



# Impact of Quantum Computing on IoT Security

Aref MEDDEB

NOCCS Lab

National Engineering School of Sousse

University of Sousse



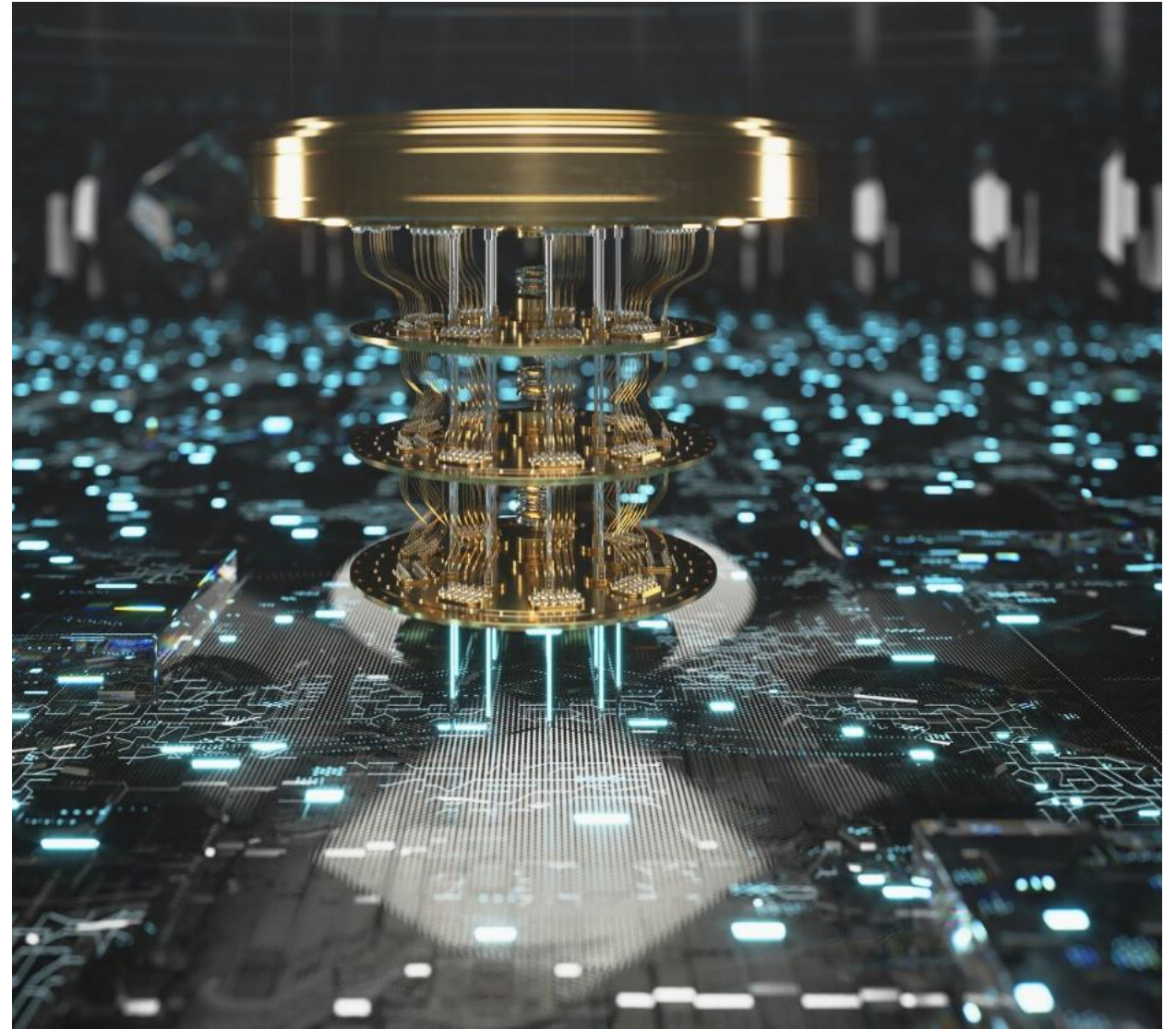
# Outline

What is Quantum Computing ?

Quantum Computing and  
Cybersecurity

Quantum Algorithms

Impact of Quantum Computing  
on IoT Security





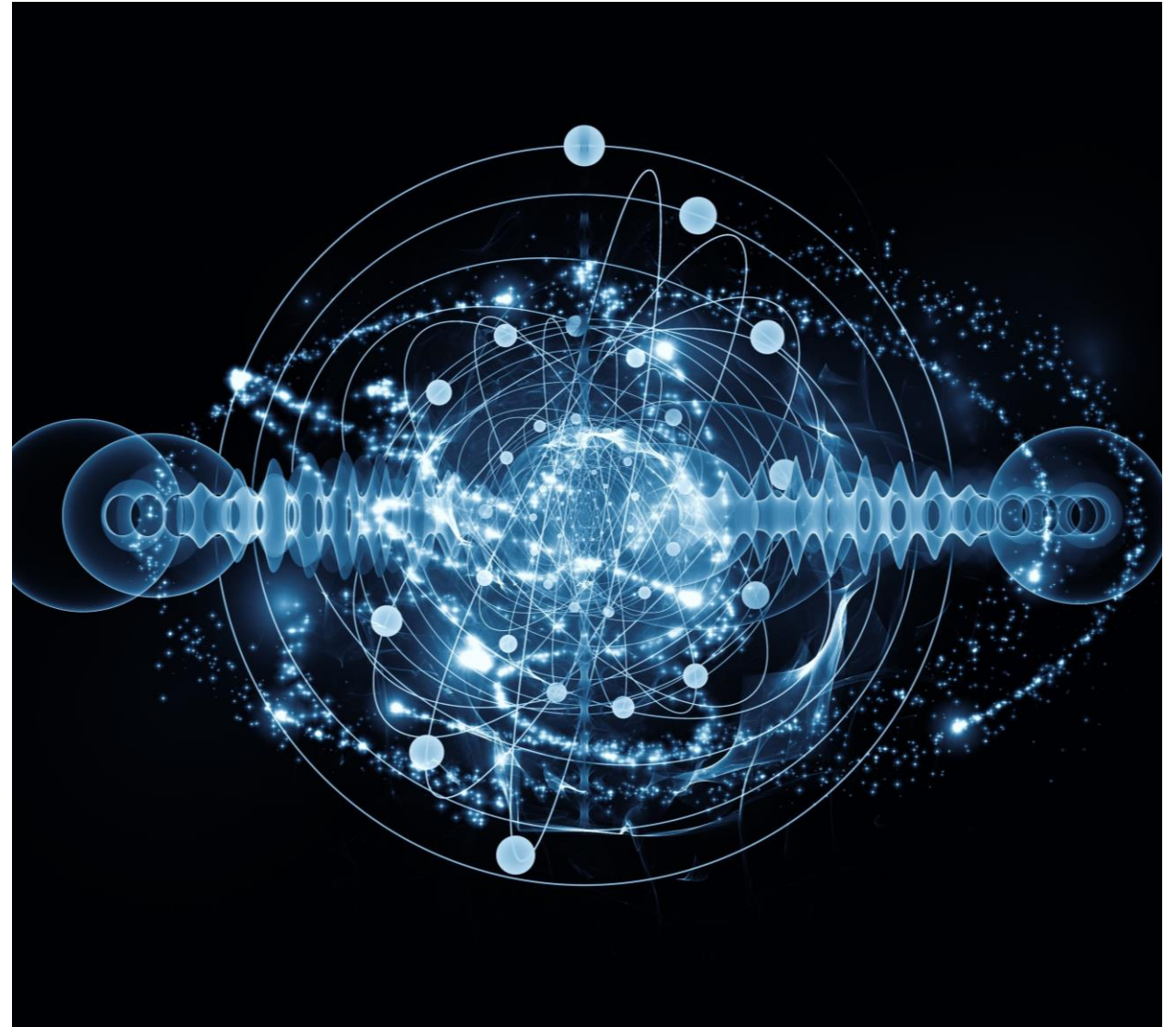
# What is Quantum Computing ?

Quantum Mechanics In a nutshell

Quantum bit

Quantum Computer

Examples



# Quantum Mechanics in a nutshell

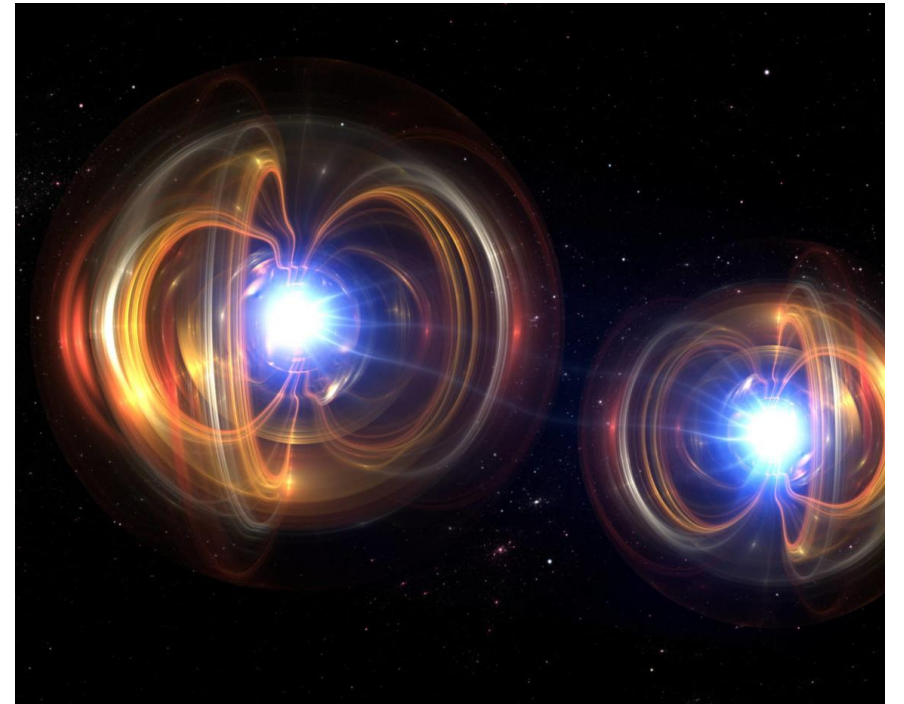
---

Quantum mechanics is a fundamental theory in physics that provides a description of the physical properties of nature at the scale of atoms and subatomic particles.

It provides a mathematical description of how elementary particles move and interact.

It is based on the wave-particle dual description formulated by Bohr, Einstein, Heisenberg, Schrödinger, and others.

The basic units of nature are particles, but the description of their motion involves wave mechanics.



Textbook: Mahan, Gerald D. "Introduction." *Quantum Mechanics in a Nutshell*, Princeton University Press, 2009, pp. 1-13. *JSTOR*, <https://doi.org/10.2307/j.ctt7s8nw.4>. Accessed 15 Feb. 2023.

# Quantum Particles

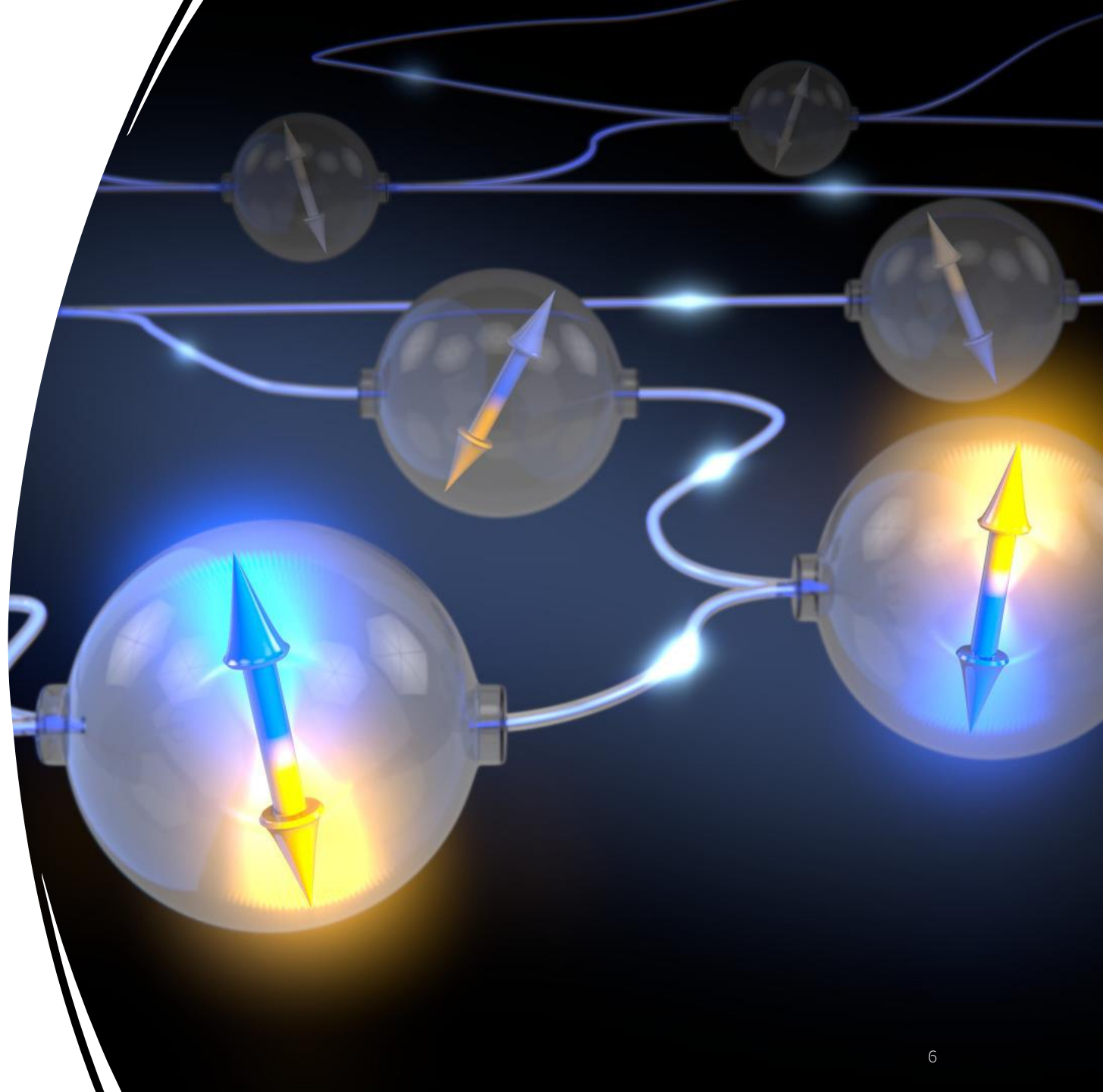
	mass →	charge →	spin →					
QUARKS	$\approx 2.3 \text{ MeV}/c^2$	$2/3$	$1/2$	<b>u</b> up	$\approx 1.275 \text{ GeV}/c^2$	$2/3$	$1/2$	<b>c</b> charm
					$\approx 173.07 \text{ GeV}/c^2$	$2/3$	$1/2$	<b>t</b> top
					0	0	1	<b>g</b> gluon
								$\approx 126 \text{ GeV}/c^2$
								0
								0
								0
								<b>H</b> Higgs boson
LEPTONS	$\approx 4.8 \text{ MeV}/c^2$	$-1/3$	$1/2$	<b>d</b> down	$\approx 95 \text{ MeV}/c^2$	$-1/3$	$1/2$	<b>s</b> strange
					$\approx 4.18 \text{ GeV}/c^2$	$-1/3$	$1/2$	<b>b</b> bottom
					0	0	1	<b><math>\gamma</math></b> photon
GAUGE BOSONS	$0.511 \text{ MeV}/c^2$	-1	$1/2$	<b>e</b> electron	$105.7 \text{ MeV}/c^2$	-1	$1/2$	<b><math>\mu</math></b> muon
					$1.777 \text{ GeV}/c^2$	-1	$1/2$	<b><math>\tau</math></b> tau
					91.2 $\text{GeV}/c^2$	0	1	<b>Z</b> Z boson
	$< 2.2 \text{ eV}/c^2$	0	$1/2$	<b><math>\nu_e</math></b> electron neutrino	$< 0.17 \text{ MeV}/c^2$	0	$1/2$	<b><math>\nu_\mu</math></b> muon neutrino
					$< 15.5 \text{ MeV}/c^2$	0	$1/2$	<b><math>\nu_\tau</math></b> tau neutrino
					80.4 $\text{GeV}/c^2$	$\pm 1$	1	<b>W</b> W boson

# Quantum information science

---

Quantum Mechanics lay the foundation of all quantum physics including quantum chemistry, quantum field theory, quantum technology, and quantum information science.

Quantum information science is a field that combines the principles of quantum mechanics with information science to study the processing, analysis, and transmission of information.



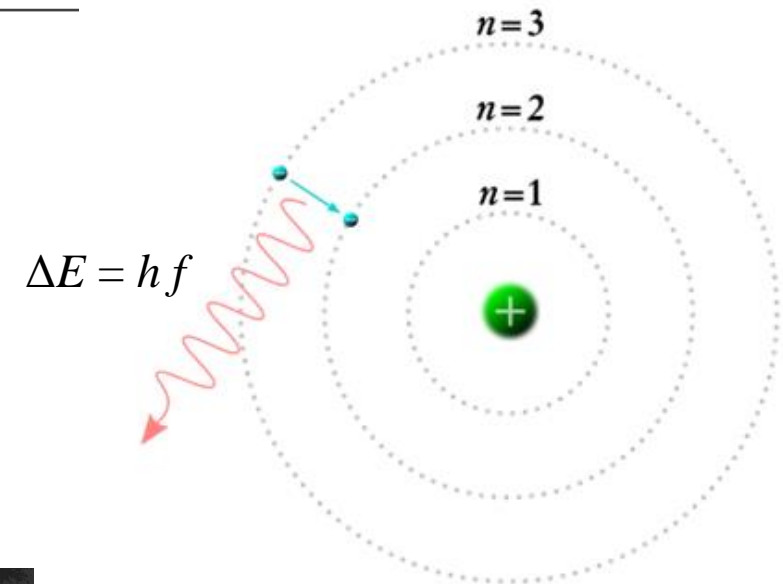
# Three fundamental principles

---

**Quantization:** Energy, momentum, angular momentum, and other quantities of a bound system are restricted to discrete values. First discovered by Max Planck in 1920.

**Wave-particle duality:** Objects have characteristics of both particles and waves. Discovered by Louis de Broglie in 1924 and further developed by others.

**Uncertainty :** There are limits to how accurately the value of a physical quantity can be predicted prior to its measurement given a complete set of initial conditions. Discovered by Werner Heisenberg in 1927.



In 1913, Niels Bohr proposed that the electrons in an atom were restricted to certain energy levels or orbits, and that when electrons absorbed or emitted energy, they did so in discrete packets or quanta.



# Planck's constant

---

The important parameter in quantum mechanics is Planck's constant  $h=6.626 \cdot 10^{-34} \text{ Js}$  or **Joule/Hz**.

It relates the energy of a photon to its frequency.

It is common to divide it by  $2\pi$  and to put a slash through the symbol:  $\hbar=1.054 \cdot 10^{-34} \text{ Js}$ .



