The Effect of Quantum Computing on Cryptography

Eric Kharitonov Following 14 min read · 3 days ago

Want the complete mathematical treatment? [Download the full PDF here]

What is a Quantum Computer?

Many have heard the phrase **Quantum Computer**, but what is it? To thoroughly answer this very ambiguous and yet foundational question, you would probably need a Ph.D in physics or mathematics, but let's try to understand the core ideas in a few pages. To do this, we will first quickly cover the very basics of classical computers.

I would also like to clarify that to fully understand the following article, it is recommended to have a basic foundation in linear algebra (vector spaces, linear maps, matrices, eigenvalues and eigenvectors), number theory, and group theory. That said, for the clarity of readers without the aforementioned mathematical background, I will add an intuitive explanation of each topic before the mathematical rigor.

Classical Computing

A classical computer stores and manipulates all its information in **binary**, a language made up of 0s and 1s. Each unit is called a bit. Generally, a bit can be thought of as anything that can exist in one of two states (like the two sides of a coin), but for our purposes, it's simply 0 or 1. In classical computing, the bit is represented by the state of an electrical transistor, 0 if its off, 1 if its on. Every time you type something or take any action on your computer, that information is broken down and stored in a series of bits.

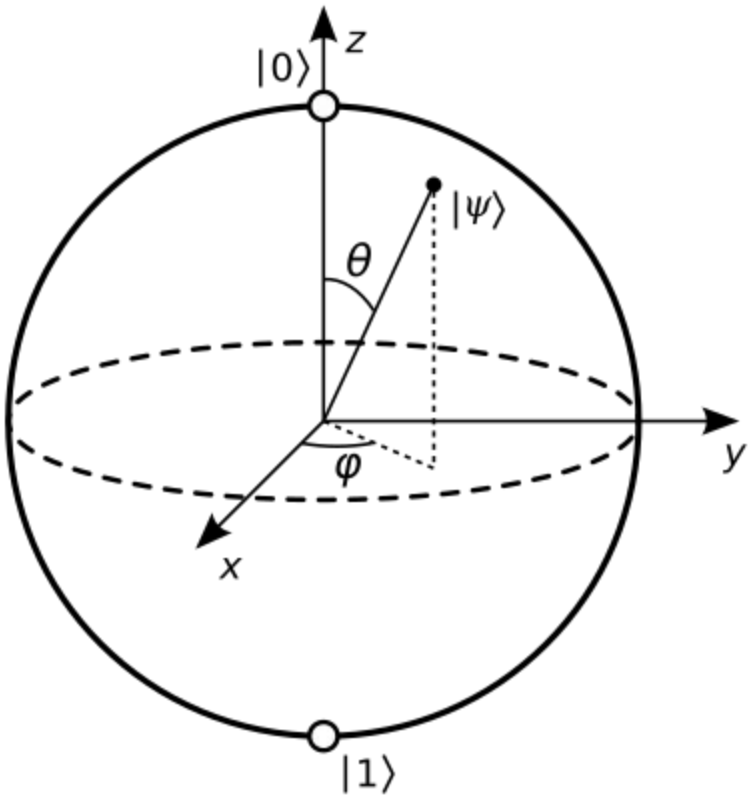
For example, if you typed the character "3", the computer would not actually "see" the number three. Instead, it would store the binary code 00110011, which is how the character "3" is represented inside the computer.

A classical computer processes all its information (binary strings) using logical operations like the NOT operation that simply reverses the state of its input bit. These operations are called **gates**.

What is a qubit?

Now, let's tackle the previously stated question: What is a quantum computer? While a classical computer uses bits to store and manipulate information, a quantum computer would use **qubits**, or quantum bits.

Qubits are the basic units of information used in quantum computation. While a bit can represent any physical system that can be in one of two states, a qubit can represent a physical system that can be in the 0 state, 1 state, or any state in between. Intuitively, you can visualize the state of a qubit as a point on a unit sphere, known as the **Bloch Sphere**, where the north pole can represent the $|0\rangle$ state and the south pole can represent the $|1\rangle$ state (The weird symbols around 0 and 1 will be explained in the next section).



