Rest A-Shor-ed: Quantum Computing Revolutionizes Cryptographic Security

Samuel Caruthers, Ljosha Kremliovsky, and Daric Zhou, mentored by Kyle Hansen

University of California, Santa Barbara

Abstract

Shor's Algorithm is a quantum algorithm used to efficiently factor large numbers. We investigate quantum computing and examine the idea of leveraging this algorithm as a quantum attack against the classical RSA cryptosystem.

Cryptography and RSA

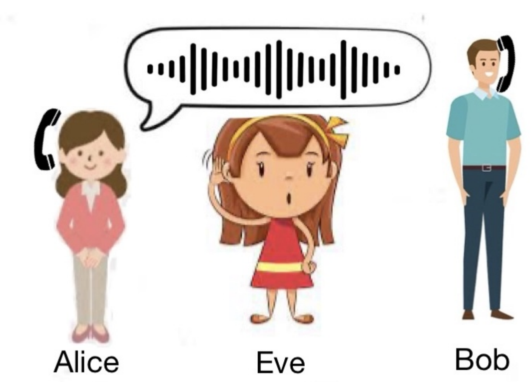


Fig. 1: Our main characters

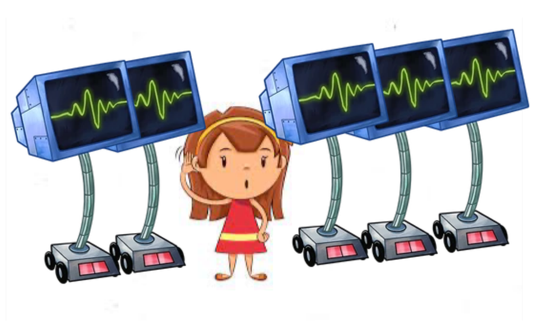


Fig. 2: Eve with her quantum computers

Euler's φ Function

Let $m \in \mathbb{N}$. The value $\varphi(m)$ is

$$\#\{k \in \mathbb{N} \mid \gcd(k, m) = 1,\ k \leq m\}$$

If $m = pq$ for p, q prime, then

$$\varphi(m) = (p-1)(q-1).$$

Euler's Theorem

For $x \in \mathbb{Z}$ with $gcd(x, n) = 1$, we have

$$x^{\varphi(n)} \equiv 1 \pmod{n}.$$

The RSA Algorithm

Bob's Private Knowledge

$p = 17$ $\quad q = 5$

$n = pq = 85$ $\quad \varphi(n) = (p-1)(q-1) = 64$

$e = 7$ $\quad d \equiv e^{-1} \pmod{\varphi(n)} = 55$

Bob's Public Key

$n = 85$ $\quad e = 7$

Bob Checks

p and q are prime

$\gcd(e, \varphi(n)) = 1$

Alice Computes

She encodes her message m as the number $m = 11$.

Computes $c \equiv m^e \pmod{n}$ or

$71 \equiv 11^7 \pmod{85}$

and sends c to Bob

Bob Decrypts

He computes $c^d \pmod{n}$ and gets

$$71^{55} \equiv 11 \pmod{85}$$

because, by Fermat's Little Theorem,

$$c^d \equiv m^{ed} \equiv m^{1+\beta\varphi(n)} \equiv m \pmod{n}.$$

Eve Wants to Know

She knows $85 = pq$ for *some* primes p and q.

If she knew p and q, then she could compute

$\varphi(n) = (p-1)(q-1)$.

Then she could compute $d \equiv e^{-1} \pmod{\varphi(n)}$, and decrypt any messages sent to Bob with this public key.

If only she could factor n...

Eve Interferes and accesses c, but...