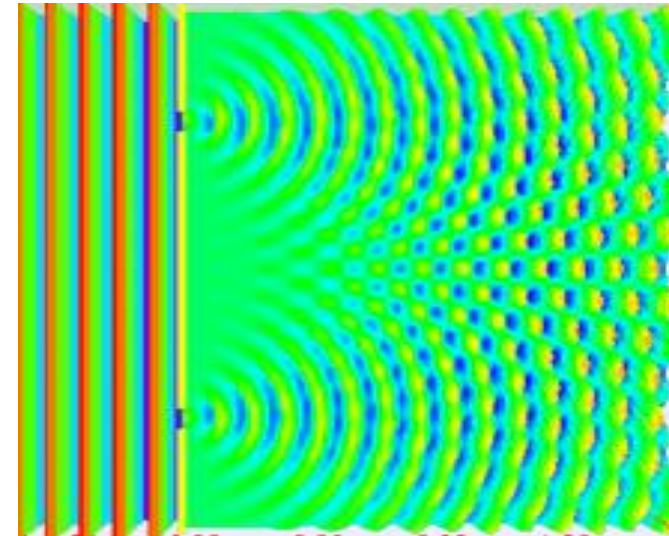
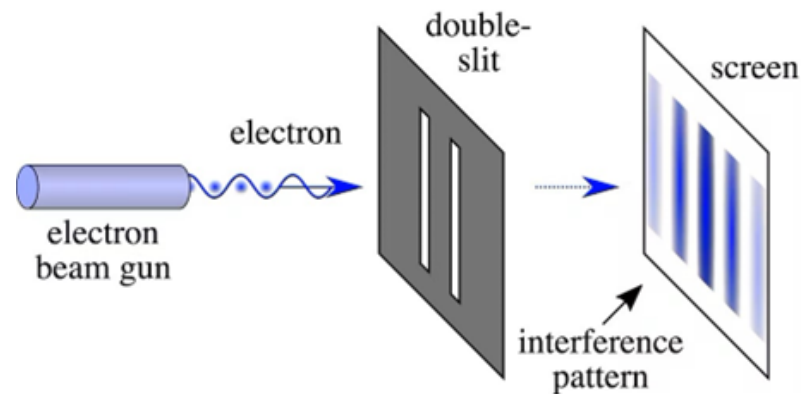


# Uncertainty principle

---

An electron can be described by a wave function, which associates to each point in space a probability amplitude.

The Born rule assigns to these amplitudes a probability density function for the position that the electron would be in when an experiment is performed to measure it.



# Superposition principle



The idea of superposition was first proposed by Erwin Schrödinger in 1926.

A particle can exist in multiple states or locations at the same time until it is observed or measured.

The Schrödinger equation relates the probability amplitudes associated to one moment of time to a collection of probability amplitudes associated to another moment in time.

This is the best theory so far!

$$i\hbar \frac{\partial}{\partial t} \Psi = \hat{H} \Psi$$

Diagram illustrating the Schrödinger equation with annotations:

- $i$ : square root of minus one
- $\hbar$ : Planck's constant
- $\frac{\partial}{\partial t}$ : rate of change with respect to time
- $\Psi$ : quantum wavefunction
- $\hat{H}$ : Hamiltonian operator

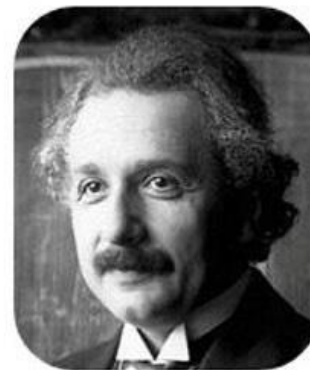
# The Entanglement Phenomenon

---

The concept of entanglement was first introduced in a 1935 paper by Albert Einstein, Boris Podolsky, and Nathan Rosen, which became known as the EPR paradox.

Consider a pair of particles created at the same time and place with opposite properties, such as spin.

The property of one particle is undefined until it is measured, at which point the property of the other particle would become defined instantaneously, even if the particles were separated by a large distance.



**A. Einstein**



**B. Podolsky**



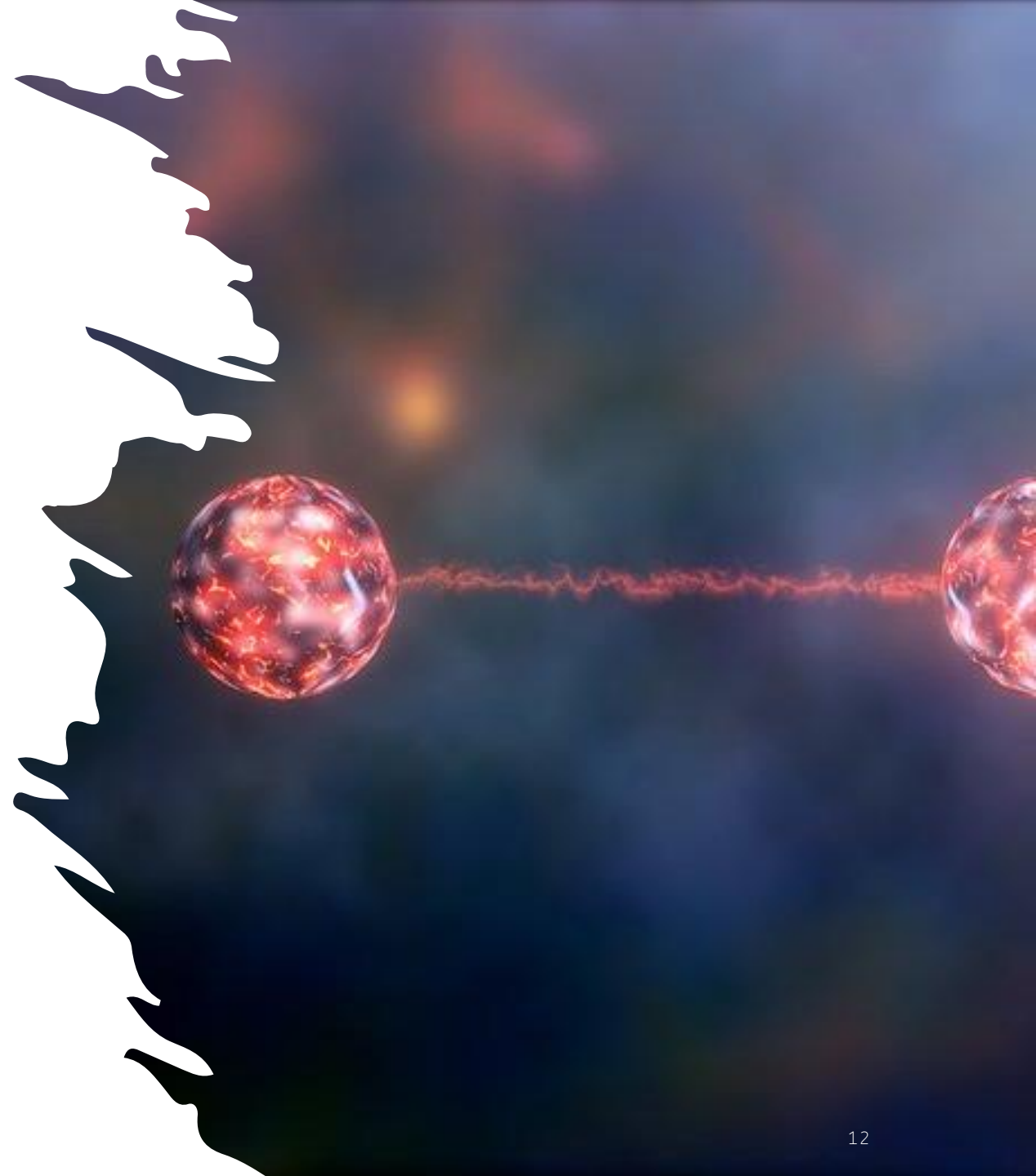
**N. Rosen**

# Entanglement is real, not “spooky”

Einstein, Podolsky, and Rosen believed that this idea violated the principles of locality and causality, which suggest that the behavior of particles can only be influenced by their immediate surroundings and that nothing can travel faster than light.

Experiments and developments have shown that entanglement is a real phenomenon

It is not inconsistent with the principles of locality and causality, but instead represents a fundamental feature of the quantum world.





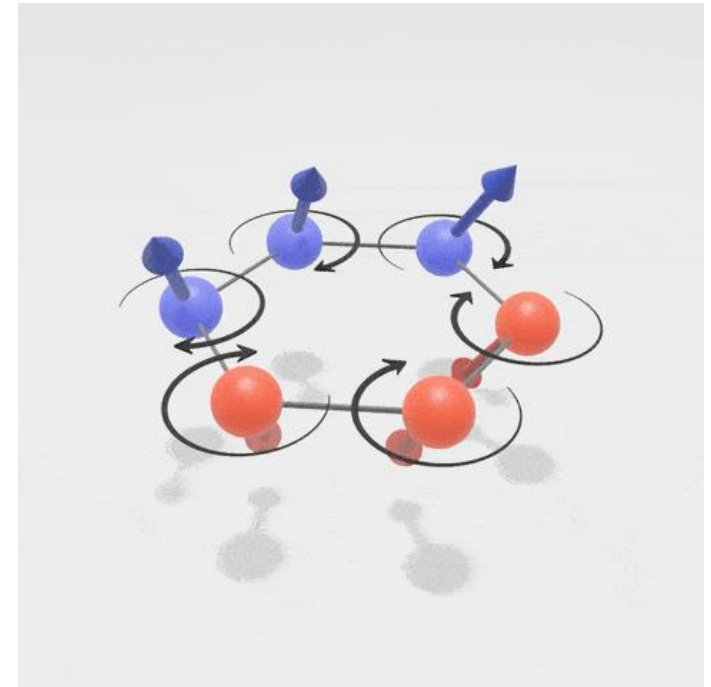
# The spin of an electron

---

The concept of electron spin was first proposed by George Uhlenbeck and Samuel Goudsmit in 1925 to explain certain features of the atomic spectra.

The spin of an electron is an intrinsic property of the electron itself that describes its angular momentum.

It is not related to the electron's orbital motion around the nucleus.



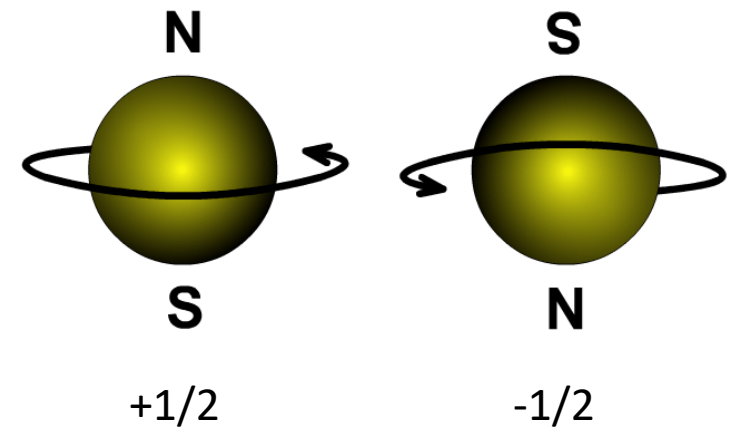
# The spin of an electron

---

The spin of an electron is typically represented by the symbol " $s$ " and has a value of  $1/2$  in units of the reduced Planck constant  $\hbar$ .

The electron can have two possible spin states, which are usually denoted as "spin-up" and "spin-down".

The spin of an electron has several important applications in physics, including the behavior of atoms and the properties of materials.



# Photon polarization

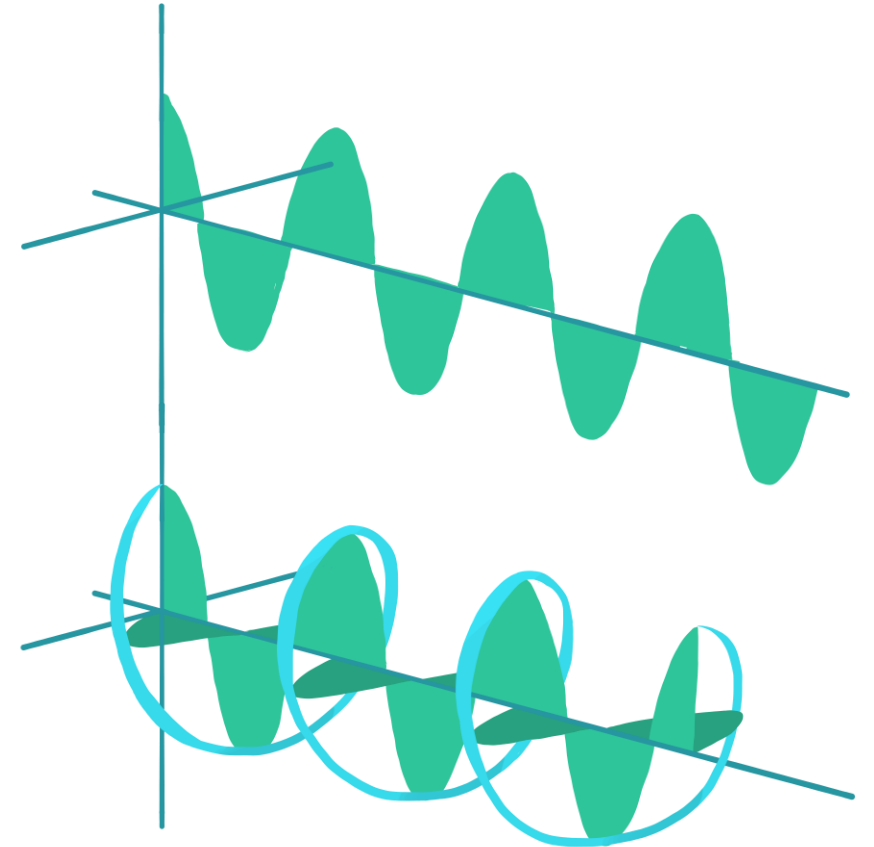
---

The polarization of a photon is a property that describes the orientation of the electric field oscillation of the photon.

The polarization of a photon can be linear or circular.

In the case of linear polarization, the electric field oscillates along a straight line, while in circular polarization, the electric field rotates around the direction of the photon's motion.

Circular polarization can be further classified as left-handed or right-handed, depending on the direction of rotation.



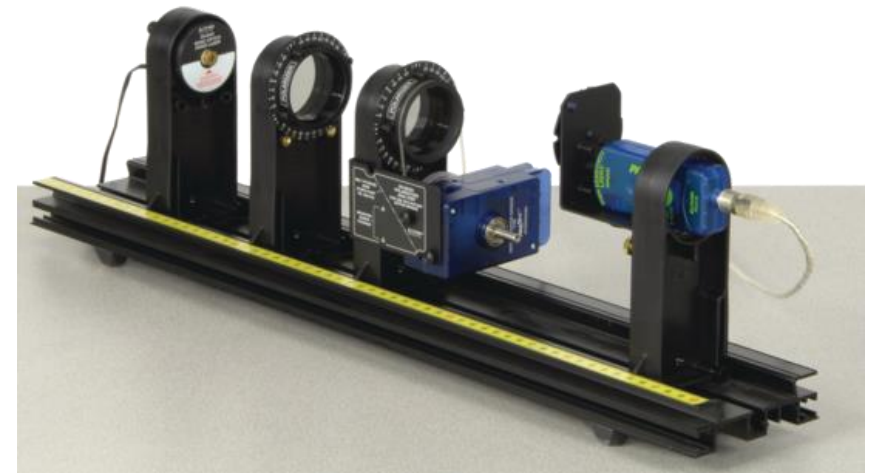
# Photon polarization applications

---

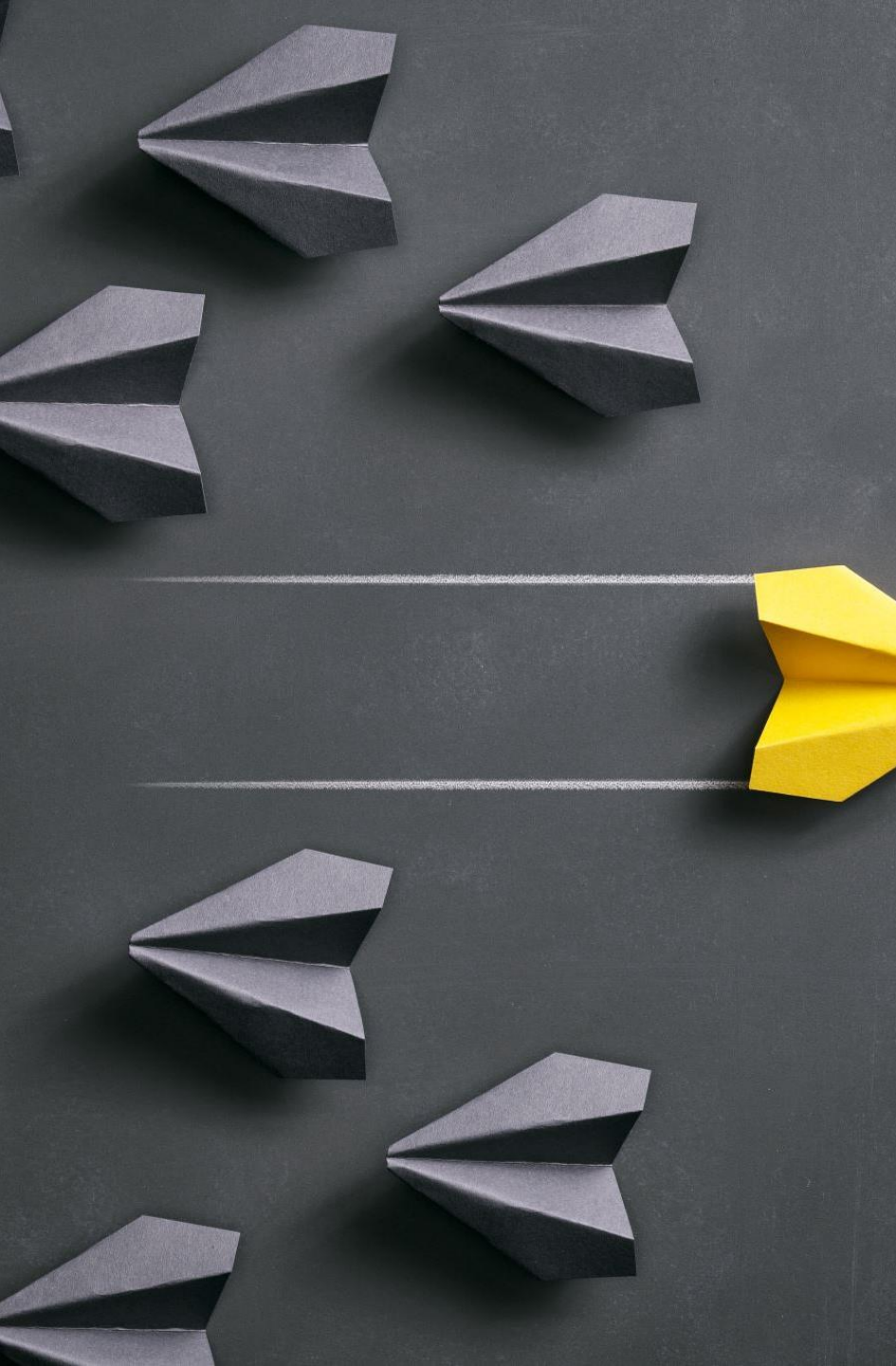
The polarization of a photon has important applications in optical communication and quantum information.

In optical communication, information can be encoded in the polarization of photons, and in quantum information, qubits can be implemented using the polarization states of photons.

The polarization of a photon can be manipulated using various techniques, such as polarization filters or wave plates, which are optical components that modify the polarization of light.







---

# The no-cloning theorem

---

A fundamental principle formulated by Wootters and Zurek in 1982,

it is impossible to create an exact copy of an unknown quantum state without changing its properties.

The theorem has important implications for quantum information processing, cryptography, and communication

It means that information encoded in a quantum state cannot be intercepted or copied without being detected.

This makes it impossible to create a perfect quantum teleportation system, where quantum states can be transmitted over long distances without being physically transported.

It is still possible to create approximate copies of quantum states using estimation.



# Quantum bit

---

A quantum bit, or qubit, is the basic unit of quantum information.

In classical computing, the basic unit of information is the classical bit, which can take on one of two values: 0 or 1.

A qubit can exist in a superposition of two states, which means it can represent a 0 and a 1 at the same time.

This property of superposition allows quantum computers to perform certain types of calculations much faster than classical computers.

