Part 4

Schrödinger's cat

- The standard interpretation of quantum mechanics places a lot of emphasis on the act of measurement. Before measurement, quantum systems exist in many states at once.
- After measurement, the system "collapses" into a specific value, so it's natural to ask what's really going on when measurements don't take place.
- There isn't a clear answer, and different ideas can go in some really wild directions.

- One of the first lessons that physicists learned when they started examining subatomic systems in the early 20th century was that we do not live in a deterministic universe.
- In other words, we cannot precisely predict the outcome of every experiment.

- For example, if you shoot a beam of electrons through a magnetic field, **half of the electrons will curve in one direction while the other half will curve in the opposite direction.**
- While we can build mathematical descriptions of where the electrons go as a group, we cannot say which direction each electron will take until we actually perform the experiment.

- In quantum mechanics, this is known as **superposition**. For any experiment that can result in many random outcomes, before we make a measurement, the system is said to be in a superposition of all possible states simultaneously.
- When we make a measurement, the system "collapses" into a single state that we observe.

- The tools of quantum mechanics are there to make some sense out of this chaos. Instead of giving precise predictions for how a system will evolve, quantum mechanics tells us how superposition (which represents all the various outcomes) will evolve.
- When we make a measurement, quantum mechanics tells us the probabilities of getting one outcome over another.
- And that's it. Standard quantum mechanics is silent as to how this superposition actually works and how measurement does the job of collapsing the superposition into a single result.

Schrödinger's cat

- If we take this line of thinking to its logical conclusion, then measurement is the most important act in the universe. It transforms fuzzy probabilities into concrete results and changes an exotic quantum system into verifiable results that we can interpret with our senses.

- Ironically, Erwin Schrödinger, one of the founders of quantum theory (it's his equation **that tells us how the superposition will evolve in time**), railed against this line of thinking. He developed his famous cat-in-a-box thought experiment, now known as Schrödinger's cat, to show how ridiculous quantum mechanics was.


Schrödinger's traffic light

- Here's a highly simplified version. Put a (live) cat in a box. Also put in the box some sort of radioactive element that is tied to the release of a poisonous gas. It doesn't matter how you do it; the point is to introduce some ingredient of quantum uncertainty into the situation. If you wait awhile, you won't know for sure if the element has decayed, so you won't know if the poison has been released and thus if the cat is alive or dead.

- In a strict reading of quantum mechanics, the cat is neither alive nor dead at this stage; it exists in a quantum superposition of both alive and dead. Only when we open the box will we know for sure, and it's also the act of opening the box that allows that superposition to collapse and the cat to (suddenly) exist in one state or the other.

- Schrödinger used this argument to express his astonishment that this could be a coherent theory of the universe. Are we really to believe that until we open the box that the cat doesn't really "exist" — at least in the normal sense that things are always definitely alive or dead, not both at the same time? To Schrödinger, this was too far, and he quit working on quantum mechanics shortly thereafter.

Probabilistic Physics *vs* Deterministic Physics

Quantum Physics *vs* Classical Physics

Encryption Methods

RSA

- The Rivest-Shamir-Adleman (RSA) encryption algorithm is an asymmetric encryption algorithm that is widely used in many products and services.
- Asymmetric encryption uses a key pair that is mathematically linked to encrypt and decrypt data.
- A private and public key are created, with the public key being accessible to anyone and the private key being a secret known only by the key pair creator.
- With RSA, either the private or public key can encrypt the data, while the other key decrypts it. This is one of the reasons RSA is the most used asymmetric encryption algorithm.

How does RSA work?

- The option to encrypt with either the private or public key provides a multitude of services to RSA users.
- If the public key is used for encryption, the private key must be used to decrypt the data.
- This is perfect for sending sensitive information across a network or Internet connection, where the recipient of the data sends the data sender their public key.

- The sender of the data then encrypts the sensitive information with the public key and sends it to the recipient.
- Since the public key encrypted the data, only the owner of the private key can decrypt the sensitive data. Thus, only the intended recipient of the data can decrypt it, even if the data were taken in transit.

- The other method of asymmetric encryption with RSA is encrypting a message with a private key.
- In this example, the sender of the data encrypts the data with their private key and sends encrypted data and their public key along to the recipient of the data.
- The recipient of the data can then decrypt the data with the sender's public key, *thus verifying the sender is who they say they are.*

- With this method, the data could be stolen and read in transit, but the true purpose of this type of encryption is to *prove the identity of the sender.*
- If the data were stolen and modified in transit, the public key would not be able to decrypt the new message, and so the recipient would know the data had been modified in transit.

- **The technical details** of RSA work on the idea that it is easy to *generate a number by multiplying two sufficiently large numbers together, but factorizing that number back into the original prime numbers is extremely difficult.*
- The public and private key are created with two numbers, one of which is a product of two large prime numbers. Both use the same two prime numbers to compute their value.
- RSA keys tend to be 1024 or 2048 bits in length, making them extremely difficult to factorize, though 1024 bit keys are believed to breakable soon.

- **Who uses RSA encryption?**
- As previously described, RSA encryption has a number of different tasks that it is used for. One of these is digital signing for code and certificates.
- **Certificates can be used to verify who a public key belongs to,** by signing it with the private key of the key pair owner. This authenticates the key pair owner as a trusted source of information.

- Code signing is also done with the RSA algorithm. To ensure the owner is not sending dangerous or incorrect code to a buyer, the code is **signed with the private key of the code creator**.
- This verifies the code has not been edited maliciously in transit, and that the code creator verifies that the code does what they have said it does.

- RSA was used with Transport Layer Security (TLS) to secure communications between two individuals.
- Other well-known products and algorithms, like the Pretty Good Privacy (PGP) algorithm, use RSA either currently or in the past.
- Virtual Private Networks (VPNs), email services, web browsers, and other communication channels have used RSA as well.