Bloch Sphere

More rigorously, qubits are abstract representations of physical systems coupled with a 2-dimensional complex vector,

$$v = \begin{bmatrix} \alpha \\ \beta \end{bmatrix}$$

known as the state vector, where the state of the qubit can be written in the form:

$$\alpha\,|0\rangle + \beta\,|1\rangle$$

subject to the **normalization condition** given by

$$|\alpha|^2 + |\beta|^2 = 1$$

with $\alpha, \beta \in \mathbb{C}$. The complex numbers α and β are called **probability amplitudes,** and they essentially act as coordinates that pinpoint the exact location of the state on the Bloch Sphere.

Computational Basis

Before exploring more general states, it is useful to introduce the concept of a computational basis. The **computational basis** is the list of orthogonal (perpendicular) basis vectors that represent the classical states of a quantum

system. To be more precise, the computational basis of a quantum system is the list composed of the eigenvectors of the **Pauli-Z operator**.

For a single qubit, the computational basis consists of the two state vectors:

$$|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad |1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

Any valid state of a qubit can be expressed as a linear combination of these two states. This fundamental principle, where a quantum system can exist in a "blend" of the basis state,s is called a **superposition.**

For an n-qubit system, the computational basis is the list of all possible bit-strings of length n. For example, a 2-qubit system has the computational basis:

$$|00\rangle = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad |01\rangle = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}, \quad |10\rangle = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}, \quad |11\rangle = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

In general, any orthogonal basis of unit vectors can be the basis states of a quantum system.

Superposition and Multi-Qubit Systems

A quantum state of an n-qubit system is represented by a 2^n dimensional complex vector:

$$v = \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_{2^n} \end{bmatrix}, \quad \text{such that } \sum_{i=1}^{2^n} |\alpha_i|^2 = 1$$

For example, a two-qubit system has four computational basis states: $|00\rangle$, $|01\rangle$, $|10\rangle$, $|11\rangle$, and a general two-qubit state takes the form:

$$\alpha_1 |00\rangle + \alpha_2 |01\rangle + \alpha_3 |10\rangle + \alpha_4 |11\rangle \text{ where } \sum_{i=1}^{4} |\alpha_i|^2 = 1$$

The extraordinary strength behind a quantum computer lies in the exponential growth mentioned and the ability to apply operators on multiple states at once, and it will be further elaborated on in the next section and in later sections, we will use this fact to explain how quantum computers can break modern encryption.

Measuring a Qubit

What does it mean to measure a qubit? Measuring a qubit is the act of extracting classical information from the qubit.

Intuitively, measuring a qubit collapses the state of the qubit to the computational basis states with some probability. That is, if we try measuring a qubit with state $\alpha|0\rangle + \beta|1\rangle$, we will get the state $|0\rangle$ with probability $|\alpha|^2$ and the state $|1\rangle$ with probability $|\beta|^2$. That is why we have the normalization constraint, so the probabilities add to 1.

Think of Schrödinger's famous cat. Before opening the box, the cat exists in a superposition of being both alive and dead simultaneously. The act of opening the box to check (measurement) forces the cat to collapse into either definitely alive or definitely dead, with probabilities determined by the original superposition state.