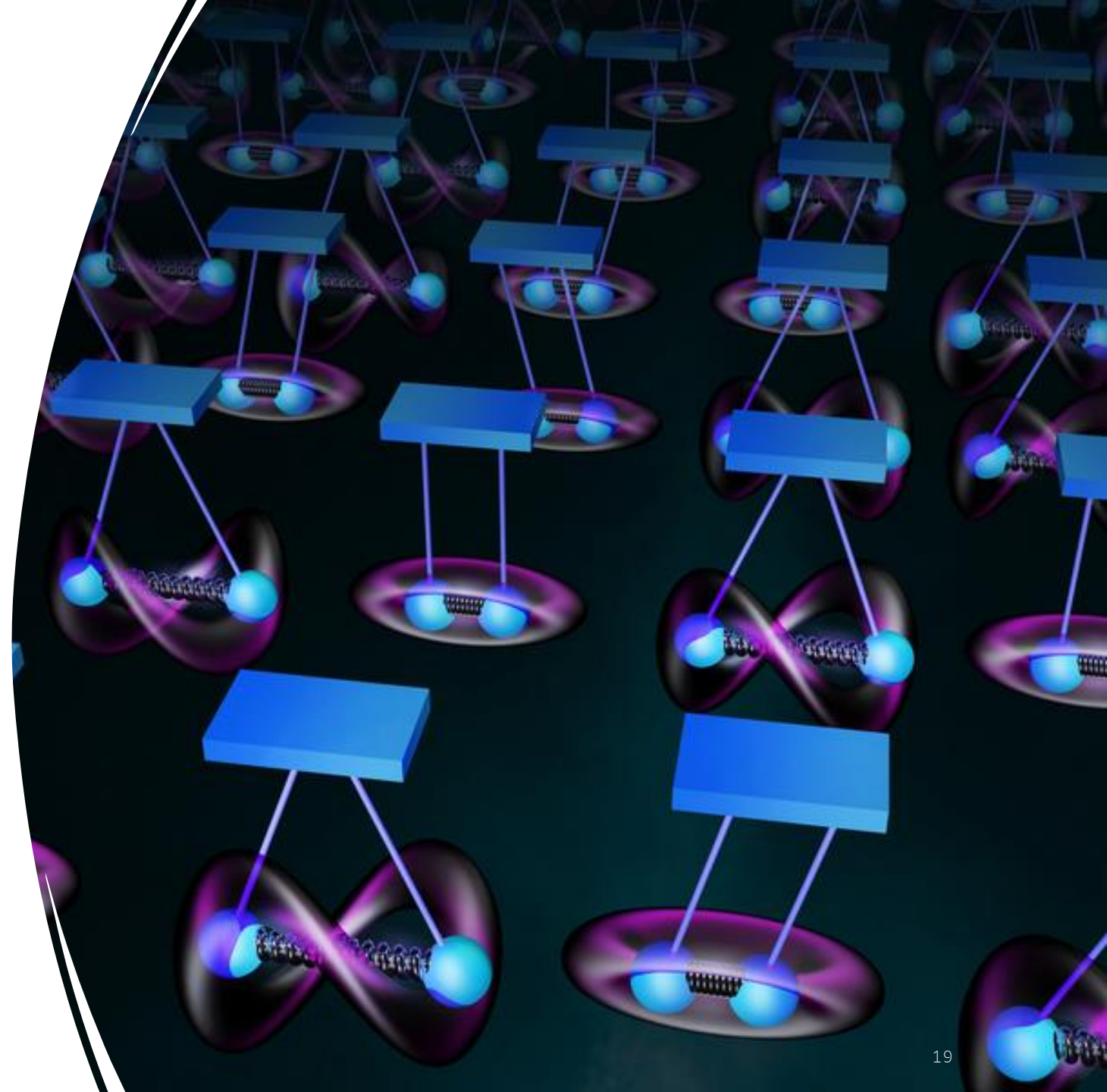
# How to make a Qubit ?

---

A qubit can be implemented using a variety of physical systems, such as the spin of an electron, the polarization of a photon, or the vibration of an atom.

The state of a qubit can be manipulated using quantum gates, which are analogous to classical logic gates, but operate on quantum states.

Measuring a qubit collapses its superposition into one of the two possible classical states, either 0 or 1.





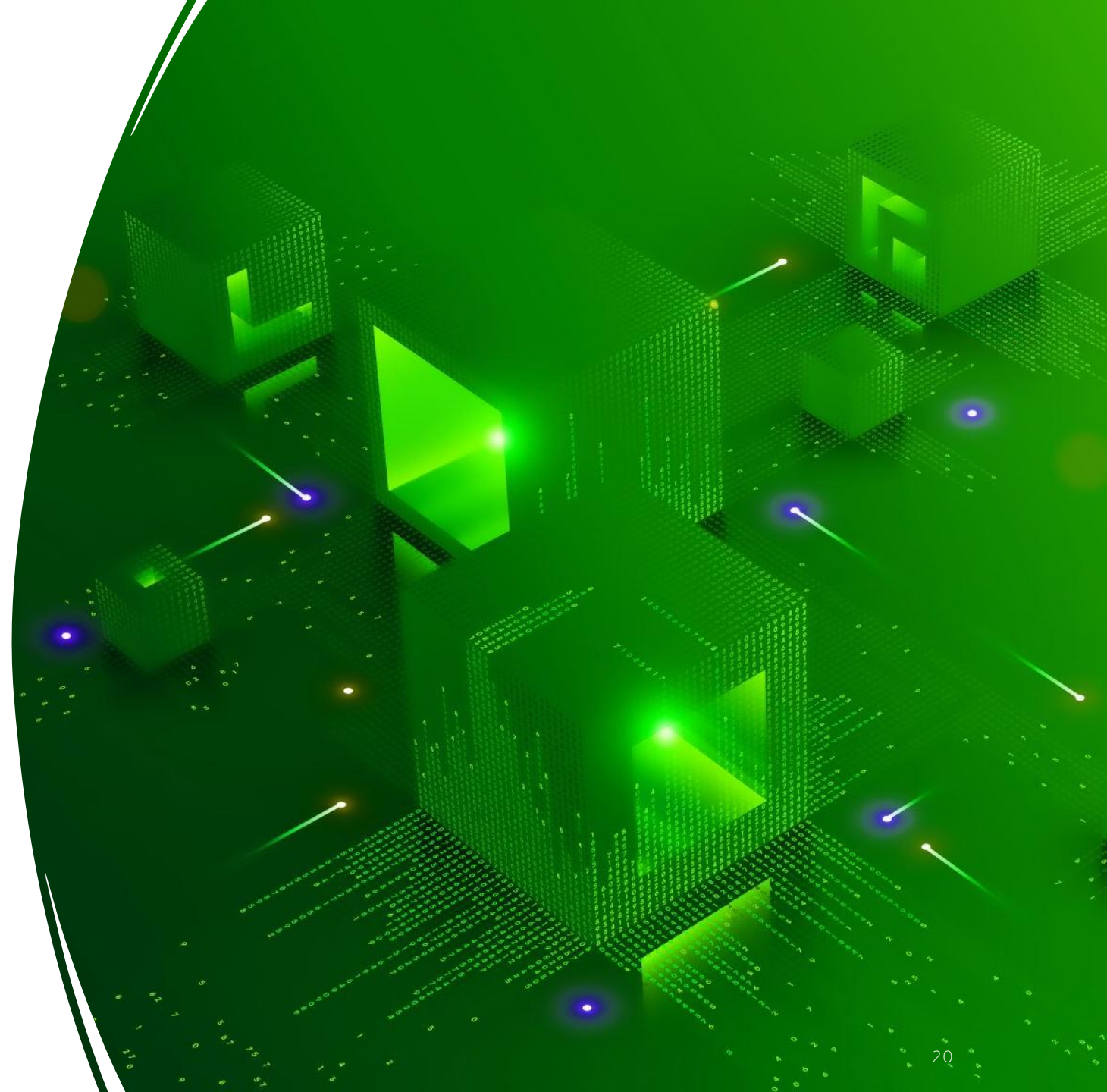
# Maintaining the coherence of the qubits

---

One of the challenges in building a practical quantum computer is to maintain the coherence of the qubits.

Qubits are highly sensitive to their environment and can easily lose their quantum properties through interactions with other particles.

The development of techniques to control and protect qubits is a major area of research in quantum computing.





# Qubit Dirac notation

Qubits are typically represented using a notation called the Dirac notation or bra-ket notation, which was developed by Paul Dirac.

A qubit is represented as a vector in a two-dimensional complex Hilbert space, which is also known as the state space.

The Dirac notation uses two symbols: the **bra** vector  $\langle |$  and the **ket** vector  $| \rangle$ .

The bra vector represents the complex conjugate of a ket vector, and the ket vector represents a quantum state.

The bra vector and ket vector are combined to form a complex inner product, which gives the probability amplitude for a quantum measurement.

$$\begin{aligned} \langle 0| &= (1 \ 0) & |0\rangle &= \begin{pmatrix} 1 \\ 0 \end{pmatrix} \\ \langle 1| &= (0 \ 1) & |1\rangle &= \begin{pmatrix} 0 \\ 1 \end{pmatrix} \end{aligned}$$

# The quantum state

---

The quantum state  $|\Psi\rangle$  (*Psi*) is the linear combination of  $|0\rangle$  and  $|1\rangle$  with the probability of being in state  $|0\rangle$  is  $|\alpha|^2$ , and the probability of being in state  $|1\rangle$  is  $|\beta|^2$ .

$\alpha, \beta \in \mathbb{C}^2$  are the probability amplitudes while  $\alpha^*$  and  $\beta^*$  are the complex conjugates of  $\alpha$  and  $\beta$ , respectively.

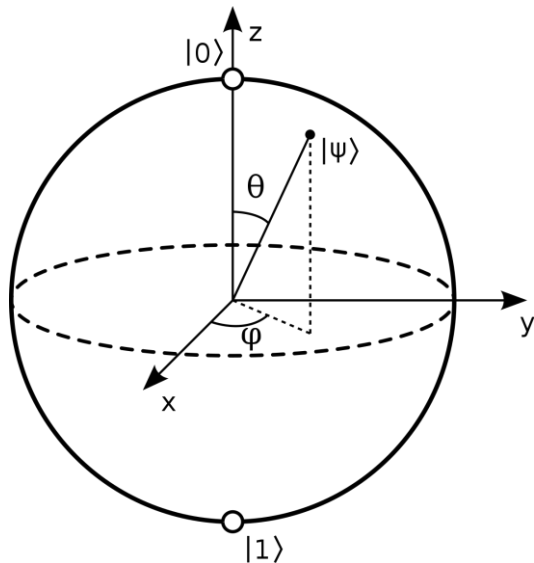
Normalization constraint:  $|\alpha|^2 + |\beta|^2 = 1$ .

$$|\Psi\rangle = \alpha |0\rangle + \beta |1\rangle$$

$$\langle 0| = (1 \ 0) \quad |0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

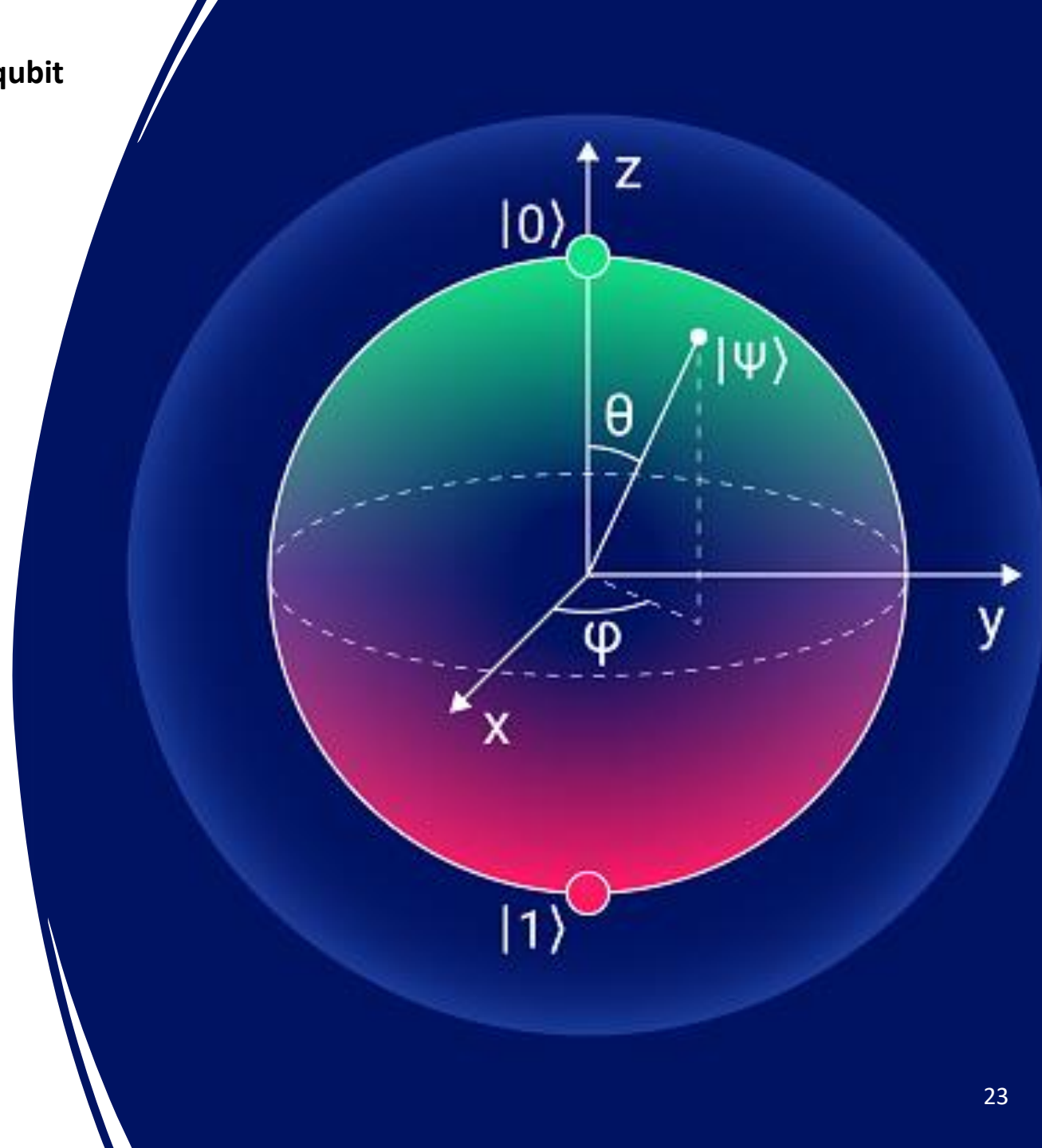
$$\langle 1| = (0 \ 1) \quad |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

## Geometrical representation of the possible states of a qubit



# Bloch sphere

$$|\psi\rangle = \cos \frac{\theta}{2} |0\rangle + e^{i\varphi} \sin \frac{\theta}{2} |1\rangle$$



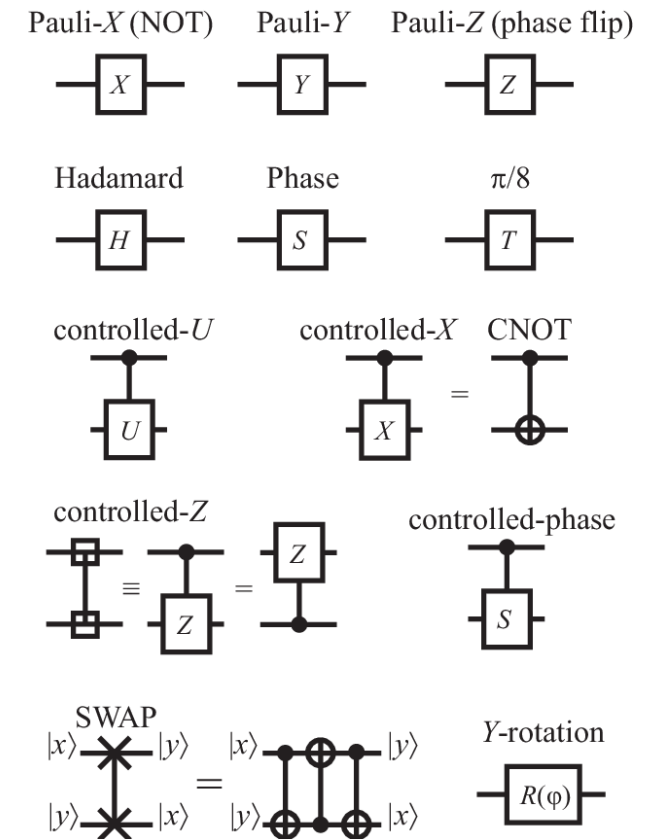
# Qubit operations

Qubit operations are the basic operations that can be performed on qubits to manipulate their quantum states.

These operations can be combined to implement more complex quantum algorithms and circuits.

Implementing quantum gates is challenging due to the inherent noise and decoherence in real-world quantum systems, which can lead to errors in the computation.

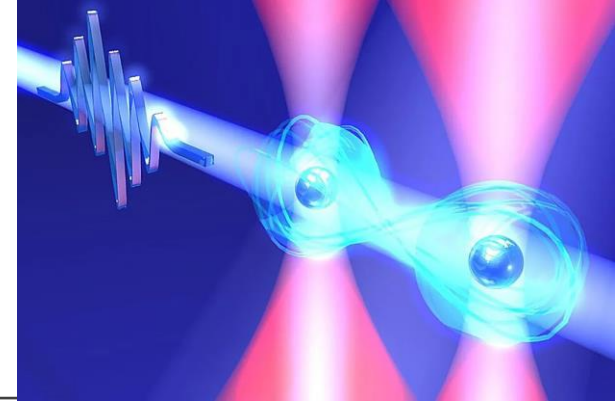
Developing error-correction codes and techniques to suppress noise and decoherence is an active area of research in quantum computing.





# Common Gates

---



1. Pauli-X, Pauli-Y, and Pauli-Z gates: Single-qubit operations that correspond to flips around the x, y, and z axes of the Bloch sphere, respectively.
2. Hadamard gate: Single-qubit gate that puts a qubit into an equal superposition of the 0 and 1 states.
3. Controlled NOT (CNOT) gate: Two-qubit gate that performs a conditional operation on the second qubit, based on the state of the first qubit. It flips the second qubit if the first is in the state  $|1\rangle$ .
4. SWAP gate: Two-qubit gate that exchanges the states of two qubits.
5. Controlled-phase gate: Two-qubit gate that adds a phase to the second qubit's state, based on the state of the first qubit.

# Pauli-X, Pauli-Y & Pauli-Z Gates

---

Single qubit gates:

*Dirac notation*

$$|0\rangle \rightarrow |1\rangle, \quad |1\rangle \rightarrow |0\rangle$$

$$|0\rangle \rightarrow i|1\rangle, \quad |1\rangle \rightarrow -i|0\rangle$$

$$|0\rangle \rightarrow |0\rangle, \quad |1\rangle \rightarrow -|1\rangle$$

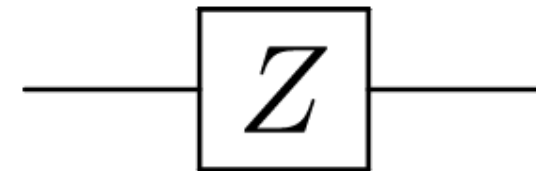
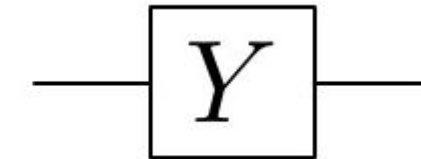
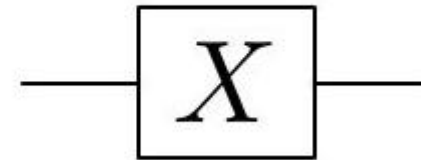
*Matrix representation*

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

$$Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}$$

$$Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

*Circuit representation*



# Hadamard gate

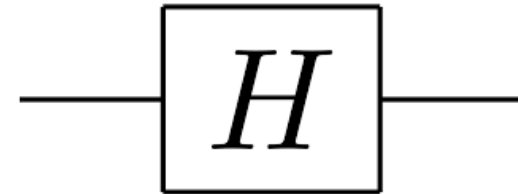
---

Single-qubit gate

Puts a qubit into an equal superposition of the 0 and 1 states

Represented by the matrix:

$$\frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$



$$\begin{aligned} |0\rangle &\rightarrow \frac{1}{\sqrt{2}} (|0\rangle + |1\rangle) \\ |1\rangle &\rightarrow \frac{1}{\sqrt{2}} (|0\rangle - |1\rangle) \end{aligned}$$

# Quantum register with $n$ bits

---

States of a quantum register with  $n$  bits are vectors in a  $2^n$  dimensional complex vector space

Example for 2 bit:

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

# The SWAP gate

---

Also known as the *exchange gate*

Two-qubit gate that exchanges the state of two qubits.