- VPNs will use TLS to implement a *handshake* between the two parties in the information exchange.
- The TLS Handshake will use RSA as its encryption algorithm, to verify both parties are who they say who they are.

- **RSA Vulnerabilities**
- Though viable in many circumstances, there are still a number of vulnerabilities in RSA that can be exploited by attackers.
- One of these vulnerabilities is the implementation of **a long key in the encryption algorithm**.
- RSA relies on *the size* of its key to be difficult to break.

- The longer an RSA key, the more secure it is. Using prime factorization, researchers managed to crack a *768 bit key RSA algorithm, but it took them 2 years*, thousands of man hours, and an absurd amount of computing power, so the currently used key lengths in RSA are still safe.
- The National Institute of Science and Technology (NIST) recommends a minimum key length of **2048 bits now**, but many organizations have been using keys of length 4096 bits.

Other ways RSA is vulnerable are:

- **Weak Random Number Generator**: When organizations use weak random number generators, then the prime numbers created by them are much easier to factor, thus giving attackers an easier time of cracking the algorithm.
- **Weak Key Generation**: RSA keys have certain requirements relating to their generation. If the prime numbers are too close, or if one of the numbers making up the private key is too small, then the key can be solved for much easier.
- **Side Channel Attacks**: Side channel attacks are a method of attack that take advantage of the system running the encryption algorithm, as opposed to the algorithm itself. Attackers can analyze the power being used, use branch prediction analysis, or use timing attacks to find ways to ascertain the key used in the algorithm, thus compromising the data.

Shor’s Factoring Algorithm

Shor’s Factoring Algorithm

- Anyone interested in learning about quantum computing **cannot** avoid hearing about **Shor’s Factoring Algorithm or (Shor’s Algorithm)**. It is one of the few textbook quantum algorithms, which means that it remains one of the rare examples of quantum computational advantage.
- In other words, *the algorithm can compute something quantumly that is harder and slower to compute classically.*
- In fact, this particular algorithm can compute something that is for all practical intents and purposes, as far as we know, *impossible* to do classically at any useful scale.

- Among all textbook algorithms, Shor's Factoring Algorithm stands out from the crowd.
- There are two excellent reasons for this. **First**, it can factor numbers exponentially *faster* than any known classical algorithm.
- While all of the textbook quantum algorithms offer computational advantages, Shor's Algorithm is one of the elite few with an *exponential speedup*, as well as one of the elite few with *a practical application*.
- **Most importantly**, its potential to factor numbers in reasonable *timeframes* directly threatens the world's most popular cryptosystems.

- The significance of that cannot be overstated. RSA encryption, protecting financial transactions worldwide, works by multiplying two huge prime numbers. Prime numbers cannot be divided into integers other than themselves and the number one.
- Their products are so large, that there is no known way to efficiently factor them classically.
- Think of factoring as reverse multiplication, determining the prime numbers used, thus allowing unauthorized decryption of internet communications.

Shor's algorithm is a quantum algorithm for integer factorization, which can be used to break certain cryptographic systems that are based on the difficulty of factoring large integers. It was proposed by mathematician Peter Shor in 1994 and is considered to be one of the most significant advances in quantum computing. The algorithm has not yet been implemented on a large scale, but it has been demonstrated in small-scale experiments and it is believed that it could be used to break many of the public key cryptography systems currently in use if a sufficiently large and powerful quantum computer were built.

ChatGPT

- Shor's algorithm works by using the principles of quantum mechanics to perform certain calculations much more quickly than is possible with a classical computer.
- Specifically, it uses the principles of *quantum parallelism* and *quantum interference* to find the prime factors of a given integer in a fraction of the time that would be required by a classical computer.

- One way to understand the algorithm is to consider the example of trying to factor a number, such as 15.
- On a classical computer, one might try dividing the number by every integer from 2 up to 15, to see if any of them divide evenly into 15.
- This is a relatively slow process, especially for large numbers.
- Shor's algorithm, on the other hand, uses quantum mechanics to perform this process much more quickly by trying **many divisions at once**, using a process called **quantum parallelism**.

- In addition to **quantum parallelism**, Shor's algorithm also makes use of **quantum interference**, which is a phenomenon that occurs when two or more quantum states "interfere" with each other in a way that can produce unexpected results.
- By carefully controlling the quantum states of a quantum computer, it is possible to use quantum interference to amplify certain desired outcomes and suppress others, effectively making it easier to find the prime factors of a given integer.

- Despite its potential power, Shor's algorithm has not yet been implemented on a large scale, and it is not clear when or if this will happen.
- However, the development of the algorithm has had a significant impact on the field of cryptography, and it has sparked ongoing research into the potential uses and limitations of quantum computers.

What is Shor's algorithm in quantum computing?

- Shor’s Factoring Algorithm put quantum computing on the well-known map.
- By threatening animated version, national governments, whole industries, and the public at large were forced to take notice of this relatively new technology.
- Decades later, this algorithm remains the standard bearer of quantum algorithms. Significant effort has been underway for a long time to protect global financial systems, national security, and all other uses of cryptography, for that matter.
- There is a clear and present geopolitical danger of anyone, especially a state actor, developing sufficient quantum computing power to employ Shor’s algorithm before quantum-safe cryptosystems can be universally deployed

- Almost as important, Shor's algorithm has led to the present-day investment of billions of dollars into quantum technologies, including quantum computing.
- Three decades since the algorithm was discovered, the search continues for other practical applications for which exponential speedups can be realized.
- If at least one algorithm can achieve this, the thinking goes, there must be others.
- After all these years, the most likely candidates borrow from Shor's algorithm.