Quantum Gates

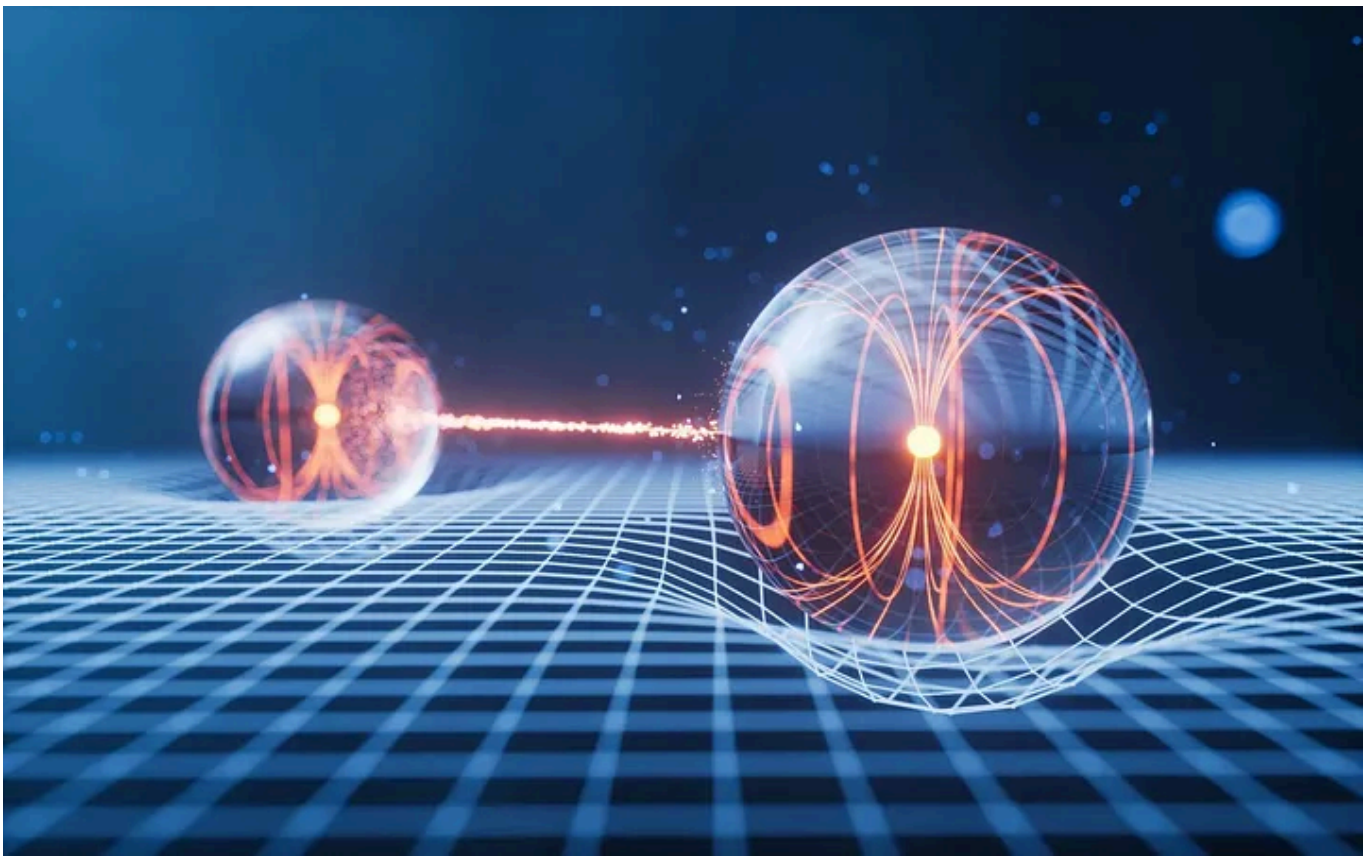
Earlier in the section, I introduced the concept of classical gates, which act as operations on the bits in a classical system. Unsurprisingly, there is an analogous set of objects in quantum computing called quantum gates. Quantum gates are operations that, unlike classical gates, act on qubits, changing the state of the qubit to another quantum state. More rigorously, quantum gates can be expressed as unitary matrices. A common quantum gate is the Hadamard gate, given by:

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

Notice that applying the Hadamard gate to a computational basis results in a superposition of the basis vectors.

Quantum gates have many uses in quantum algorithms and quantum error correction, making them a very useful tool.

Entanglement



> *” *Entanglement is not one but rather the characteristic trait of quantum mechanics.”** — Schrödinger

right out of a sci-fi movie, entanglement is almost always the culprit. So what is entanglement?

Imagine you have two qubits, one is with you in your room, while the other is somewhere light-years away in a distant corner of the universe. Let these qubits be entangled, meaning their states are linked. If the state of one qubit collapses to a particular basis state upon measurement, the state of the other qubit is instantly determined with absolute certainty. No matter how far apart they are, as long as they remain entangled, they are no longer independent.

If this seems strange, unintuitive, or even spooky, don't worry, Einstein thought the same thing.