A non-trivial gate because it cannot be decomposed into a sequence of one-qubit gates.

Represented by the following matrix:

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Fundamental gate often used in quantum algorithms and protocols, such as quantum teleportation and quantum error correction.

Exchanges the quantum states of two qubits, which can be represented as:

$$\text{SWAP } |a\rangle|b\rangle = |b\rangle|a\rangle$$

If the first qubit is in state  $|a\rangle$  and the second qubit is in state  $|b\rangle$ , applying the SWAP gate will result in the first qubit being in state  $|b\rangle$  and the second qubit being in state  $|a\rangle$ .



# CNOT Gate

---

Conditional gate

Performs a conditional operation on the second qubit, based on the state of the first qubit.

Flips the second qubit if the first qubit is in the state  $|1\rangle$ .

Allows the implementation of if-else type of construction:

$$\begin{array}{l} |00\rangle \rightarrow |00\rangle \\ |01\rangle \rightarrow |01\rangle \\ |10\rangle \rightarrow |11\rangle \\ |11\rangle \rightarrow |10\rangle \end{array} \quad \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

# Controlled phase gate

---

The controlled phase gate, also known as the CPHASE gate or the controlled Z gate, is a two-qubit quantum gate that applies a phase shift to the second qubit if and only if the first qubit is in the state  $|1\rangle$ .

If the first qubit is in the state  $|0\rangle$ , the CPHASE gate has no effect on the second qubit.

If the first qubit is in the state  $|1\rangle$ , the CPHASE gate applies a phase shift of  $-1$  to the second qubit, which corresponds to a rotation around the Z-axis of the Bloch sphere by an angle of  $\pi$ .

Often used in quantum algorithms to create and manipulate superpositions of quantum states

Performs quantum entanglement and quantum teleportation.

CPHASE Gate matrix

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix}$$

# Quantum Circuit

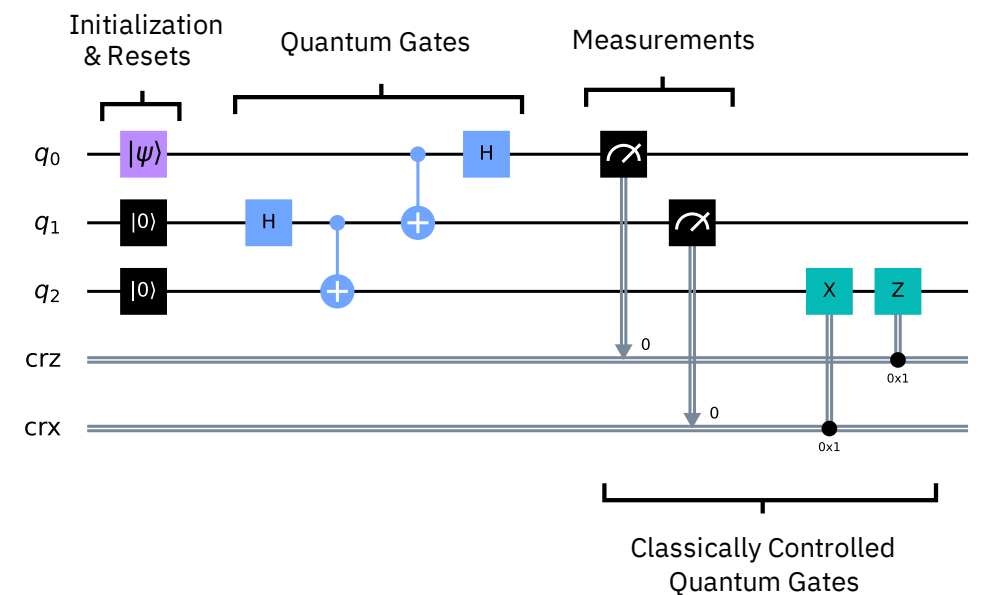
A quantum circuit is a mathematical model that represents the behavior of a quantum computer.

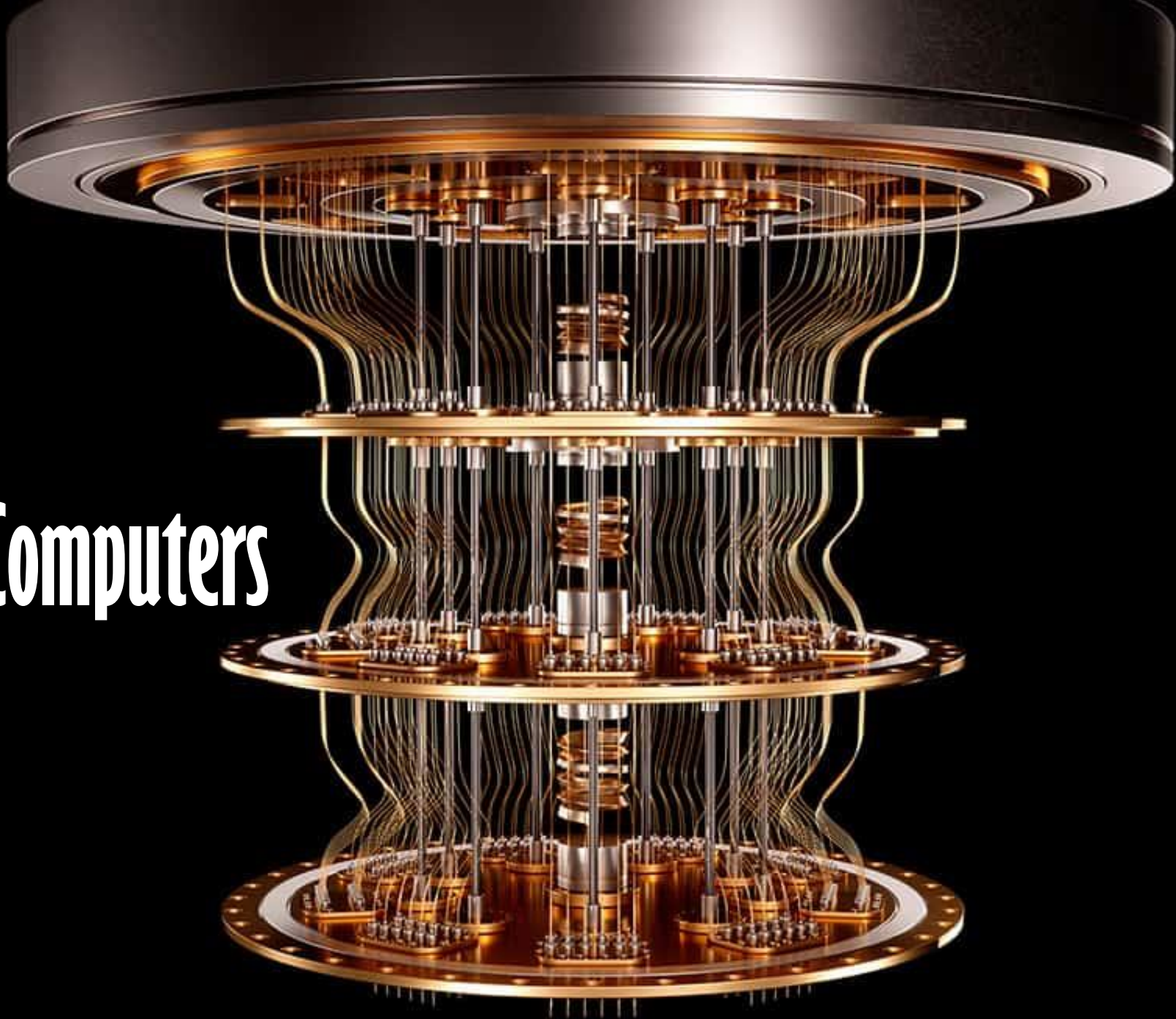
A quantum circuit consists of a series of quantum gates, which are mathematical operations that can be applied to qubits.

Qubits are represented as lines, and the gates are represented as boxes that operate on the qubits.

Each gate represents a specific operation that can be performed on one or more qubits, such as flipping the state of a qubit, entangling two or more qubits, or performing a measurement.

Quantum circuits are used to design and implement quantum algorithms.





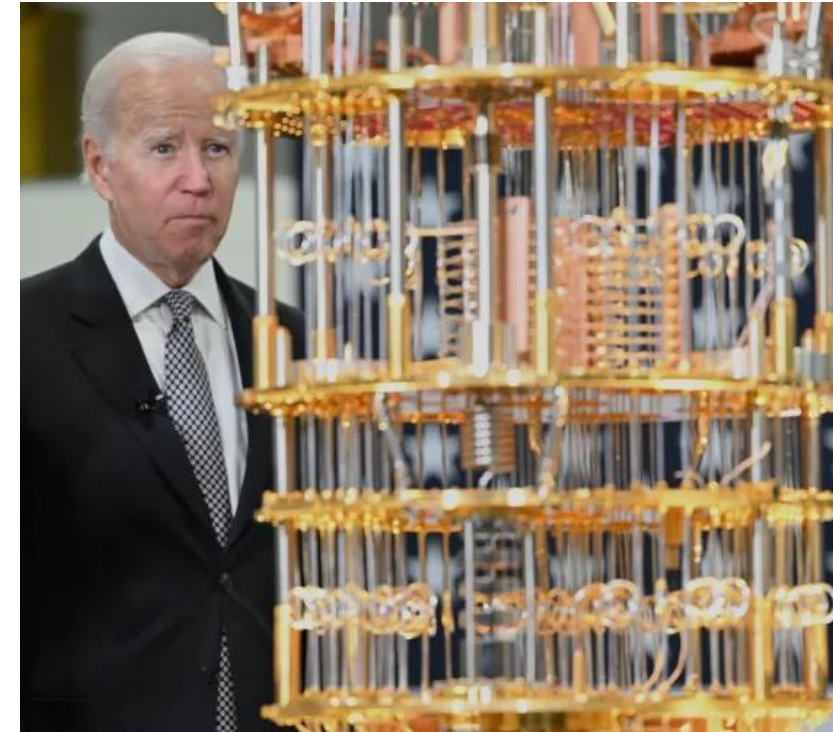
# Quantum Computers

# Quantum Computers

---

Quantum computers are typically made by assembling a set of physical components that can control and manipulate quantum states, which are the fundamental building blocks of quantum computing. These include:

1. Qubits: The basic unit of quantum information. Typically implemented using ions, superconducting circuits, or quantum dots.
2. Quantum Gates: basic Operations that can be performed on qubits to manipulate quantum states.
3. Control and Readout Electronics: Control the flow of information between the qubits and the outside world and allow for the measurement of quantum states.
4. Cryogenic Cooling System: Quantum systems are typically very sensitive to noise and decoherence: they need to be operated at very low temperatures, typically in the millikelvin range.





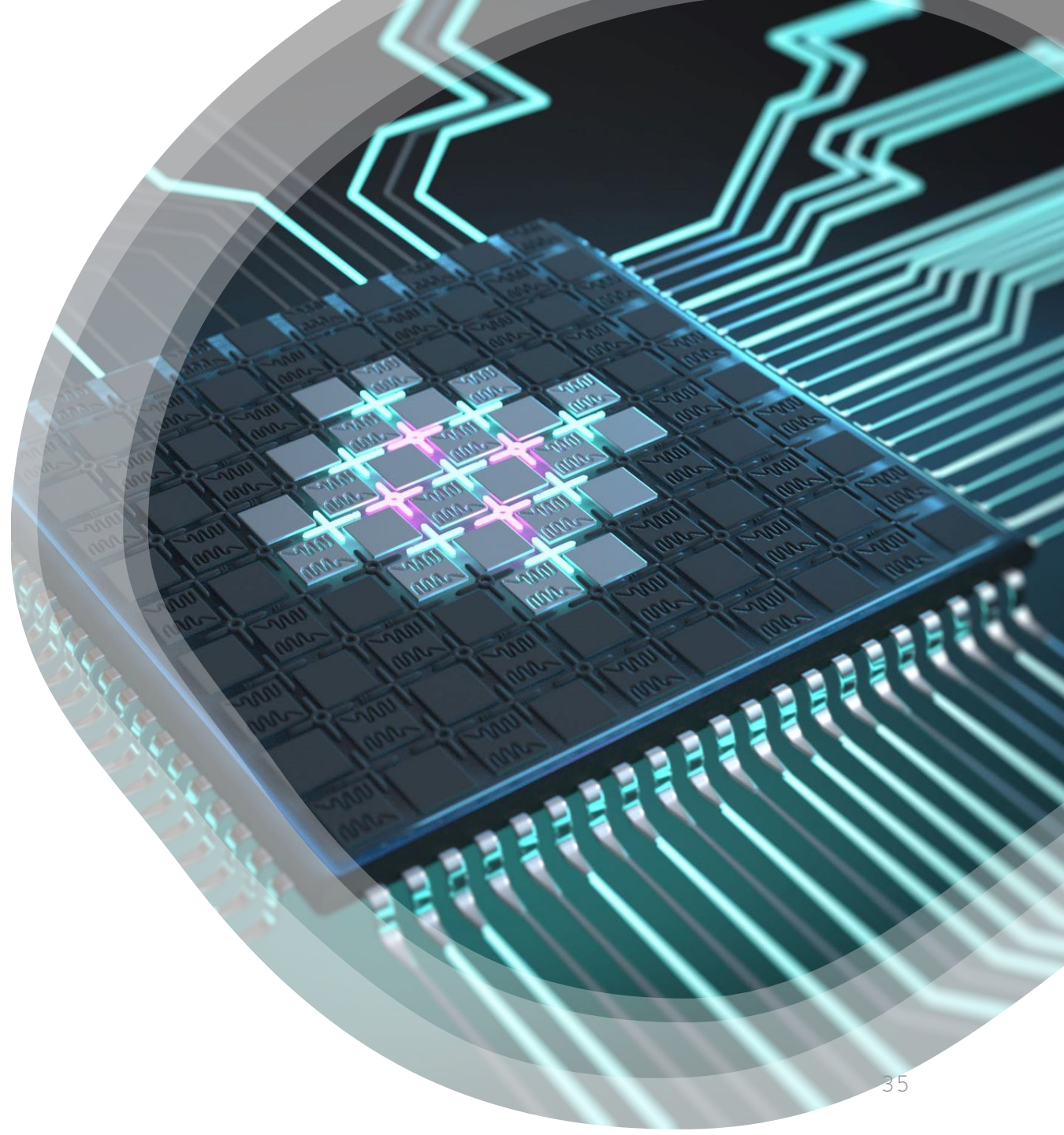
# Implementation of a quantum computer

The actual implementation of a quantum computer vary depending on the specific physical system used to realize the qubits.

Superconducting qubits are typically fabricated on a chip using thin-film deposition and lithography techniques

Ion trap qubits are implemented in vacuum chambers using lasers and other optical components.

Once the basic components are assembled, a quantum computer can be programmed using quantum algorithms, which take advantage of the unique properties of quantum systems, such as superposition, entanglement, and interference, to perform tasks that are difficult or impossible for classical computers.



# Companies working on quantum computers

---

Several companies have built quantum computers, each with their own approach to constructing and programming these complex machines.



**IBM** was one of the first companies to build a quantum computer and is a leader in the field. IBM developed a series of quantum computers, including the IBM Q System One, which is one of the most advanced ones for public use.



**Google** has developed a quantum computer called Sycamore, which is designed to solve specific problems. They have also developed a cloud-based platform called Cirq that allows developers to build and run quantum algorithms.



**Microsoft** has developed a quantum computer called Azure Quantum (not to confuse with the Canadian Company Azur Quantum), which is part of their Azure cloud computing platform. Microsoft has also developed a programming language called Q# that allows developers to write quantum algorithms.

# Companies working on quantum computers

---



**Honeywell** has developed a quantum computer called the System Model H1, which is designed to be one of the most powerful quantum computers in the world. They are also developing a cloud-based platform called Honeywell Forge that allows users to access their quantum computer remotely.



**Rigetti Computing** has developed a quantum computer called the Aspen-8, which is designed to be one of the most powerful quantum computers for public use. They have also developed a cloud-based platform called Forest that allows developers to build and run quantum algorithms.



**IonQ** has developed a quantum computer based on trapped ions, which is designed to be one of the most stable and reliable quantum computers. They have also developed a cloud-based platform called IonQ Cloud that allows users to access their quantum computer remotely.



**D-Wave Systems** is a Canadian company based in British Columbia. It is one of the first companies to sell computers that exploit quantum effects using quantum annealing, designed to solve optimization problems.



**Intel** is leveraging its expertise in high-volume transistor manufacturing to develop 'hot' silicon spin-qubits, much smaller computing devices that operate at higher temperatures.

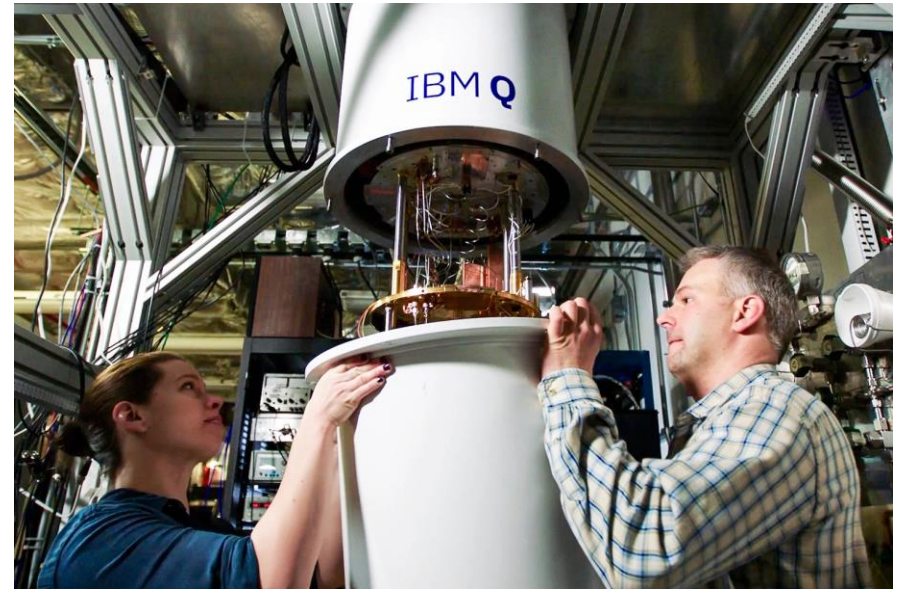
# The IBM Q quantum computer

---

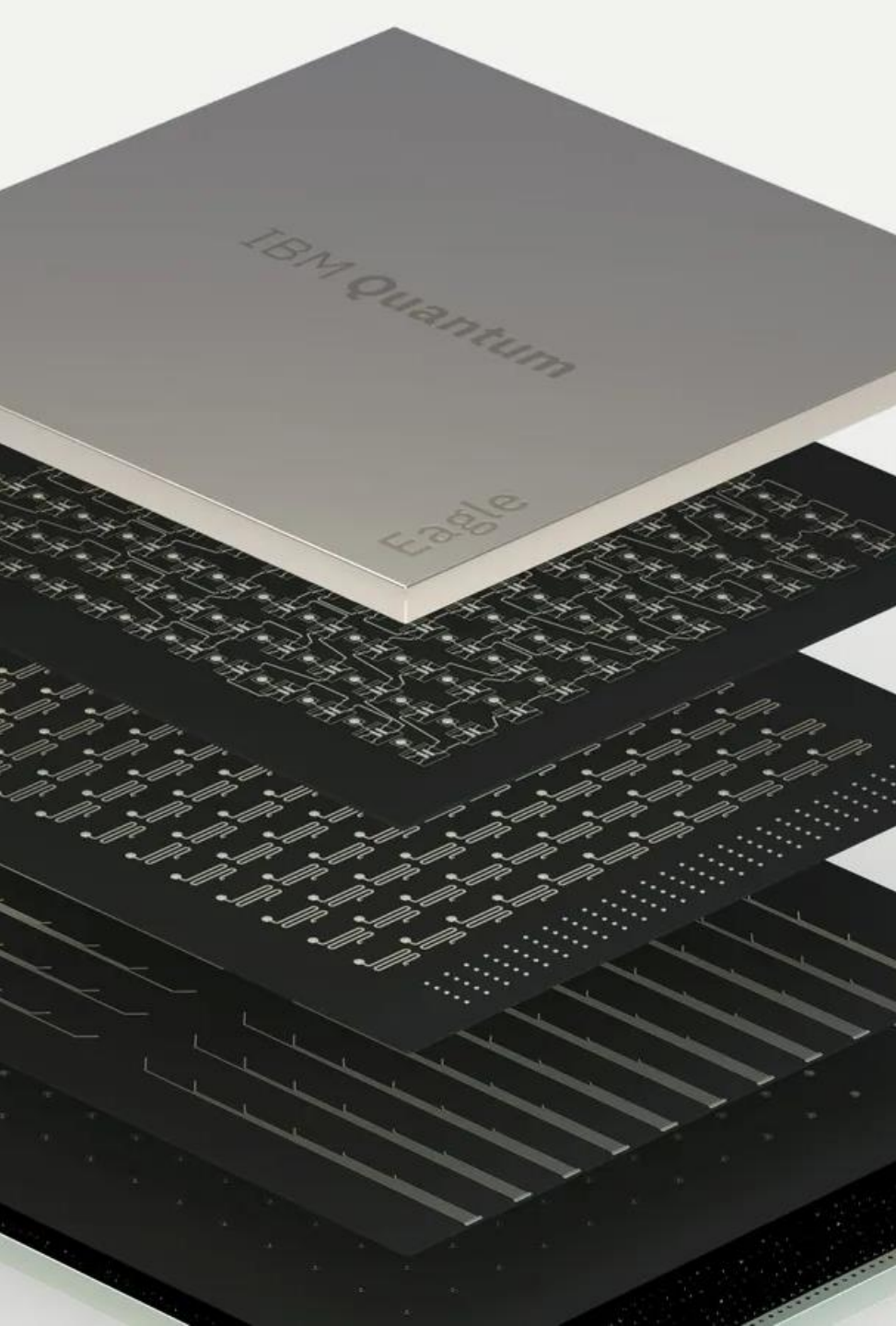
IBM has been a major player in the field of quantum computing and has developed several generations of quantum computers over the years.

