the output is no longer random, and it's highly likely to be the correct answer.

$$g^{x} = m \cdot N + r$$

$$\Downarrow$$

$$g^{x+p} = m_2 \cdot N + r$$

Example

$$g^{42} = m_1 \cdot N + 3$$

$$g^{42+p} = m_2 \cdot N + 3$$

$$g^{42+2p} = m_3 \cdot N + 3$$

- The best tool for finding frequency is the Fourier transform, which converts audio signals into a wave and presents it as a graph of sine and cosine functions to show the different frequencies that make up the wave.
- A regular computer would need to go through these steps separately for each power of p, which means it would take years to correctly calculate p and ultimately find the multiples of N for very large values of N. However, a quantum computer can compute all these possible values at once, putting all the wrong states into a destructive interference pattern, so the final answer is the correct value of p.
- In this way, Shor's algorithm, by correctly guessing the multiples of N, essentially renders current encryption methods based on the product of two prime numbers useless.

- The main challenge is the extremely long time it takes for a regular computer to turn our random, unlikely guess into a better guess that is likely to be the right answer.
- Quantum computer They only increase the speed of getting an answer; in fact, it's a remarkable increase, reducing the time to solve a problem from millions of years to just a few minutes.
- The report showed that a hypothetical quantum computer could easily solve an unsolvable math problem without needing to perform separate calculations for different values.

Applications of Shor's algorithm in cryptography

- All cryptographic algorithms that are related to the factorization problem can be broken by Shore's algorithm.
  a. RSA Problem
  b. b. Rabin Problem
  c. c. quadratic residuosity Problem
  d. d. The square root modulo n problem (SQROOT)

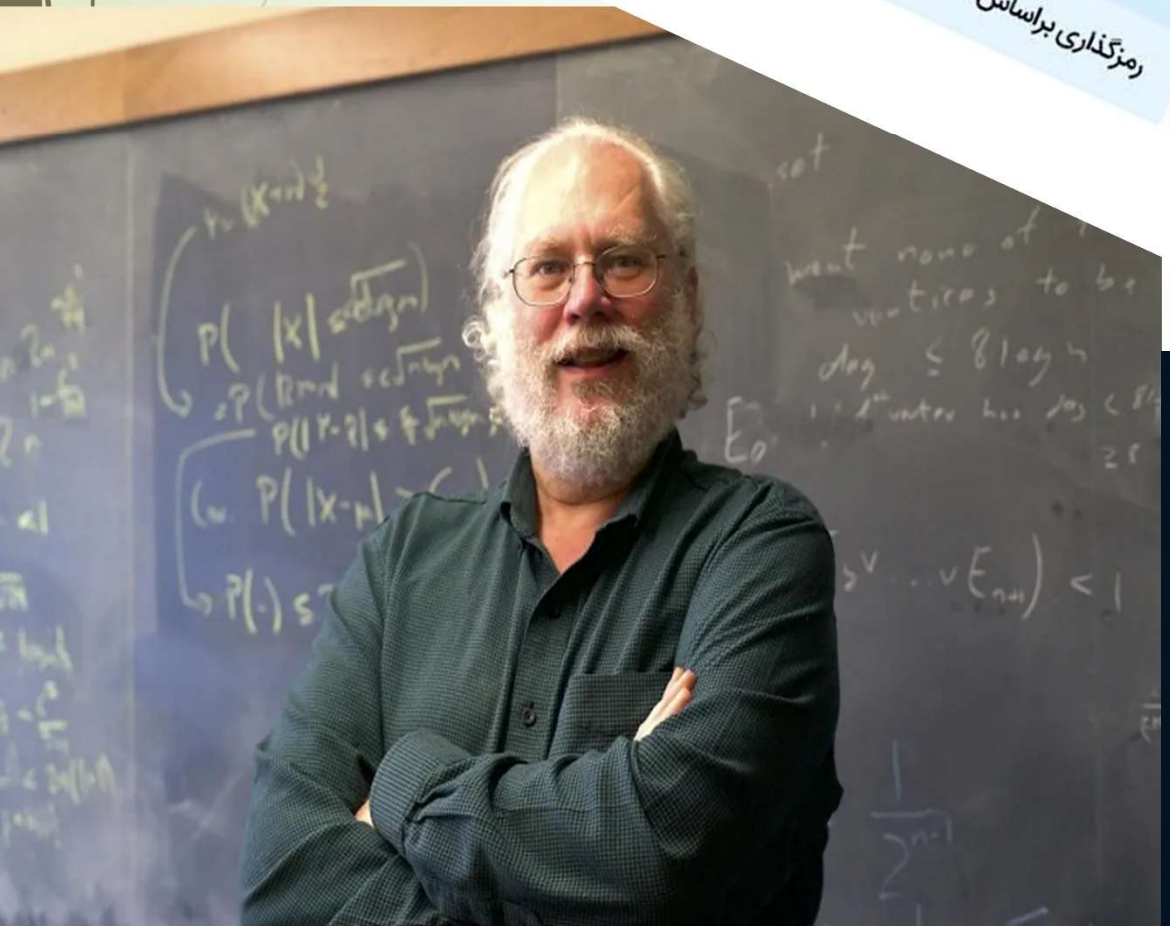
- All algorithms that somehow depend on the discrete logarithm problem will break under Shore's algorithm, which includes the following
  a. Diffie-Hellman key agreement and its derivatives
  b. b. ElGamal encryption, and the ElGamal signature scheme and its variant
  c. c. ECC
- One of the most important challenges of this algorithm is the existence of a quantum computer in the implementation of the quantum algorithm

To get a better understanding of how the Schur algorithm works, you can watch the video below. In this video, this algorithm is used to find two multiples of the number 314191.

https://www.aparat.com/v/l4110yg

رمزنگاری کلاسیک در مقایسه با رمزنگاری کوانتومی

رمزنگاری کلاسیک	رمزنگاری کوانتومی
از منطق مبتنی بر دیجیتال استفاده می‌کند	مبتنی بر نظریه کوانتومی است
ارسال سینگال های دیجیتال با استفاده از بیت ها	ارسال داده ها با استفاده از فوتون ها
معمولا محدوده مرتبطی ندارد	به طور معمول دارای یک محدوده مرتبط با آن است
رمزگذاری براساس الگوریتم های ریاضی است	که نیاز به سیم فیبرنوری و تکرار کننده دارد
	رمزگذاری براساس ویژگی های کوانتومی است



مزایای استفاده از رمزنگاری کوانتومی

- امنیت بی‌سابقه
- حفاظت در برابر حملات کوانتومی
- دسترسی کاربران به اطلاعات
- ردیابی تغییرات
- پویایی و زمان‌بندی

با تشکر از استاد گرامی

جناب آقای دکتر رجایی

شهریور 1403