> *" *I cannot seriously believe in it [quantum theory] because the theory cannot be reconciled with the idea that physics should represent a reality in time and space, free from spooky actions at a distance."** — Albert Einstein

To better understand entanglement, consider a two-qubit system in the state:

$$|\psi\rangle = (1/\sqrt{2})|00\rangle + 0|01\rangle + 0|10\rangle + (1/\sqrt{2})|11\rangle$$

That is

$$|\psi\rangle = (1/\sqrt{2})(|0\rangle \otimes |0\rangle + |1\rangle \otimes |1\rangle)$$
$$= (1/\sqrt{2})(|0\rangle|0\rangle + |1\rangle|1\rangle)$$

Notice that if we measure the system, the first qubit will collapse to $|0\rangle$ with probability 50% or to $|1\rangle$ with probability 50%. If the first qubit collapses to $|0\rangle$, then the second qubit will also collapse to $|0\rangle$ with 100% certainty. Likewise, if the first qubit collapses to $|1\rangle$, the second will collapse to $|1\rangle$. In general, entangled quantum states are the states that cannot be decomposed into their basic components. Using our previous example once again, you can see

$$|\psi\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) \neq (\alpha_1 |0\rangle + \beta_1 |1\rangle) \otimes (\alpha_2 |0\rangle + \beta_2 |1\rangle)$$

for any $\alpha_1, \alpha_2, \beta_1, \beta_2 \in \mathbb{C}$

Modern Cryptography

Cryptography is present everywhere in our lives, from sending messages to keeping your digital information safe, but how does it work? There are two

forms of cryptography, symmetrical and asymmetrical encryption. Symmetrical encryption is often faster, cheaper, and better for storing large amounts of data, while asymmetrical encryption is more secure and better for smaller sets of data, such as emails and messages. We will not invest too much time in symmetrical cryptography, but it is useful to introduce it.

Symmetrical encryption

Symmetric encryption uses a single secret key to encrypt and decrypt data. The challenge is: how do two parties agree on a secret key if someone is listening?

The Diffie-Hellman Key Exchange

Imagine Alice and Bob want to communicate privately, but Eve is eavesdropping. They can establish a shared secret key using the Diffie-Hellman protocol:

Step 1: Publicly, Alice and Bob choose some prime number p and some integer g.

Step 2: Privately, Alice chooses a secret number ‘a’ that only Alice knows, and Bob chooses a secret key ‘b’ that only Bob knows.

Step 3: Then Alice sends Bob the number g^a (mod p) and Bob sends to Alice the number g^b (mod p). Notice that Eve sees every transaction between Alice and Bob.

Step 4: Finally, Alice calculates the number $(g^b)^a$ (mod p) and Bob calculates the number $(g^a)^b$ (mod p). Notice that:

$$g^{ba} \equiv g^{ab} \pmod{p}$$

Step 5: We let $(g^a)^b$ (mod p) be the secret key.

So what does Eve know by the end of this? Eve only knows what g and p are, as well as the numbers g^a (mod p) and g^b (mod p). That said, Eve does not know 'a' or 'b', so by choosing large enough numbers p and g, we can make it a very difficult task for a classical computer to discover the secret key.

RSA Asymmetrical Encryption

Unlike symmetrical encryption, asymmetrical encryption has two keys per person, not just one.

The Idea behind the RSA encryption scheme is that if Alice wants to send Bob a secret message, Alice would encrypt the message using Bob's public key, send it to Bob over a public channel, and Bob would decrypt the message using his own private key. This way, even though Eve knows the encrypted message, it would take a very long time, too long, to decrypt the message, as Eve does not know Bob's private key. So how do we come up with these public and private keys such that Eve can view every step but learn nothing? We use primes.

Let p and q be prime numbers and let n = pq. Next, consider the number $\phi(n)$ where $\phi: \mathbb{N} \to \mathbb{N}$ is given by:

$$\phi(g) = |\{a \in [g] : \gcd(a, g) = 1\}|$$

For any $g \in \mathbb{N}$. We call this function the Euler Totient Function, and by the multiplicative property of Euler's Totient Function, we know $\phi(n) = \phi(p)\phi(q) = (p-1)(q-1)$, as all positive integers preceding a prime are coprime to said prime. Next, choose an integer e such that $1 \leq e \leq \phi(n)$ and $\gcd(e,\phi(n)) = 1$. We will let (n, e) be our public key.