IBM's current quantum computing platform is called IBM Quantum, which is a cloud-based platform that provides access to quantum computers over the internet.

**IBM Q™**







# IBM Quantum

---

IBM Quantum is based on superconducting qubits, which are implemented using superconducting circuits that are cooled to extremely low temperatures.

The current generation of IBM Quantum devices support up to 65 qubits and are designed to be scalable, with the goal of eventually building a quantum computer with hundreds or even thousands of qubits.

IBM Quantum provides users with access to a variety of software tools, including a web-based interface called the IBM Quantum Composer, which allows users to design and run quantum circuits using a drag-and-drop interface.

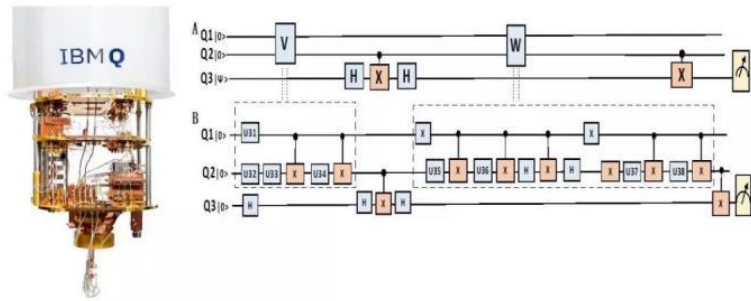
IBM Quantum also provides users with a software development kit (SDK) called Qiskit, which is a collection of open-source software tools for programming and simulating quantum circuits.



# IBM Q Composer

The screenshot displays the IBM Q Composer interface. On the left is a sidebar with navigation options: 'Composer docs & tutorials' (with a 'Full docs' link), 'Getting started' (with sub-items like 'Create your first circuit walkthrough'), 'Composer features' (with sub-items like 'Use operations & gates'), 'Quantum Composer user guide', 'Learn quantum computing: a field guide', 'Try out some circuit examples', 'IBM Quantum compute resources', 'Glossaries', and 'How to cite'. The main workspace shows a quantum circuit with three qubits: q[0], q[1], and c2. q[0] has an H gate, followed by a CNOT with q[1] as control and q[0] as target. q[1] has a CNOT with q[0] as control and q[1] as target, followed by a Z gate. Both q[0] and q[1] have Z gates. The c2 register is measured. Below the circuit is a 'Probabilities' section with a bar chart showing 50% probability for states 00 and 11. To the right is a 'Q-sphere' visualization showing the state vector pointing to the |00> state. The top right contains a search bar, help icon, user icon, and a 'Setup and run' button. The bottom right shows the OpenQASM 2.0 code for the circuit.

```
1 OPENQASM 2.0;
2 include "qelib1.inc";
3 qreg q[2];
4 creg c[2];
5 h q[0];
6 cx q[0],q[1];
7 measure q[0] -> c[0];
8 measure q[1] -> c[1];
```



## IBM Quantum is cloud-based

IBM Quantum is cloud-based, which allows users to access their quantum computers from anywhere in the world using a standard web browser.

IBM provides a range of resources and tutorials to help users get started with quantum computing, including access to a community of researchers and developers who are working on cutting-edge quantum algorithms and applications.

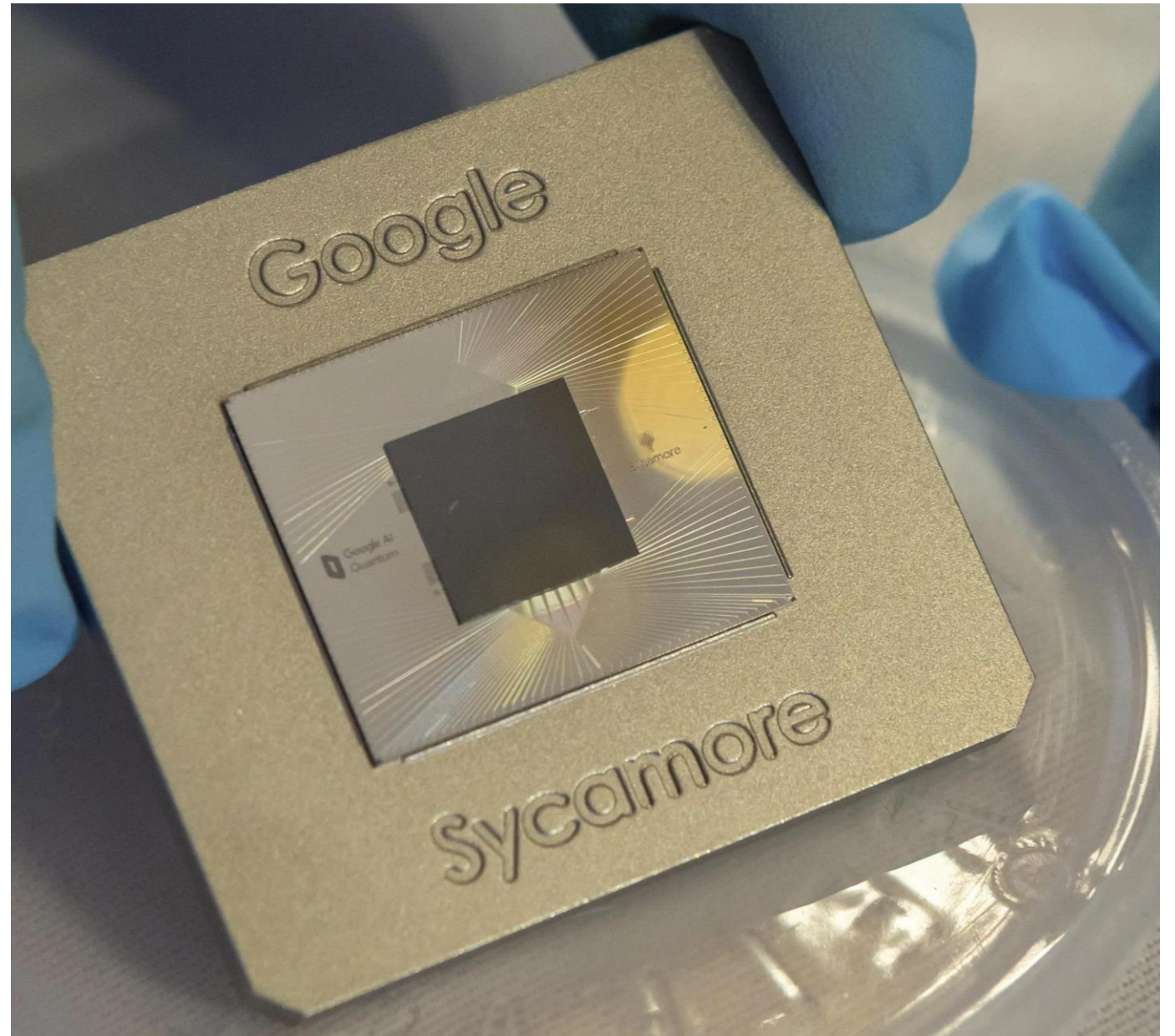
IBM Quantum is involved in several collaborations and partnerships with academic institutions and industry partners to advance the field of quantum computing and develop practical applications of this emerging technology.

# The Google Quantum Computer

Google's quantum computer is called Sycamore

Has a 54-qubit quantum processor designed to perform quantum computations faster than classical computers for *specific problems*.

The processor consists of a chip containing superconducting qubits that are kept at ultra-low temperatures to maintain their quantum states.





# The random sampling problem

---

In October 2019, a team of researchers at Google announced that they had achieved “quantum supremacy” with Sycamore: the quantum computer could solve a specific problem faster than the world's most powerful classical supercomputers.

Sycamore could perform a specific quantum calculation in 200 seconds, whereas the same calculation would take the world's most powerful supercomputer, Summit, over 10,000 years to solve.

The specific problem solved by Sycamore is *random sampling* where the goal is to generate a set of random numbers with a specific probability distribution.

This type of problem is relevant in cryptography, finance, and materials science.



Sundar Pichai and Daniel Sank with a Google quantum computer



## The quantum supremacy

Google's achievement of quantum supremacy is a significant milestone in the development of quantum computing.

It demonstrates that quantum computers can perform computations that are infeasible with classical computers.

The calculation that Sycamore performed was highly specific.

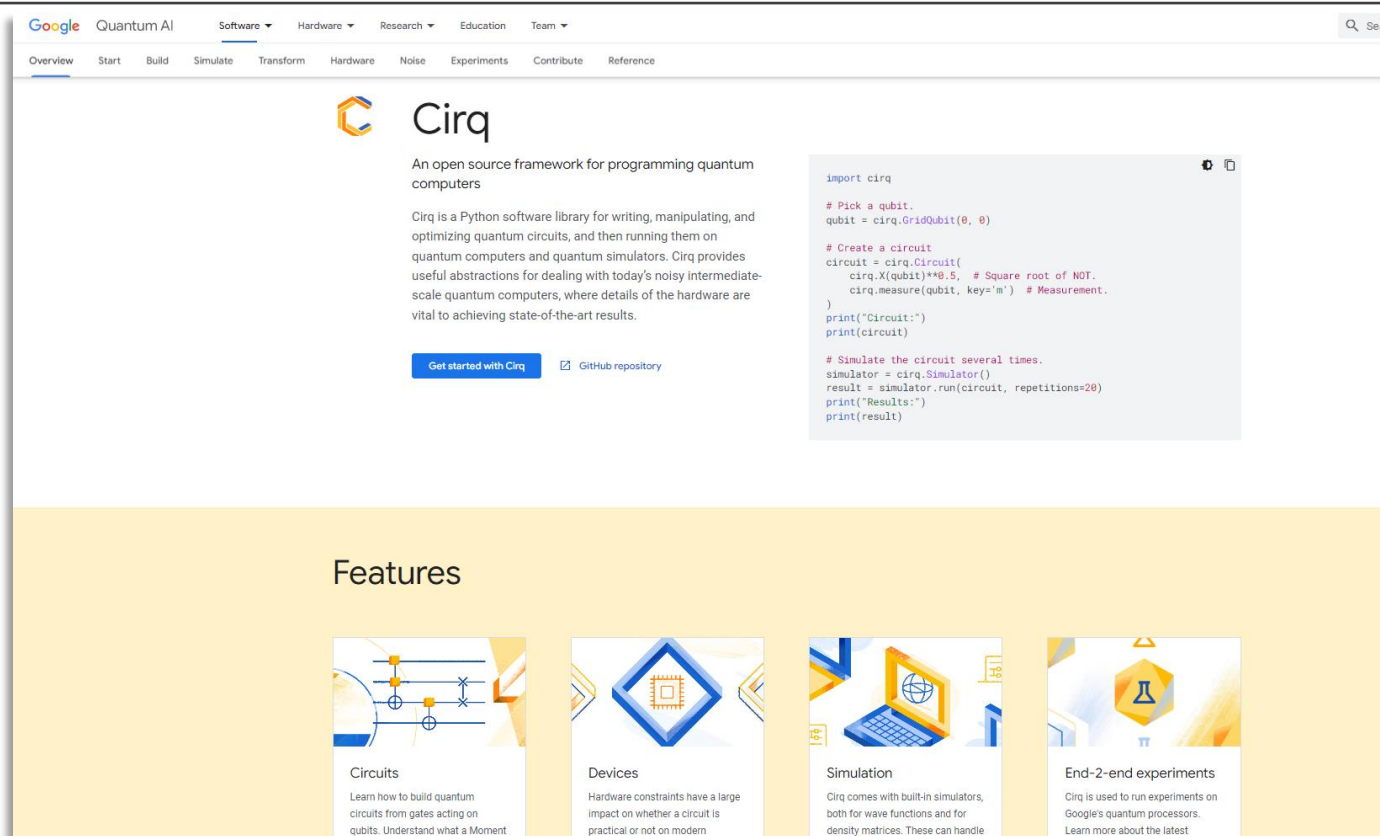
Google announced plans to develop a 1,000-qubit quantum computer, which would be a significant advance in the field of quantum computing.



Sycamore is available for use through the cloud-based platform: **Cirq**.

Cirq is a Python library for writing, manipulating, and optimizing quantum circuits and running them against quantum computers and simulators.

<https://github.com/quantumlib/Cirq>



The screenshot shows the Cirq website interface. At the top, there's a navigation bar with "Google Quantum AI" and various menu items like "Software", "Hardware", "Research", "Education", and "Team". Below the navigation, the main heading "Cirq" is displayed with a sub-heading "An open source framework for programming quantum computers". A brief description follows, stating that Cirq is a Python software library for writing, manipulating, and optimizing quantum circuits, and then running them on quantum computers and quantum simulators. Below this, there are two buttons: "Get started with Cirq" and "GitHub repository". To the right, a code block shows a Python script for creating and simulating a quantum circuit. The code includes comments and uses the Cirq library to create a qubit, define a circuit with an X gate and a measurement, and then simulate it.

```
import cirq

# Pick a qubit.
qubit = cirq.GridQubit(0, 0)

# Create a circuit
circuit = cirq.Circuit(
    cirq.X(qubit)**0.5, # Square root of NOT.
    cirq.measure(qubit, key='m') # Measurement.
)
print("Circuit:")
print(circuit)

# Simulate the circuit several times.
simulator = cirq.Simulator()
result = simulator.run(circuit, repetitions=20)
print("Results:")
print(result)
```

Below the main content, there is a "Features" section with four cards:

- Circuits**: Learn how to build quantum circuits from gates acting on qubits. Understand what a Moment
- Devices**: Hardware constraints have a large impact on whether a circuit is practical or not on modern
- Simulation**: Cirq comes with built-in simulators, both for wave functions and for density matrices. These can handle
- End-2-end experiments**: Cirq is used to run experiments on Google's quantum processors. Learn more about the latest



# Microsoft Azure Quantum

---

Microsoft Azure Quantum is a cloud-based platform that allows customers to access a variety of quantum computing technologies from Microsoft and its partners.

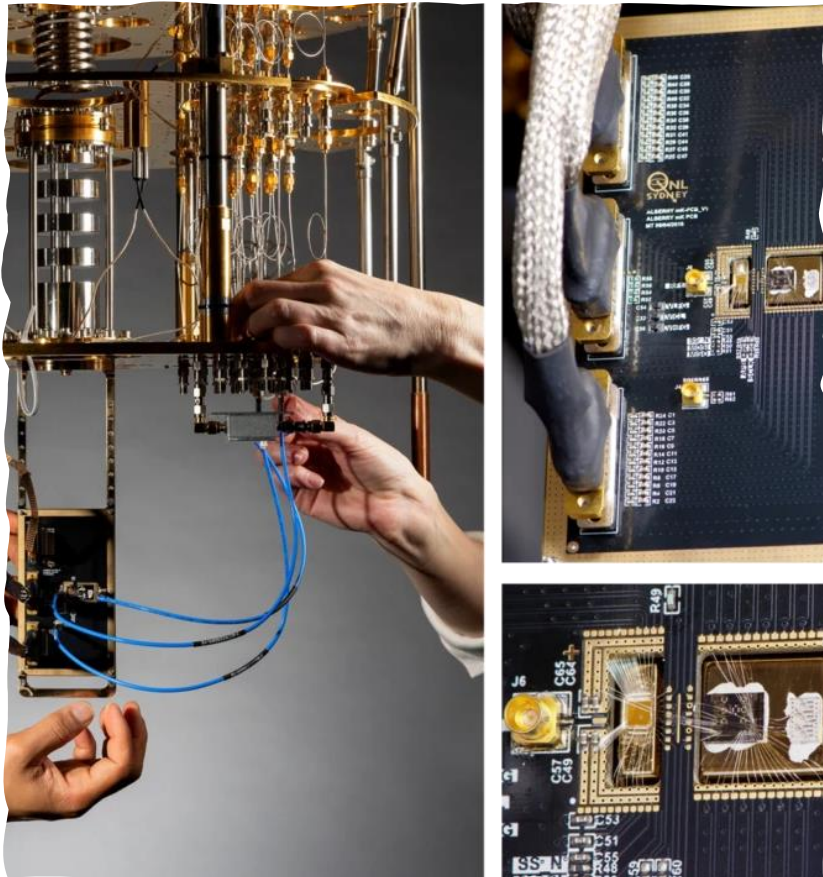
The platform is intended to provide customers with a way to experiment with and develop quantum applications, without having to invest in expensive quantum hardware themselves.

Azure Quantum is designed to be an open platform that supports a variety of quantum programming languages and development tools, including Q#, Python, and Microsoft's Quantum Development Kit.

Customers can use Azure Quantum to access gate-based quantum computers, quantum annealers, and quantum-inspired classical computing.

Microsoft Azure Quantum aims to democratize access to quantum computing technology and accelerate the development of practical quantum applications.

# Unique features of Azure Quantum



One of the unique features of Azure Quantum is the ability to seamlessly integrate quantum computing with classical computing.

Customers can use Azure Quantum to run hybrid quantum-classical algorithms, where part of the algorithm is executed on a quantum computer, and the remainder is executed on a classical computer.

Azure Quantum also offers a variety of tools and services to help customers develop and optimize quantum applications, including a quantum simulator for testing and debugging, and access to Microsoft's Quantum-inspired optimization service for solving optimization problems.



# Noisy Intermediate-Scale Quantum

Noisy Intermediate-Scale Quantum (NISQ) designates quantum computing devices that are currently available and can be used for practical applications.

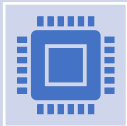
These devices are called “noisy” because they have a relatively high error rate compared to ideal quantum computers, and “intermediate-scale” because they have a limited number of qubits, typically ranging from a few dozen to a few hundred.