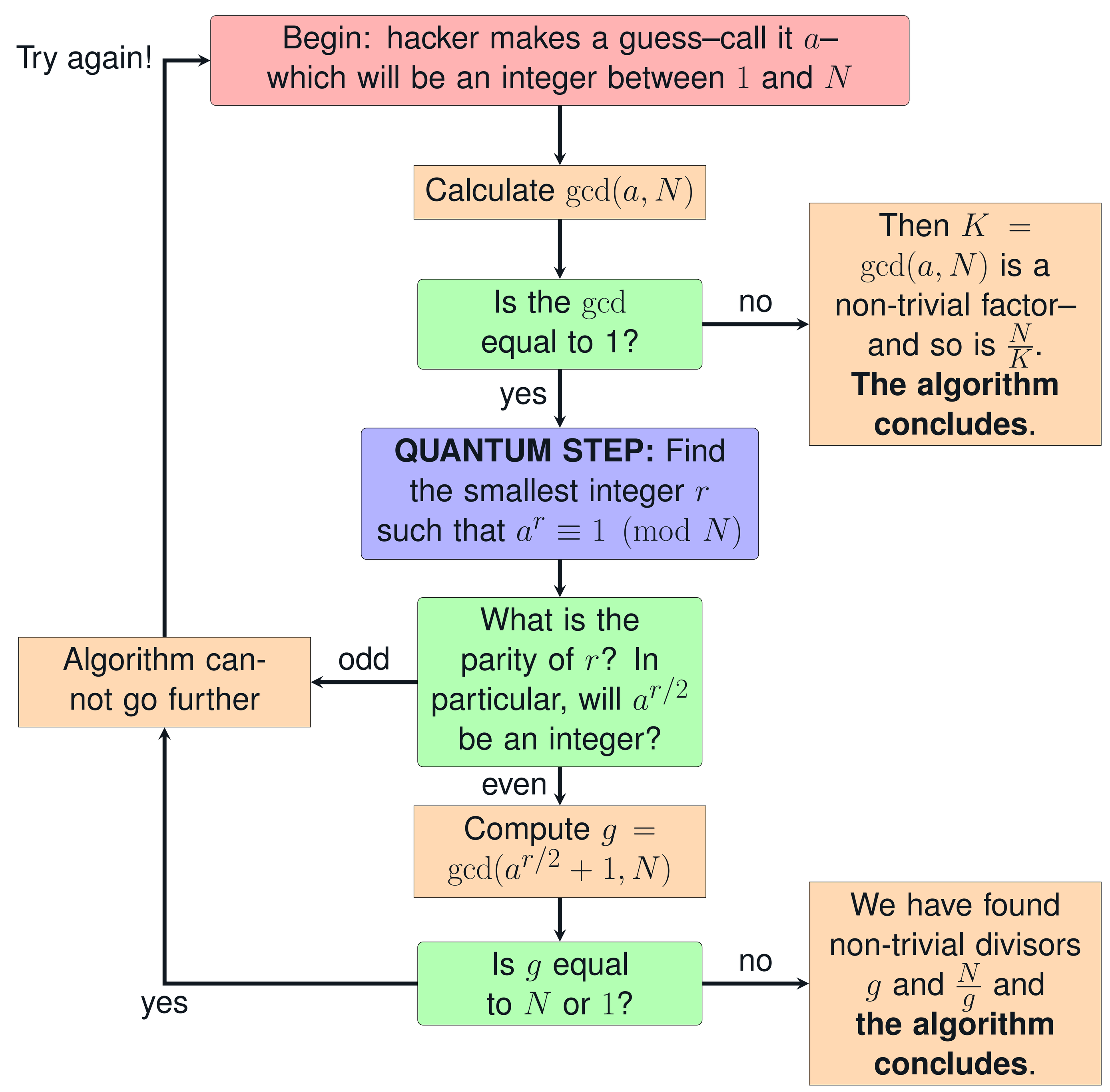
She only knows c, n, and e. She knows that