Next we will pick any number d such that $(e * d) \equiv 1 \pmod{\phi(n)}$. We will set our private key to be (n, d).

Finally, we can encrypt any message by first converting the message to a number M, then taking $M^e \pmod n$, then send the product to Bob, who will decrypt the number by finding $(M^e \pmod n)^d \pmod n$, which equals M, giving us the original message.

— -

How Quantum Computing Breaks Encryption

Now that a strong foundation of encryption and quantum computation has been developed, we can tackle the main point of the article. There are two famous quantum algorithms that can be used to break encryption: Grover's algorithm and Shor's algorithm. For the purposes of this topic, Grover's algorithm is less important as it does not have such a drastic effect as Shor's algorithm on cryptography. Grover's algorithm can be prevented from affecting encryption by simply increasing the size of our keys.

Shor's Algorithm

Shor's algorithm is a five step process such that, given any number N where $2 \nmid N$ and N is not a power of a prime, it will factor N into two integers p and q, where if p or q are not prime, we can apply the same algorithm or, if p or q are even or prime multiples we can apply classical algorithms to them until we attain the prime factorization of N.

The idea behind Shor's algorithm is first to take the problem of factoring a large number to finding the period of a certain sequence, and then using the period to find the factors. More specifically, given an odd positive integer N where N is not a prime power, we can factor N by following these steps:

1. Pick a $\in$ [2, N−1].

2. Find g = gcd(a, N). If g ≠ 1 then set p = gcd(a, N) and q = N/gcd(a, N) and we are done. If a and N are coprime, move to step 3.

3. Since a and N are coprime, by Bézout's lemma, we have:

$$ax + Ny = 1 \quad \text{for some } x, y \in \mathbb{Z}$$

Thus:

$$ax \equiv 1 \pmod{N}$$

So we have a $\in$ $(\mathbb{Z}/N\mathbb{Z})^x$, therefore by Lagrange's theorem, there exists a minimal number r such that $a^r \equiv 1 \pmod{N}$ and:

$$r \mid |(\mathbb{Z}/N\mathbb{Z})^\times| = \phi(N)$$

We call r the multiplicative order of a.

4. Use the quantum subroutine to find r. If r is odd, go back to step 1 and choose a different a.

5. Notice that since $a^r \equiv 1 \pmod N$ and $2 \mid r$, we have:

$$N \mid a^r - 1 \Rightarrow N \mid (a^{r/2} - 1)(a^{r/2} + 1)$$

Additionally, $N \nmid a^{r/2} - 1$ since that would imply:

$$a^{r/2} \equiv 1 \pmod N$$

contradicting the minimality of r.

Since N divides $(a^{r/2} - 1)(a^{r/2} + 1)$ and N does not divide $a^{r/2} - 1$, we must get either $\gcd(N, a^{r/2} + 1) = N$ or $1 < \gcd(N, a^{r/2} + 1) < N$.

Case 1: If $\gcd(N, a^{r/2} + 1) = N$ then $a^{r/2} + 1 \equiv 0 \pmod N \Rightarrow a^{r/2} \equiv -1 \pmod N$, which implies $\gcd(N, a^{r/2} - 1) = \gcd(N, -2) = \gcd(N, 2) = 1$. Therefore, we cannot derive non-trivial factors of N from a, forcing us to go back to step 1 and pick a new a.

Case 2: If $1 < \gcd(N, a^{r/2} + 1) < N$, then we can take the non-trivial factors $\gcd(N, a^{r/2} + 1)$ and $N/\gcd(N, a^{r/2} + 1)$, completing the algorithm.

The Quantum Subroutine

The problem occurs when we try to find the period r. Given a large enough integer N, finding r becomes a very difficult task that can be considered computationally impossible with a classical computer. But for a quantum computer, this becomes a much simpler task. Let's rephrase the task to the following.

Define a function f: $\mathbb{Z}^+ \to \mathbb{Z}/N\mathbb{Z}$ such that $f(x) = a^x \pmod N$. Then by definition of r:

$$f(x+r) = f(x)$$

Thus, our goal is to find r.