NISQ devices are not yet powerful enough to solve some of the most complex problems that quantum computers are theorized to be able to solve, but they are still useful for a variety of applications, including cryptography, optimization, and machine learning.



# NISQ devices

---



NISQ devices are being developed and improved by companies such as IBM, Google, and Microsoft, as well as startups such as Rigetti Computing and IonQ.



These companies are working to improve the performance and reliability of NISQ devices, as well as developing software tools and applications that can run on these machines.



NISQ represents an exciting area of research and development in quantum computing, with the potential to have a significant impact on a wide range of industries and fields.





# Qiskit (<https://qiskit.org/>)

---

Qiskit is an open-source SDK developed by IBM for working with quantum computers.

Provides tools for developing and running quantum computing experiments, including simulators and access to real quantum devices through IBM's cloud service.

Qiskit provides a Python-based interface for programming quantum circuits, which are the basic building blocks for quantum algorithms.

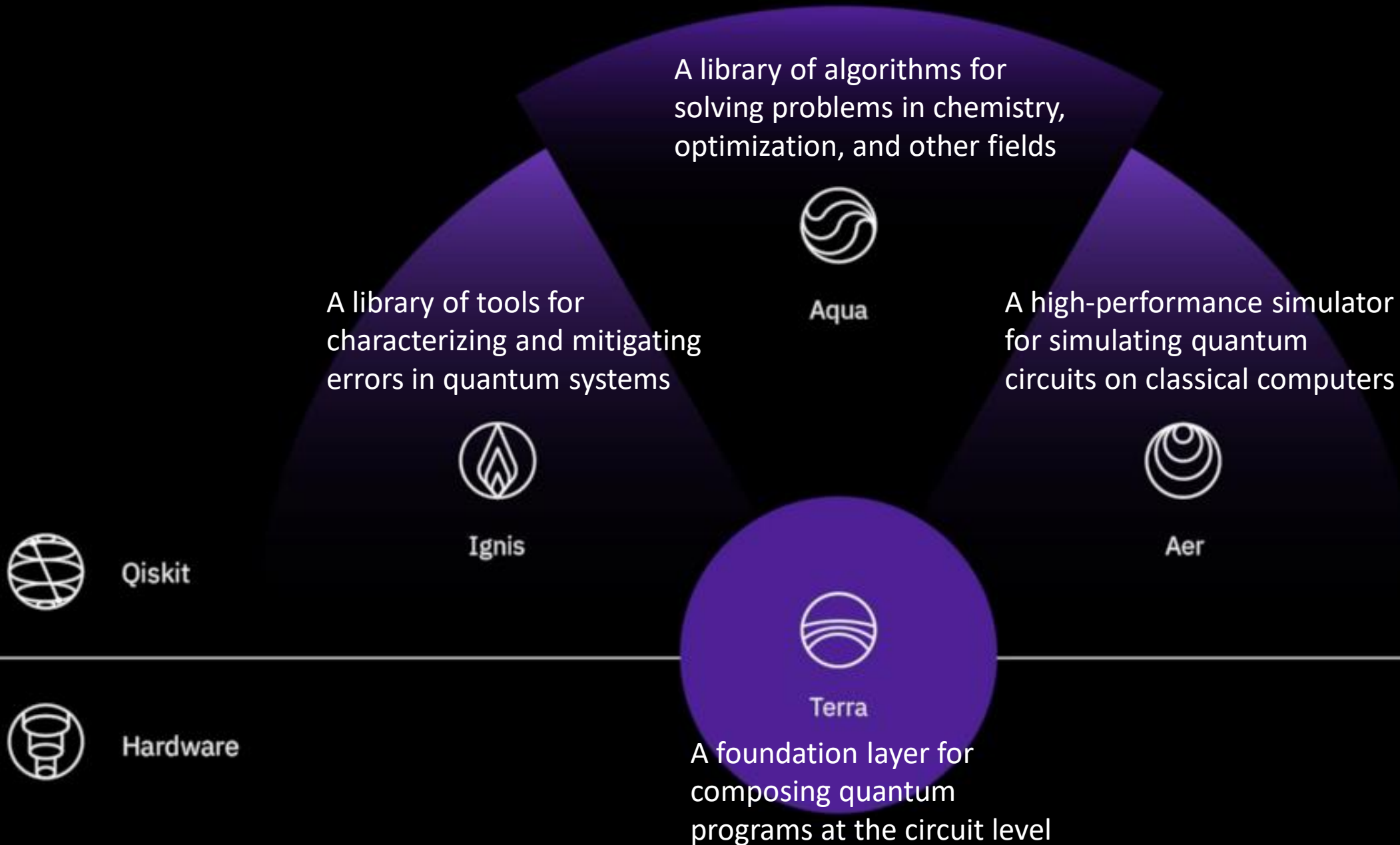
The SDK includes a variety of tools and libraries for quantum circuit design, simulation, and optimization, as well as for accessing and controlling real quantum devices.

Qiskit is widely used by researchers and developers in the quantum computing community, and it has become one of the most popular quantum SDK.

It is designed to be accessible to both experienced researchers and those new to quantum computing, with extensive documentation, tutorials, and community support available.



# Key components of Qiskit



# Quantum Computing and Cybersecurity





## Quantum computing impact on cybersecurity

Quantum computing has the potential to significantly impact cybersecurity, both in terms of the vulnerabilities it may expose and the new security mechanisms it may enable.

Quantum computing poses a threat to traditional cryptographic systems.

Many modern cryptographic protocols rely on the difficulty of certain mathematical problems, such as factoring large numbers, for their security.

Quantum computers have the potential to solve these problems much faster than classical computers, rendering many existing cryptographic systems vulnerable to attacks.

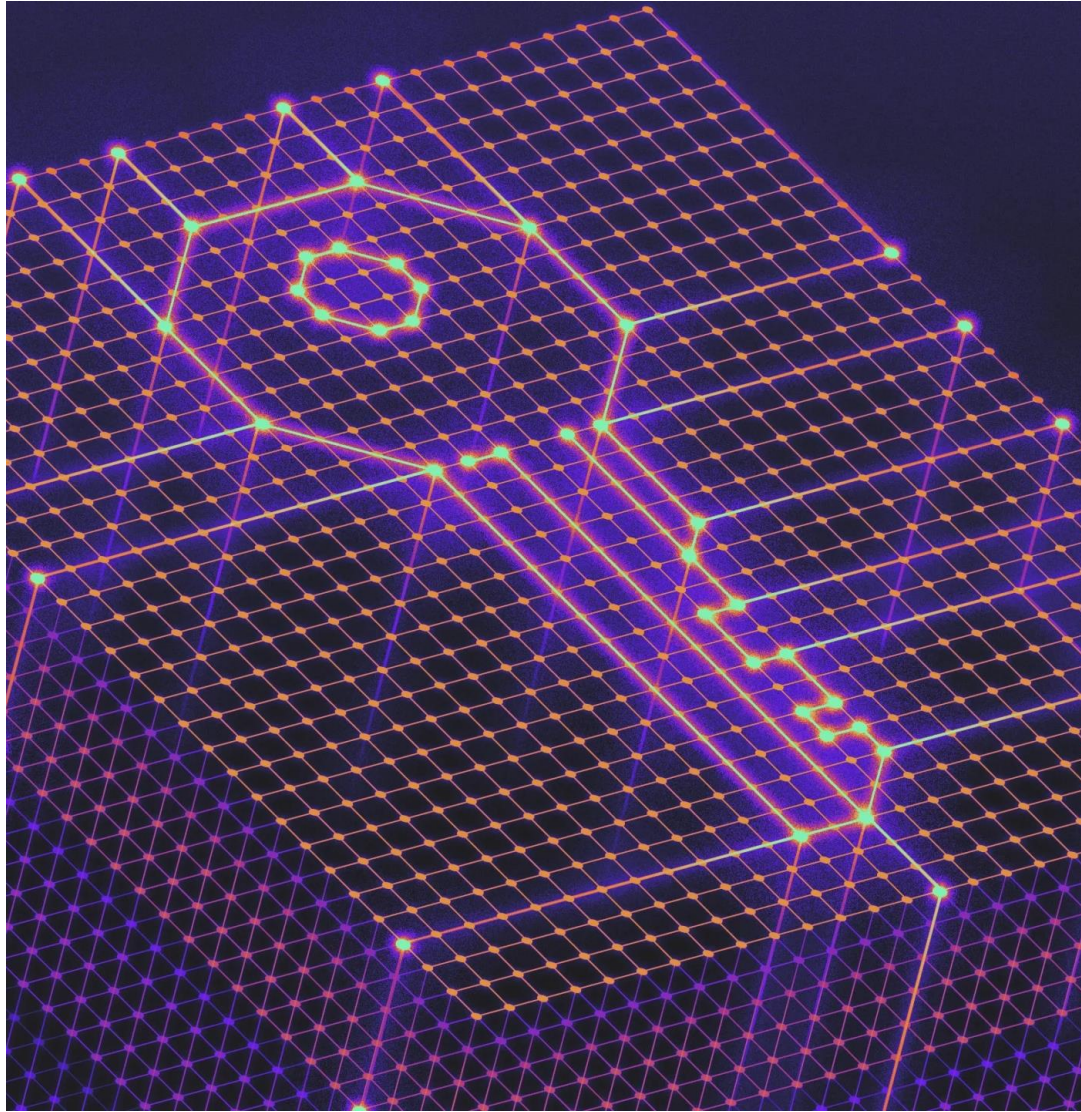




# POST-QUANTUM CRYPTOGRAPHY

To address the threats, researchers are exploring the development of “Post-Quantum Cryptography” (PQC) that is resistant to attacks by quantum computers.

PQC algorithms rely on different mathematical problems that are believed to be hard for both classical and quantum computers, making them potentially more secure in the era of quantum computing.



# Approaches to PQC

**Lattice-based cryptography:** use mathematical lattices to perform encryption and decryption. Resistant to attacks by quantum computers because the algorithms that quantum computers can use to solve lattice problems do not provide a significant speedup over classical algorithms.

**Code-based cryptography:** use error-correcting codes to perform encryption and decryption. Resistant to attacks by quantum computers because the algorithms that quantum computers can use to break these codes require large amounts of memory, which is a resource difficult to efficiently utilize by quantum computers.

Other approaches to PQC include **hash-based cryptography**, **isogeny-based cryptography**, and **multivariate cryptography**, among others.

These approaches are still being actively researched and developed, and it is not yet clear which ones will ultimately prove to be the most effective for PQC.



# Quantum Key Distribution

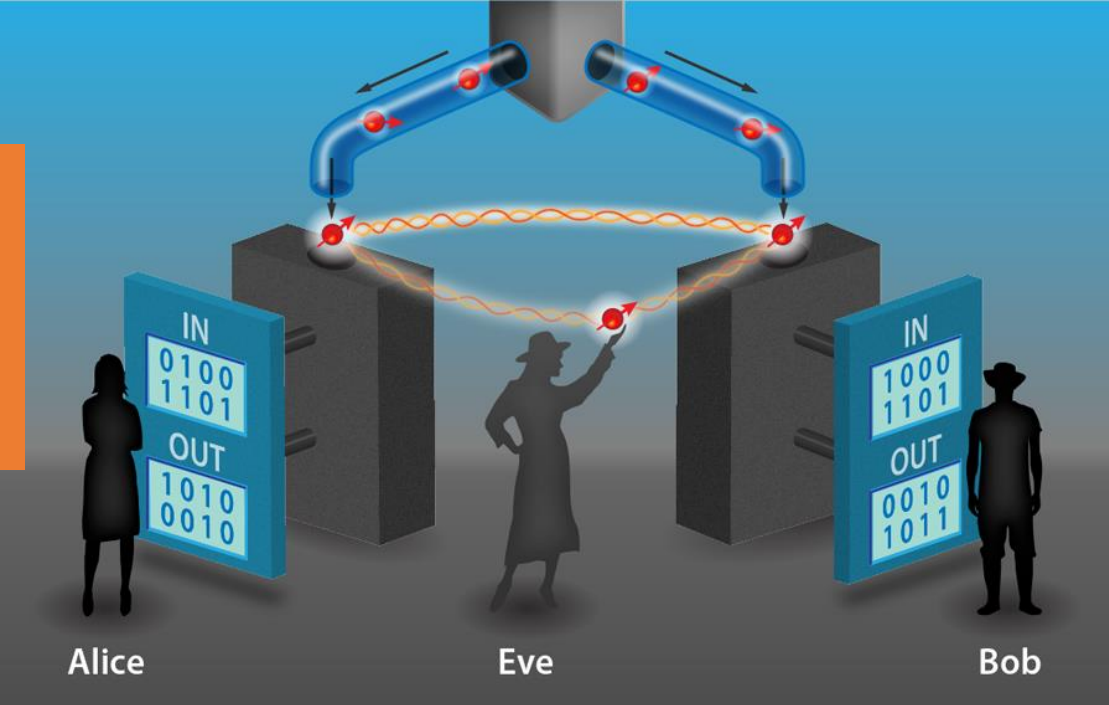
---



Quantum Key Distribution (QKD) allows two parties to securely establish a shared secret key using the principles of quantum mechanics.

QKD provides a level of security that is impossible to achieve with classical cryptographic protocols.

Quantum computing can leverage quantum entanglement to securely transmit keys.



# Basic idea behind QKD

Use the properties of quantum states to enable two parties to establish a shared secret key without the risk of eavesdropping.

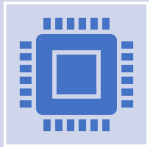
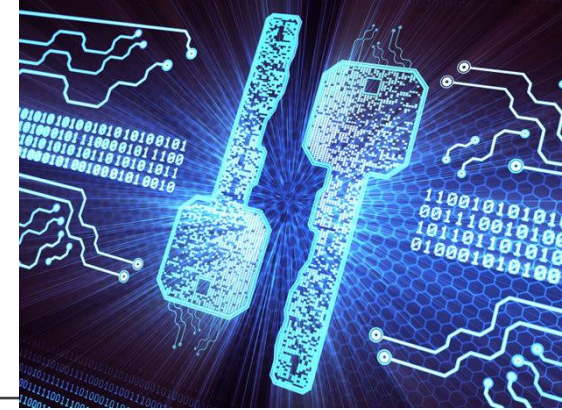
The two parties, often referred to as Alice and Bob, exchange quantum states over a quantum communication channel.

The properties of these quantum states, such as their polarization or phase, are used to encode the key.

Any attempt by an eavesdropper, often called Eve, to intercept or measure the quantum states would result in a disturbance that can be detected by Alice and Bob, thereby alerting them to the presence of an eavesdropper and allowing them to discard the affected key.

# QKD protocols

---



There are a variety of QKD protocols, but they generally involve using quantum states to encode information that is used to establish a shared secret key between Alice and Bob.



Once the key is established, it can be used to encrypt and decrypt messages using traditional cryptographic protocols.

# Popular QKD protocols



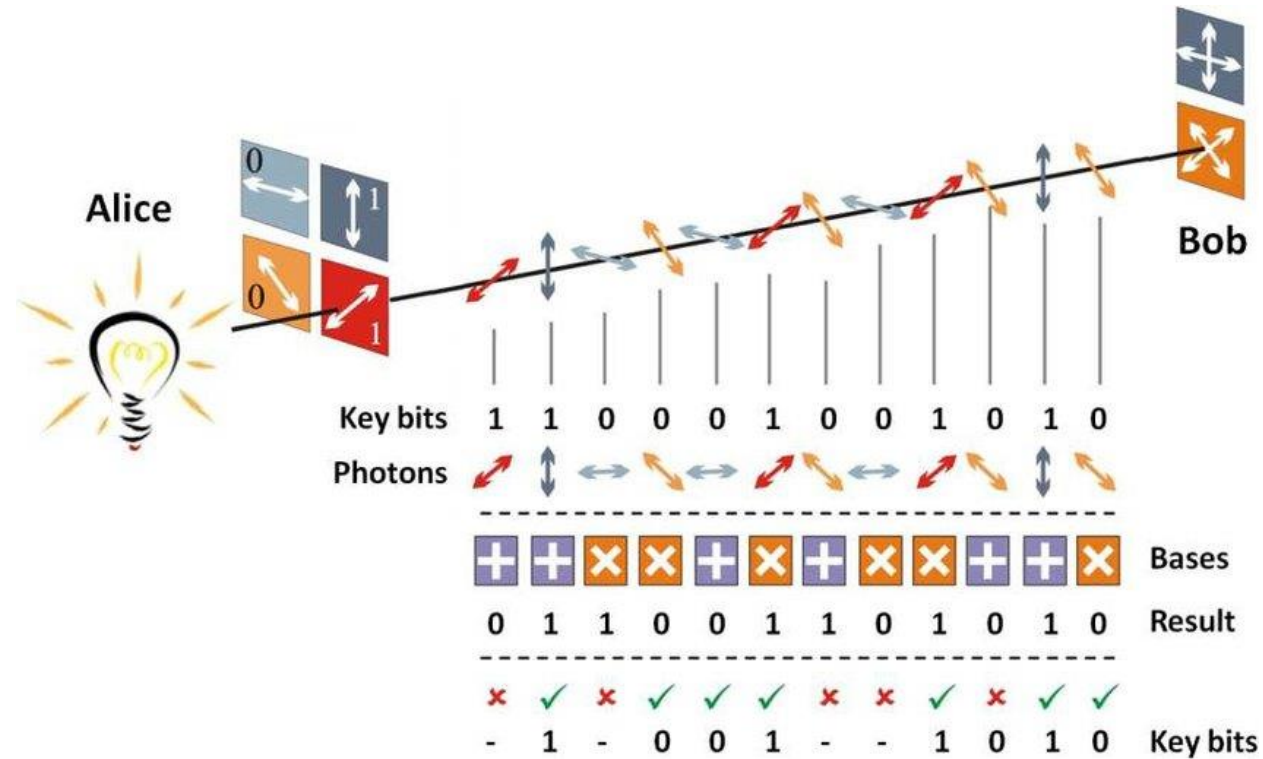
**BB84 protocol** (Bennett and Brassard, 1984): One of the earliest and widely used. Involves the transmission of single photons over a quantum channel, with the polarization of the photon representing the bit value. Uses a randomized basis to encode the key. Transmission errors are detected using the same basis.

**E91 protocol**, also known as the entanglement-based protocol (Ekert, 1991): Involves the creation and distribution of entangled photon pairs over a quantum channel. The polarization of one photon in each pair is then used to encode the key. Used to detect eavesdropping.

**SARG04 protocol** (Pirandola, Mancini, Lloyd, and Braunstein, 2004): Uses coherent states of light to transmit the key over a quantum channel. The key is encoded in the phase of the coherent states. Transmission errors are detected using homodyne detection.