$$c \equiv m^e \pmod{n},$$

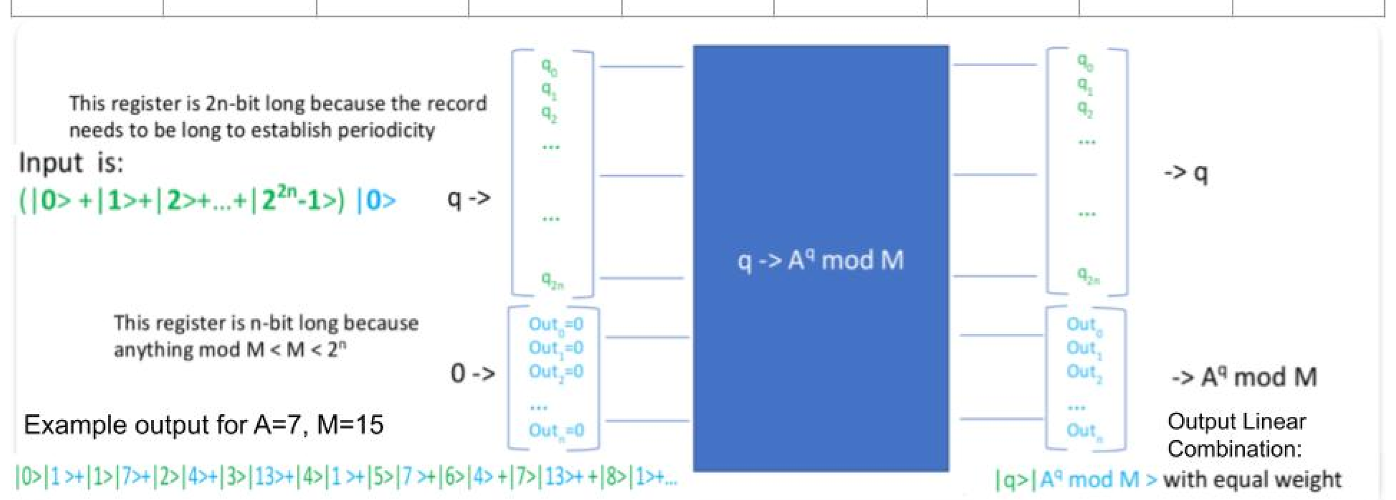
but can't figure out what m is without factoring n.

Shor's Algorithm

Herein lies our ultimate tool for decrypting the once-impenetrable fortress of RSA encryption: Shor's algorithm, a quantum breakthrough poised to shatter the very foundation of digital security. We have a simple goal: **find the factors of some integer** N. Let's go through the process to do so:



0	1	2	3	4	5	6	7	q
1	7	4	13	1	7	4	13	7^qmod15



Shor's Algorithm used to find the period of

$$f(q) = A^q \pmod{M}$$

where $M = 15$, and $A = 7$. In this case, it is easy to see that $p = 4$. Images adapted from [5].

The Quantum Step

Without regard for the quantum step, Shor's algorithm is a rather straightforward way to find the factors of large integers and destroy cybersecurity. But the quantum step is crucial–how does it go? Let's discuss:

In quantum mechanics, information is encoded by **qubits** which can exist simultaneously (until measured). These qubits can be superpositioned or entangled using **quantum gates** in ways that determine their outcome when measuring them–how likely certain qubits are measured are determined by their **probability amplitude**. Here is the process:

$$\begin{array}{l} |x\rangle \; -\boxed{H}- \\ |a^x \pmod{N}\rangle \end{array} \quad \frac{1}{\sqrt{Q}} \sum_{x=0}^{Q-1} |x\rangle |a^x \pmod{N}\rangle \quad (*)$$

where $Q > N^2$. We measure $(*)$, and we obtain a quantum interference that collapses the first register containing $|x\rangle$ into a singular $y = a^{x_0} \pmod{N}$. But properties of modular arithmetic tell us that $x_0 + kr$ satisfy the equation for all k and a singular r, so we take a superposition of x_0 like so:

$$|x_0\rangle \; -\boxed{H}- \; \frac{1}{\sqrt{r}} \sum_{k=0}^{r-1} |x_0 + kr\rangle \quad (**)$$