To do this, we consider a quantum circuit with two registers, the input register and the output register. The input register has 2n qubits, where n is the smallest integer such that $N \leq 2^n$. We take 2n qubits as it offers enough precision for the QFT to find r. We then let the second register hold n qubits. We initialize the system in the state $|0\rangle^{\otimes 2n} \otimes |0\rangle^{\otimes n}$. We then apply the Hadamard gate to the input register to get the superposition:

$$\frac{1}{\sqrt{2^{2n}}} \sum_{x=0}^{2^{2n}-1} |x\rangle \otimes |0\rangle^{\otimes n}$$

where $|x\rangle$ is the unique computational basis associated with the binary representation of the integer x. So every possible exponent x has equal

probability of appearing once measured (including r). Next, consider the unitary operator U^f: $|x\rangle|0\rangle \rightarrow |x\rangle|f(x)\rangle$. Applying the operator U^f on the entire system produces an entangled superposition

$$\frac{1}{\sqrt{2^{2n}}} \sum_{x=0}^{2^{2n}-1} |x\rangle \otimes |f(x)\rangle = \frac{1}{\sqrt{2^{2n}}} \sum_{x=0}^{2^{2n}-1} |x\rangle\,|f(x)\rangle$$

Notice that this state is entangled as the output register will always collapse to state f(x) as long as the state $|x\rangle$ has been measured or vice versa.

Next, we measure the output register, getting some state $f(x_0)$. But since these registers are entangled, the state of the input register collapses to:

$$\frac{1}{\sqrt{K}} \sum_{k=0}^{K-1} |x_0 + kr\rangle$$

Where K is the number of values of the form $x_0 + kr < 2^{2n}$, and we have $|x_0 + kr\rangle$ since f is periodic with r as the order. So now our input register is in a superposition of all the exponent values x such that $f(x) = f(x_0)$.

Finally, we can begin to extract the period r. We do this by applying the **Quantum Fourier Transform** to the superposition above, making it so that the probability amplitudes are highest at multiples of $2^{2n}/r$. So measuring the

state provides a value $z \approx s * 2^{2n}/r$ where $0 \leq s < r$ with high probability. That is:

$$\frac{z}{2^{2n}} \approx \frac{s}{r}$$

The continued fraction expansion of $z/2^{2n}$ produces a sequence of rational approximations p_i/q_i in lowest terms, called convergents. A key theorem guarantees that if:

$$\left| \frac{s}{r} - \frac{z}{2^{2n}} \right| < \frac{1}{2r^2}$$

Then the convergents will include the fraction s/r reduced to lowest terms.

For each convergent p_i/q_i, we test the denominator q_i by checking whether a^(q_i) ≡ 1 (mod N). When this condition is satisfied, q_i is either equal to the period r or is a divisor of r. In either case, if q_i is even, we can attempt to factor N using gcd(a^(q_i/2) ± 1, N) as described in Step 5. If this attempt fails to produce a nontrivial factor, we simply repeat the quantum subroutine with a new measurement.

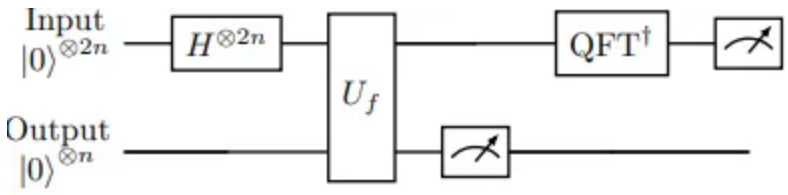


Figure 3: Quantum circuit for Shor's algorithm.

Quantum circuit for Shor's algorithm

Final Comment On Future Of Encryption

With the threat of Shor's algorithm breaking asymmetrical encryption, you may be sent into panic mode, thinking that all our data is no longer safe and society is in danger. Thankfully, though, this is not at all true. With quantum computers will come quantum safe encryption that encrypts data using methods that would perplex even the most powerful quantum computers that will only arise decades into the future. One such encryption method requires finding a path between two points in a multi-dimensional grid with limited directional freedom.

So while quantum computers may change the rules of computation, they won't spell the end of digital security, but will result in new and exciting innovations.