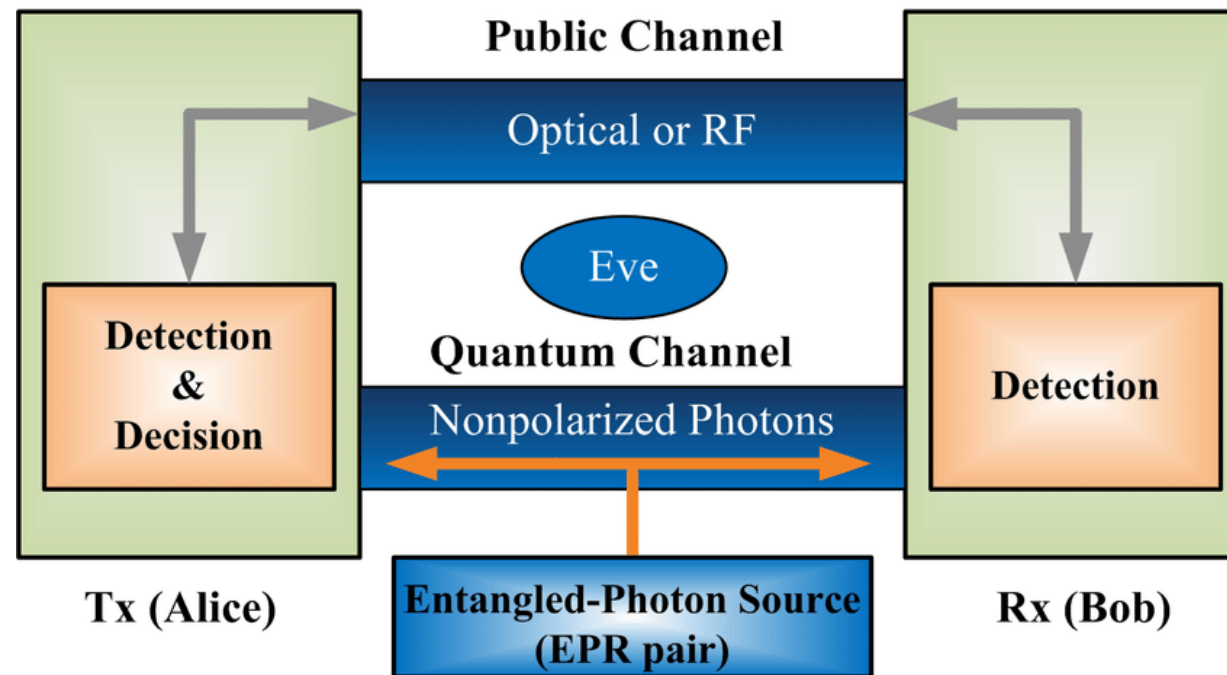
**B92 protocol** proposed (Bennett, 1992): Uses two non-orthogonal states to transmit the key over a quantum channel. The key is encoded in the state of the photon. Transmission errors are detected by comparing the expected value of the states with the actual measured value.

# BB84 protocol





# E91 protocol



N. Alshaer, A. Moawad and T. Ismail, "Reliability and Security Analysis of an Entanglement-Based QKD Protocol in a Dynamic Ground-to-UAV FSO Communications System," in *IEEE Access*, vol. 9, pp. 168052-168067, 2021, doi: 10.1109/ACCESS.2021.3137357.

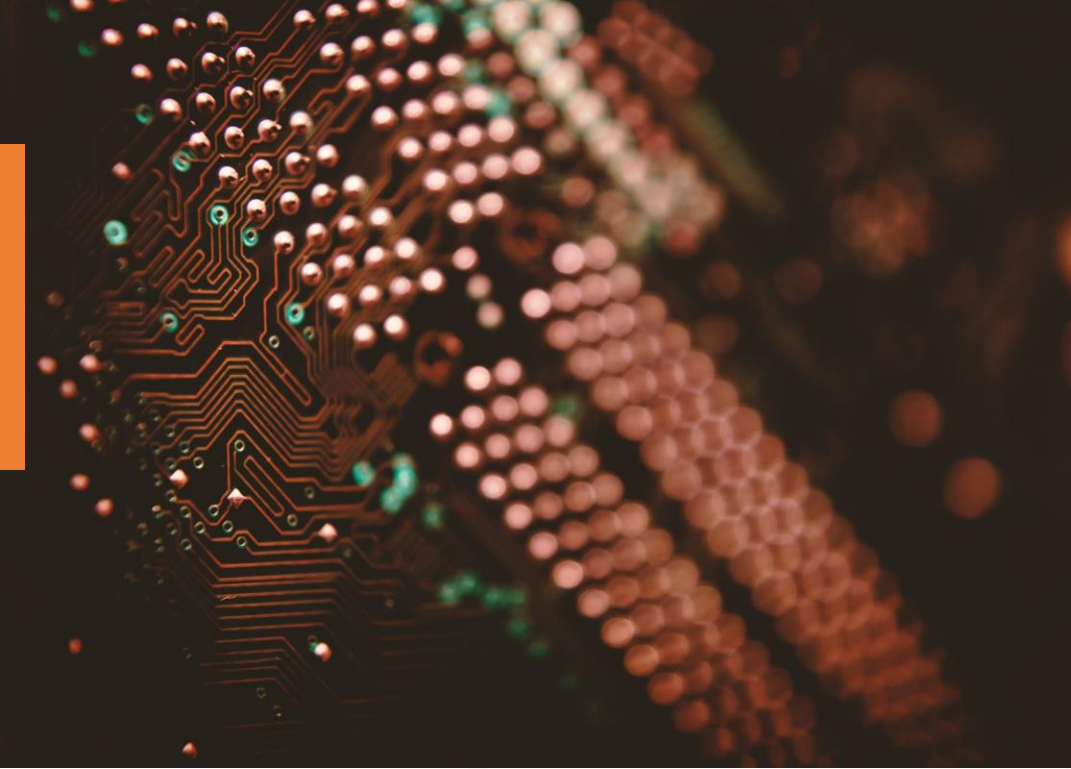


## QKD level of security

QKD has the potential to provide a level of security that is impossible to achieve with classical cryptographic protocols.

This is because any attempt to intercept or measure the quantum states would result in a disturbance that can be detected by Alice and Bob, thereby alerting them to the presence of an eavesdropper.

As a result, QKD can provide a level of security that is independent of the computational power of an adversary, making it a promising technology for secure communication in the era of quantum computing.



# Challenges associated with QKD

**Practical implementation:** QKD requires specialized hardware that can be expensive and difficult to operate.

**Quality of the quantum communication:** Typically limited to relatively short distances : It is difficult to maintain the quality of the quantum communication channel over long distances: Noise and loss, can affect the performance. Typically limited to line-of-sight transmission. Sensitive to environmental factors, such as temperature and vibration. May not be suitable for fiber optics cables.

**Key Rate:** The rate at which secure keys can be generated and renewed. QKD systems typically generate keys at a much lower rate than classical methods..



# More challenges associated with QKD

**Security assumptions:** QKD is based on some fundamental assumptions such as the no-cloning theorem and the uncertainty principle. While these assumptions are well-established, they are not immune to attacks.

**Key management:** QKD generates a shared secret keys between two parties, but these keys must be securely stored and managed to prevent unauthorized access. Key management can be complicated, and it requires careful attention to ensure that the keys remain secure over time.



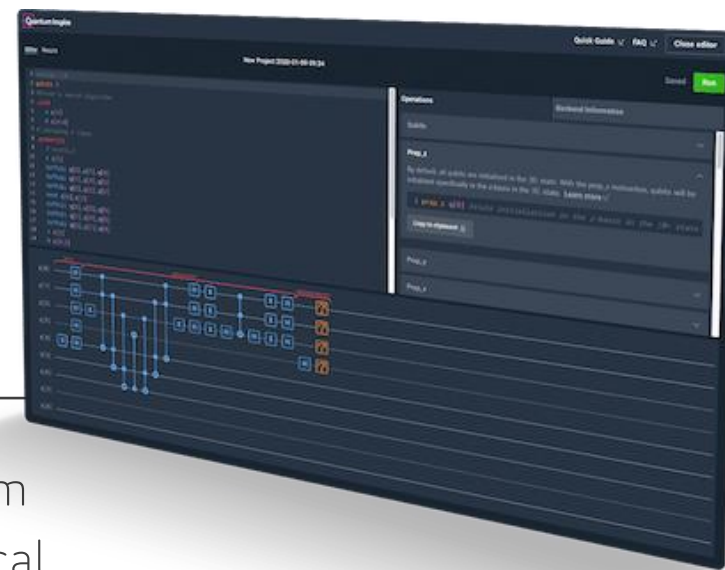
# Quantum algorithms

---

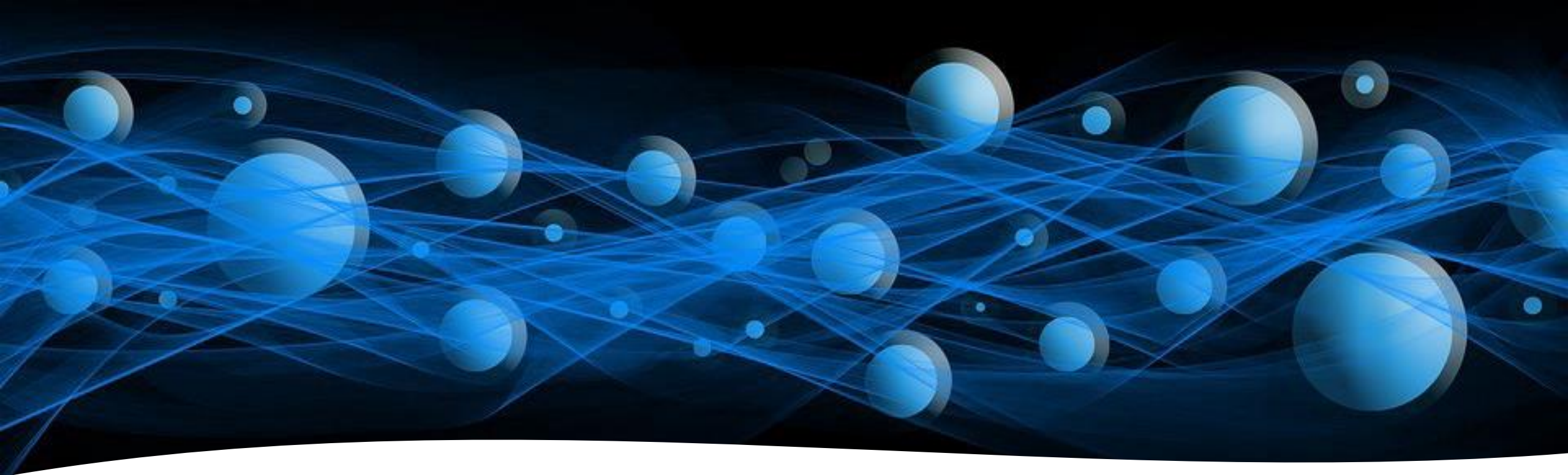
Quantum algorithms take advantage of the unique properties of quantum mechanics to solve computational problems more efficiently than classical algorithms. Some examples of well-known quantum algorithms include:

**Grover's Algorithm** for searching an unsorted database. It provides a quadratic speedup over the best classical algorithm, allowing the search to be performed in  $O(\sqrt{N})$  time instead of  $O(N)$  time.

**Shor's Algorithm** for integer factorization. It can factor an  $N$ -digit integer in  $O((\log N)^3)$  time, whereas the best-known classical algorithm takes exponential time in the number of digits  $N$ .





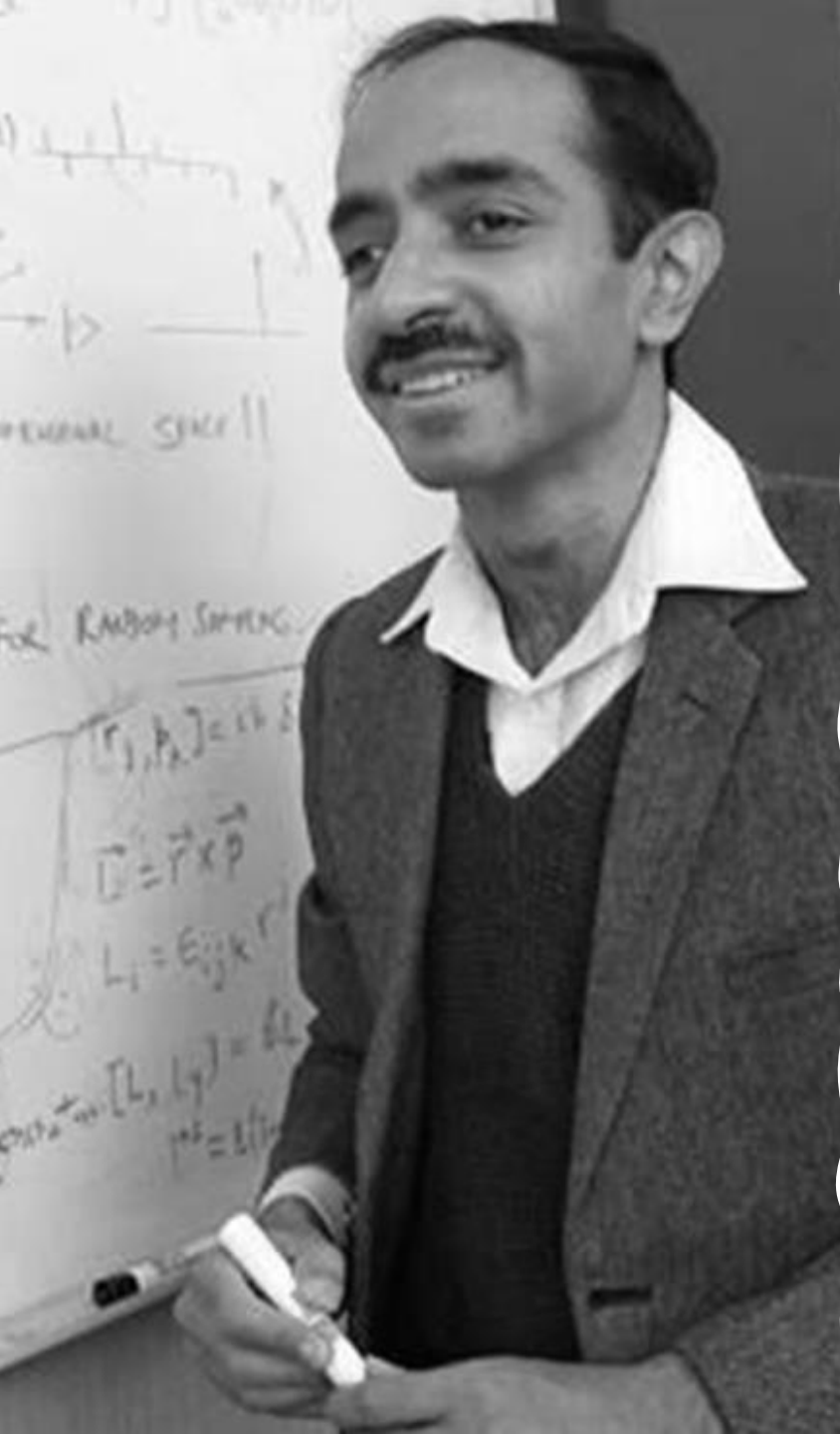


## Other Quantum Algorithms

**Quantum Walk:** quantum analogs of classical random walks. They can be used for a variety of tasks, including searching, element distinctness, and graph problems.

**Quantum Phase Estimation** for estimating the phase of an eigenvector of a unitary operator. It is an important subroutine in several quantum algorithms, including Shor's algorithm and for solving linear systems of equations.

**HHL Algorithm** (Harrow-Hassidim-Lloyd) for solving linear systems of equations. It provides an exponential speedup over the best-known classical algorithm for certain types of matrices.



# Grover's Algorithm

---

Invented by Lov Grover in 1996 and is one of the most well-known quantum algorithms.

Grover's algorithm can be used to search an unsorted database of  $N=2^n$  items in  $O(\sqrt{N})$  time, which is quadratically faster than the best known classical algorithm's  $O(N)$  time complexity.

The algorithm works by creating a superposition of all possible states and applying a unitary operator that reflects the amplitude of the state about the mean value.

This operator increases the probability of measuring the desired state and decreases the probabilities of measuring the other states.

By repeating this process multiple times, the probability of measuring the desired state increases and the probability of measuring the other states decreases.

Eventually, the desired state can be found with high probability.

# The task

The task that Grover's algorithm aims to solve can be expressed as follows: given a classical function  $f(x):\{0,1\}^n \rightarrow \{0,1\}$ , where  $n$  is the bit-size of the search space, find an input  $x_0$  such that  $f(x_0)=1$ .

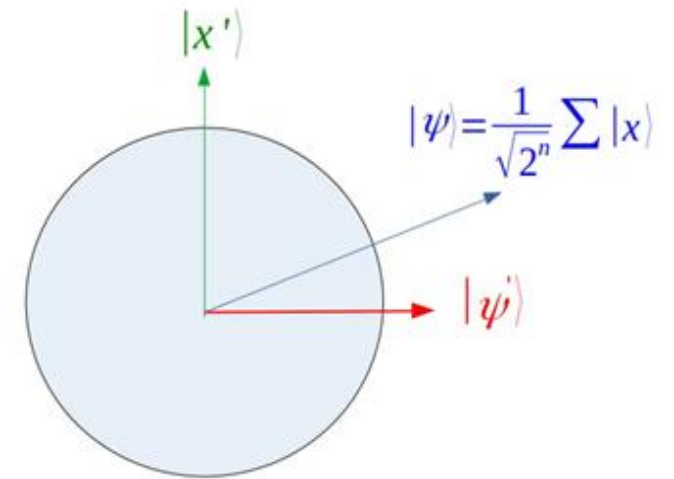
The idea is to use an oracle (black box) which can *recognize* the solution to the search problem and this recognition is signaled by making use of an oracle qubit.

In the worst-case scenario, we must evaluate  $f(x)$  a total of  $N-1$  times trying out all the possibilities. After  $N-1$  elements, we know it must be the remaining element.

# High-level description of Grover's algorithm

---

1. Prepare a quantum state  $|s\rangle = (1/\sqrt{N}) \sum |x\rangle$ , where  $x$  ranges over all possible states in the database.
2. Apply the Grover iteration operator  $G = D.W.D^{-1}$ , where  $D$  is the diffusion operator and  $W$  is the oracle operator.
3. Repeat step 2  $\sqrt{N}$  times, where  $\sqrt{N}$  is the square root of the number of items in the database.
4. Measure the quantum state, which will correspond to the desired state with high probability.





# The oracle operator

---

The oracle operator is a unitary operator that marks the desired state in the superposition by flipping the sign of the amplitude of that state.

The oracle operator is typically implemented using a black-box function that evaluates to 1 for the desired state and 0 for all other states. The oracle operator  $W$  can be expressed as:

$$W = I - 2|w\rangle\langle w|$$

where  $|w\rangle$  represents the desired state, and  $I$  is the identity matrix.

The oracle operator flips the sign of the amplitude of the desired state, while leaving the amplitudes of the other states unchanged.

This effectively “marks” the desired state in the superposition and allows the algorithm to amplify its amplitude.

# The Grover diffusion operator

---

The diffusion operator is a unitary operator that is applied to the superposition state  $|s\rangle$  after the oracle operator is applied.

The diffusion operator reflects the amplitude of the state about the mean value of the amplitudes, which amplifies the amplitudes of the states that are close to the mean value and decreases the amplitudes of the states that are far from the mean value. The diffusion operator can be expressed as:

$$D = 2|s\rangle\langle s| - I,$$

where  $I$  is the identity matrix and  $|s\rangle = (1/\sqrt{N})\sum|x\rangle$  is the equal superposition state.

The diffusion operator can be thought of as rotating the state vector towards the desired state.

# The number of iterations

---

By applying the oracle and diffusion operators alternately, Grover's algorithm iteratively amplifies the amplitude of the desired state in the superposition until it can be measured with high probability.

The number of iterations required is approximately  $O\sqrt{N}$ , where  $N$  is the size of the search space, which is significantly faster than classical algorithms that require  $O(N)$  time.

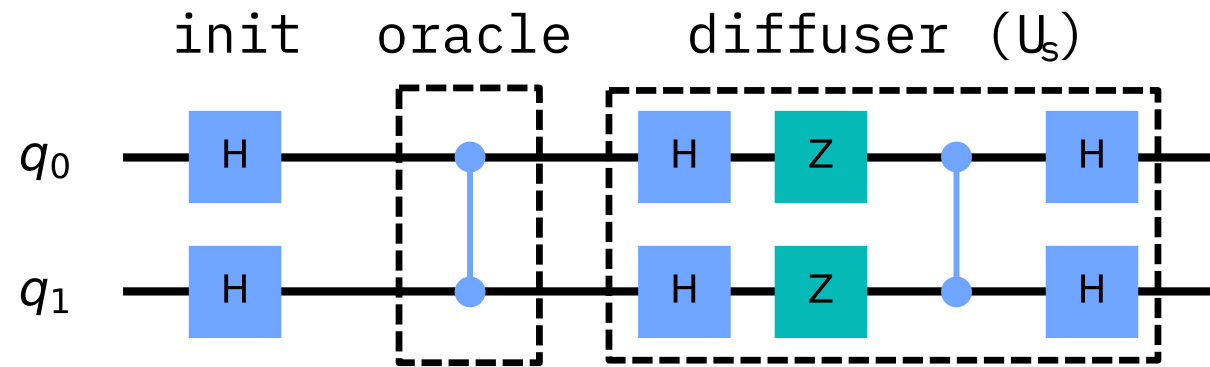
In an unstructured search algorithm, to find the *marked item* using classical computation, one would have to check on average  $N/2$  of the items, and in the worst case, all of them.