The output reveals the periodicity of the function, precisely captured by r. And in order to extract this value from our periodic function, we will apply a **quantum Fourier transform**, which, when applied to $(**)$, gives us

$$\frac{1}{\sqrt{Q}} \sum_{c=0}^{Q-1} e^{\frac{2\pi i x_0 c}{Q}} \left(\sum_{k=0}^{r-1} e^{\frac{2\pi i k r c}{Q}} \right) |c\rangle$$

This looks quite messy, but what's important to us is that this QFT will **constructively interfere** at multiples of r–take a look at the series in the parenthesis. It is a geometric series whose value will be large when $r\frac{c}{Q}$ is close to an integer. We can measure the c that gives us an integer multiple of r by using a classical post-processing algorithm such as continued fractions to obtain r–and once we do, the process concludes, and the algorithm continues.

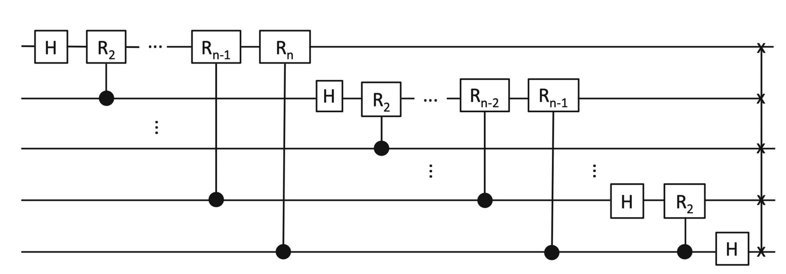


Fig. 5: Quantum Fourier Transform

Image from [4]

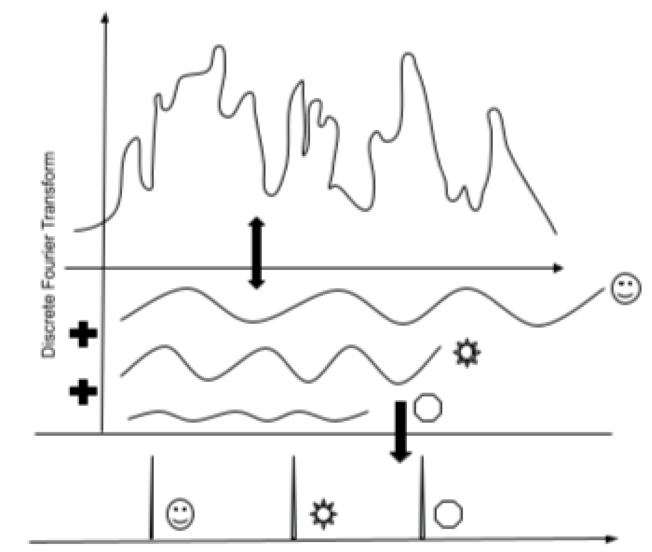


Fig. 6: Fourier Transform

Acknowledgements & References

We would like to thank the DRP team for organizing the program this year. We would also like to thank our mentor Kyle for his patient guidance and mentorship during this whole process.

References

[1] *Bloch Sphere*. `https://en.wikipedia.org/wiki/Bloch_sphere`. Accessed: 1 May 2024.

[2] David J. Hunter. *Cryptography and Coding Theory*. URL: `https://djhunter.github.io/cryptography/`.

[3] *Karen Plankton*. URL: `https://spongebob.fandom.com/wiki/Karen_Plankton`.

[4] Ray LaPierre. *Introduction to quantum computing*. Springer Nature, 2021.

[5] Michael Pushkarsky. "Seminar - quantum computers". Unpublished.

[6] Wade Trappe. *Introduction to cryptography with coding theory*. Pearson Education India, 2006.