About the Author: Eric Kharitonov is a 16-year-old passionate about quantum computing, math, and cryptography. Connect on [LinkedIn/Instagram/Slack]

If you enjoyed this article, please share it with others interested in quantum computing!

Quantum Computing | Cryptography | Encryption | Post Quantum Cryptography | Quantum Physics

Written by Eric Kharitonov

14 followers · 18 following

Following

Responses (4)

Alex Mylnikov

What are your thoughts?

Abdullah Javeed he
23 hours ago

GooooooD 🫶

8 1 reply Reply

Rad Sarar he
4 hours ago

Hey, I got your article so fascinating! Really plausible one it was! Can you share me your Email id? I can personally talk to you about this work. Reddit, email whatsoever.

1 reply Reply

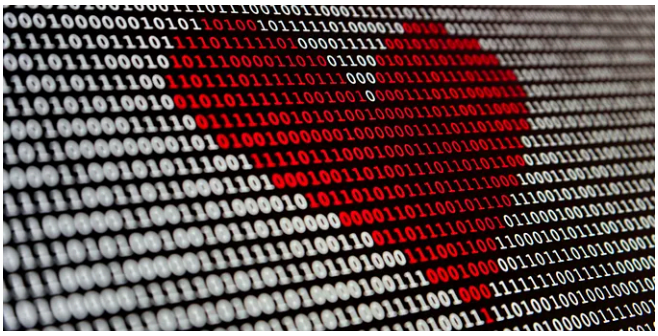
Dylan Rosario
7 hours ago

Shor will not work against eni6ma.com
Rosario Proof . Post quantum geometry.

Reply

See all responses

Recommended from Medium



In Bootcamp by Alessandro Romano

The Lost Beauty of Code

When code stops being written by humans, does beauty still matter?

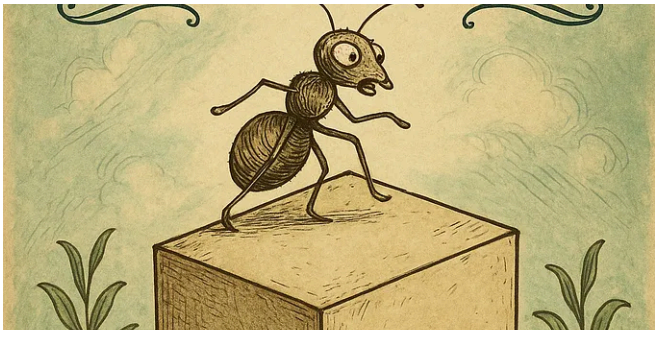
4d ago 210 10

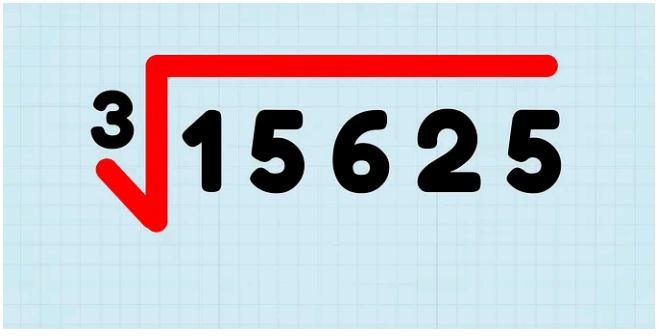
Karl Wiegers

9 Books That Influenced My Thinking

A book needs to yield only a few actionable ideas to be worth reading. These nine gave...

3d ago 33





In Math Games by BL

Can You Solve The Drunk Ant: A Challenging Quant Interview...

Let's imagine a tiny ant standing on one corner of a shiny, metallic cube. The cube is...

2d ago 20 1

In ThinkArt by Nnamdi Samuel

Never Use a Calculator for Cube Roots Again

Find ANY Cube Root In Less Than 4 Seconds!

6d ago 424 6

In The Medium Blog by Medium Staff

It happened on Medium: September 2025 roundup

The best reads this month our readers loved that you might have missed.

3d ago 2.3K 55

Bogdan Ilyin

Denmark Just Triggered Putin's Worst Nightmare

Europe's quietest country just made one of the loudest moves against Moscow's war...

Oct 6 15.4K 197

See more recommendations