Grover's algorithm can serve as a general subroutine to obtain quadratic run time improvements for a variety of other algorithms.


$$O\sqrt{N}$$

# Quantum Circuit for Grover's Algorithm

---





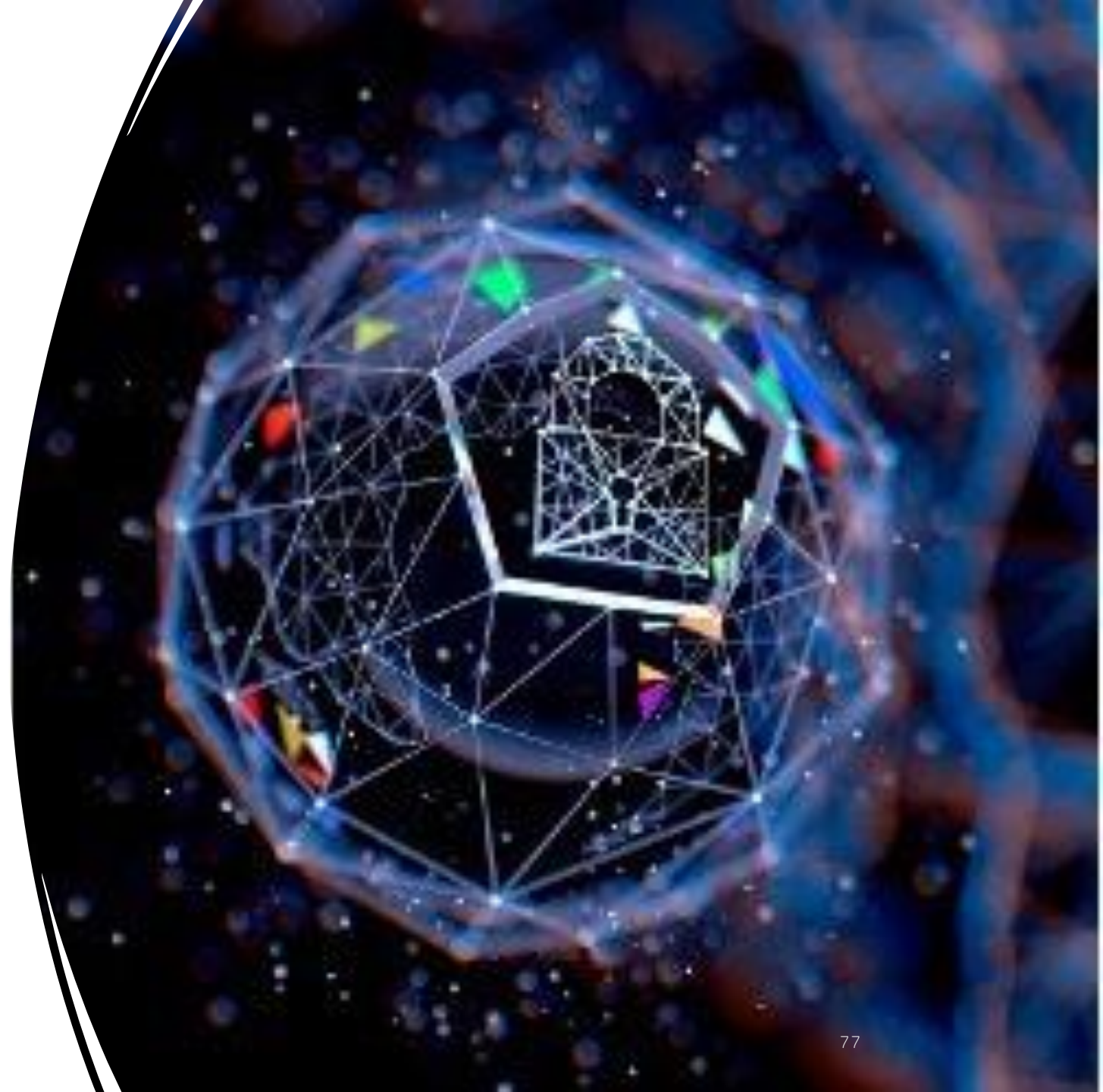
# Grover's algorithm vs cryptographic systems

---

The current state of quantum computing is not yet advanced enough to break real-world cryptographic systems using Grover's algorithm.

This could change in the future as quantum computers become more powerful and scalable.

Grover's algorithm can be used to attack cryptographic systems that rely on the difficulty of the classical brute-force search, such as symmetric key encryption and hashing.



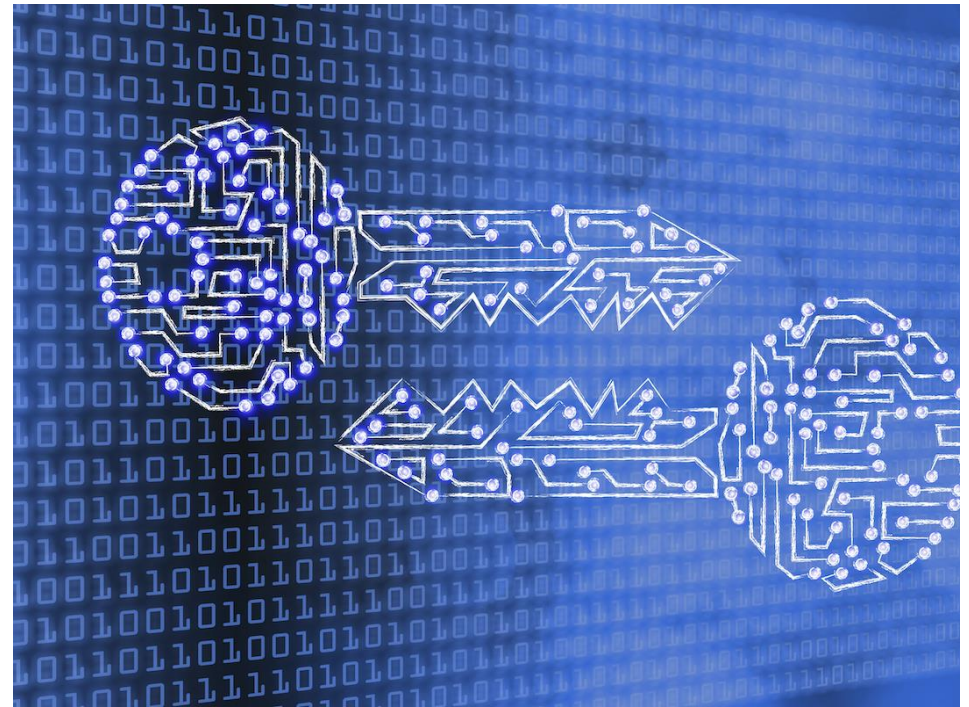
# Grover's algorithm vs symmetric encryption

Suppose that we have a symmetric encryption key of  $n$  bits.

In classical computing, the only way to find the key is to try all possible  $2^n$  keys until the correct one is found. This is a very time-consuming process, as it takes  $O(2^n)$  time.

However, Grover's algorithm can search the  $n$ -bit key space in  $O(\sqrt{2^n})$  or  $O(2^{n/2})$  time, which is much faster than the classical algorithm.

This roughly means that the encryption key size is equivalent to  $n/2$  bits.





# Grover's algorithm vs hash functions

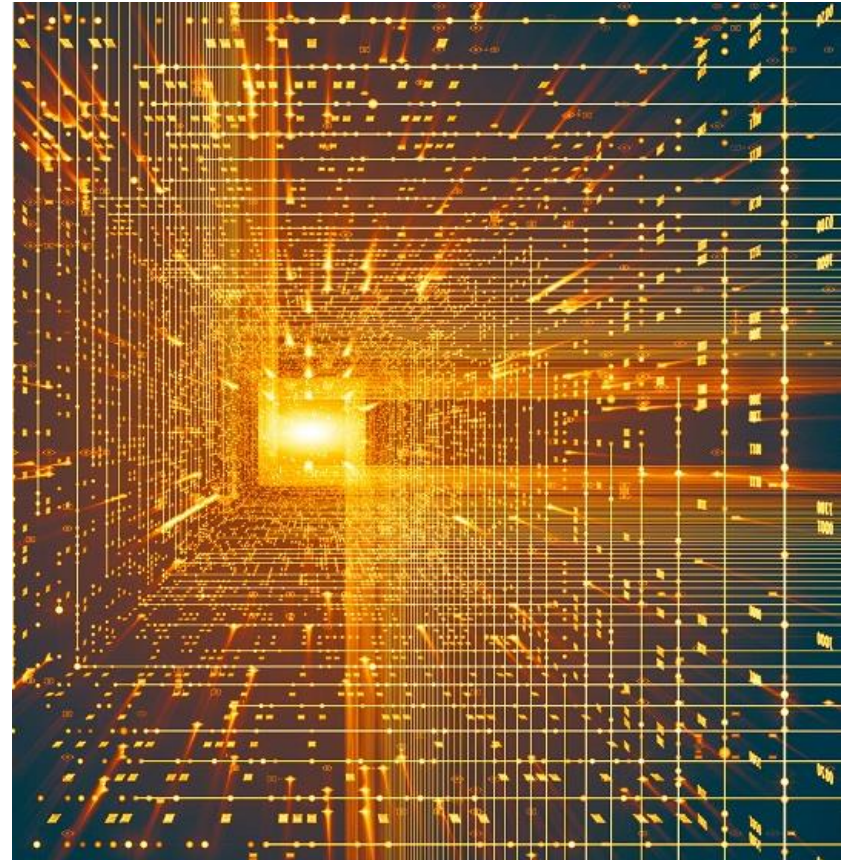
---

Grover's algorithm can be used to attack cryptographic hash functions.

A hash function is a one-way function that maps input data to a fixed-size output.

In classical computing, the only way to find a message that hashes to a specific value is to try all possible messages until the correct one is found.

Grover's algorithm can search the hash output space in  $O(2^{n/2})$  time, which could potentially allow an attacker to find a message that hashes to a specific value faster than classical methods.



# How to attack hash functions ?

---

Grover's algorithm can be used to attack cryptographic hash functions by searching for collisions, which are pairs of messages that produce the same hash value.

Collisions are a security weakness of hash functions because they allow an attacker to create two different messages that have the same hash value, which can be used for various types of attacks.

To perform a Grover's algorithm attack on a hash function, the attacker first needs to determine the number of bits  $n$  in the hash output.

Then, they can use Grover's algorithm to search for two messages that have the same hash value by creating a quantum circuit that generates a superposition of all possible messages, and then apply the hash function to each message.





# Consequence of the attack on hash functions

Once the attacker finds a collision, he/she can use it to mount various types of attacks, depending on the context of the hash function.

For example, in a *digital signature* scheme, an attacker could use a collision to forge a signature for a message that was not signed by the legitimate signer.



# Shor's algorithm

---

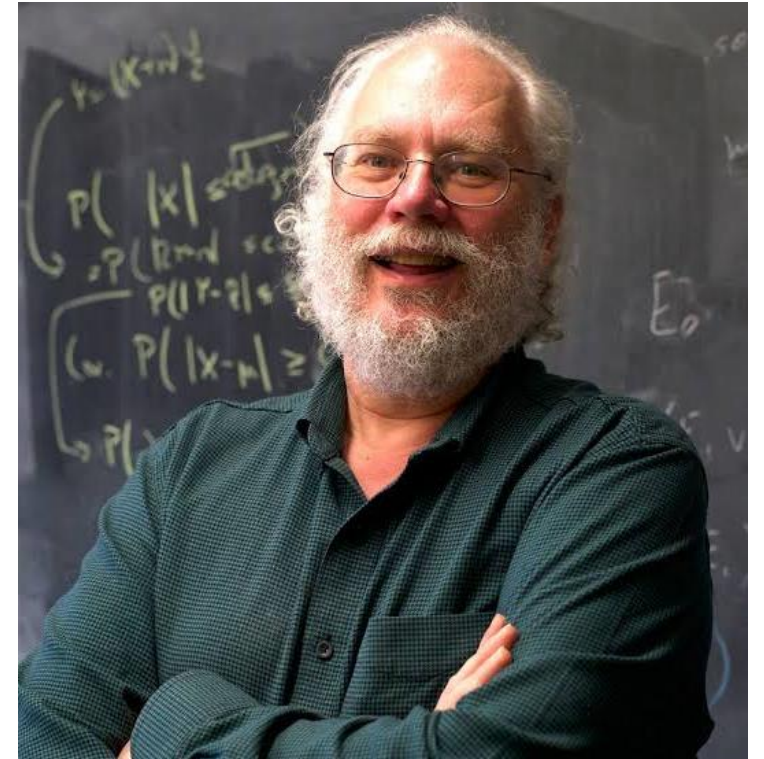
Introduced by Peter Shor in 1994.

Shor's algorithm has the potential to factor large integers much more efficiently than classical algorithms.

Shor's algorithm can factor an  $N$ -bit integer in  $O(N^3)$  time complexity, which is much faster than any known classical algorithm.

This is because quantum algorithms can exploit the quantum parallelism and interference to speed up computations.

The potential of quantum algorithms to efficiently factor large integers has significant implications for cryptography, as many widely used public key cryptography algorithms are based on the difficulty of factoring large integers.



# Integer factorization

---

Integer factorization is the process of finding the prime factors of a composite integer, which is an integer greater than one that is not itself a prime number.

For example, 126 can be factored into the product of its prime factors 2, 3 and 7:  $126 = 2 \times 3 \times 3 \times 7$ .

Integer factorization is a fundamental problem in number theory and has important applications in cryptography, including in the security of widely used public key cryptography algorithms such as RSA.

Prime Factorization of 126:

2	63
3	21
3	7

$= 2 \times 3 \times 3 \times 7$

# Classical factorization algorithms

The most widely used classical algorithms for factorization include the Trial Division Algorithm, the Pollard-Rho algorithm, and the General Number Field Sieve (GNFS) algorithm.

These algorithms have varying levels of efficiency and are able to factor different sizes of integers, but all of them have high time complexity and become increasingly slow for larger integers.

```
int n;
double temp,
cin>> n;      // number to be factored
for (int i=2; i <= sqrt((double)(n)); i++)
{
    temp=(double)n/(double)i;
    if (temp == (int)temp)
        cout<< i << ", " ;
}
```



# Shor's period-finding subroutine

Shor's algorithm uses the properties of quantum computers to efficiently find the period of a particular function.

The period-finding subroutine is the key step in Shor's algorithm, and it is used to find the factors of an integer by finding the period of a modular exponentiation function.

The modular exponentiation function takes three integers as input: a base number, an exponent, and a modulus. It calculates the result of raising the base to the exponent and then taking the result modulo the modulus.

The function in question is related to the integer to be factored and the factors of the integer can be obtained by using the period of this function.



# The modular exponentiation function

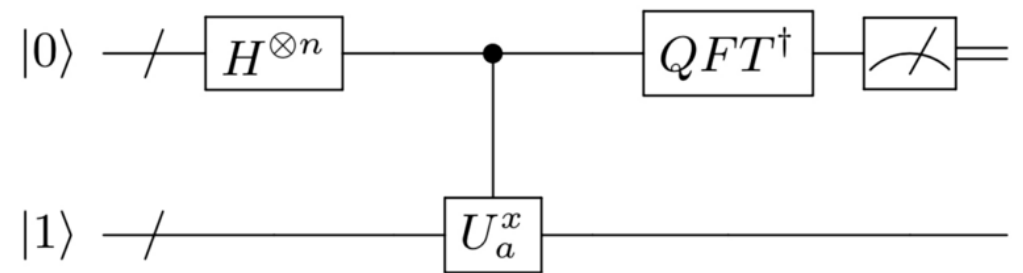
The modular exponentiation function can be expressed as:

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

where  $a$  is the base,  $x$  is the exponent, and  $N$  is the modulus.

To perform modular exponentiation efficiently, Shor's algorithm uses Quantum Fourier Transform (QFT) and modular arithmetic.

The QFT is used to efficiently calculate  $a^x \bmod N$  for a range of values of  $x$ .



# Shor's algorithm

---

1.  $N$  is a composite integer to be factored.
2. Choose a random integer  $a < N$ .
3. Use the period-finding subroutine to find the period  $r$  of the modular exponentiation function  $f(x) = a^x \bmod N$ . (The period  $r$  is the smallest positive integer such that  $a^r \bmod N = 1$ )
4. If  $r$  is odd or  $a^{r/2} \equiv -1 \pmod N$ , then go back to step 2 and choose a different random integer  $a$ .
5. If  $r$  is even and  $a^{r/2} \not\equiv -1 \pmod N$ , then the factors of  $N$  can be found using the greatest common divisor of  $(a^{r/2} + 1) \bmod N$  and  $(a^{r/2} - 1) \bmod N$ .

# Example for $N=21$

---

The period  $r$  is 6, which is even.

Let  $a=11$

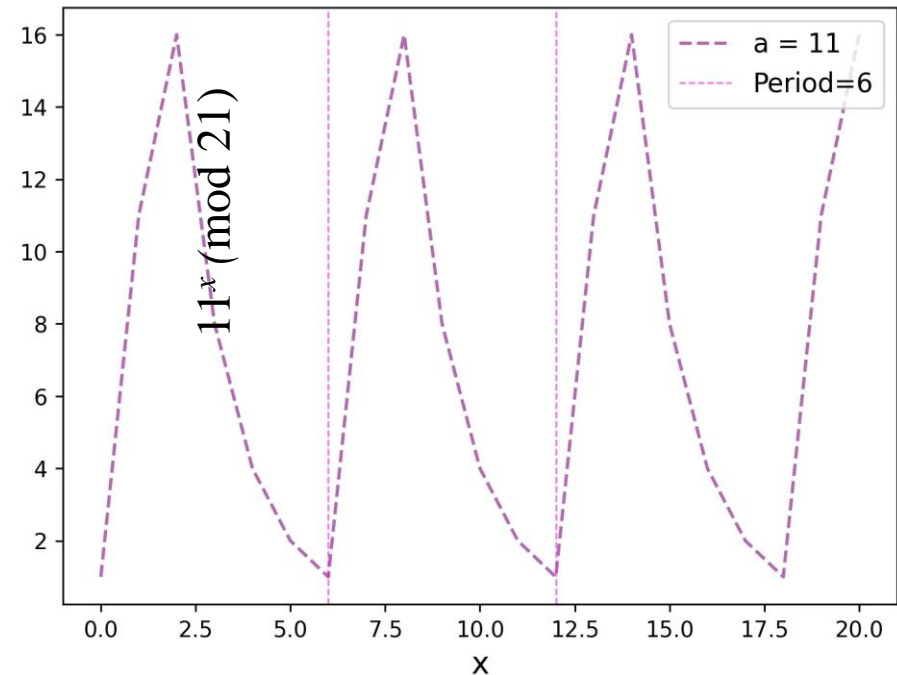
Following the steps of Shor's algorithm.

$$r/2 = 3, 11^3(\bmod 21) = 8.$$

The prime factors of 21 are:

$$\mathbf{gcd}(8+1, 21)=3$$

$$\mathbf{gcd}(8-1, 21)=7$$



# Time complexity of Shor's algorithm

---

The time complexity of Shor's algorithm arises from the two main steps involved in the algorithm:

1. The first step involves using a quantum Fourier transform to find the period of the function in question. This step takes  $O(N^2)$  time complexity on a quantum computer.
2. The second step involves classical post-processing to obtain the factors from the period found in the first step. This step takes  $O(N^3)$  time complexity on a classical computer.

Therefore, the overall time complexity of Shor's algorithm is dominated by the classical post-processing step, leading to a total time complexity of  $O(N^3)$ .


$$O(N^3)$$

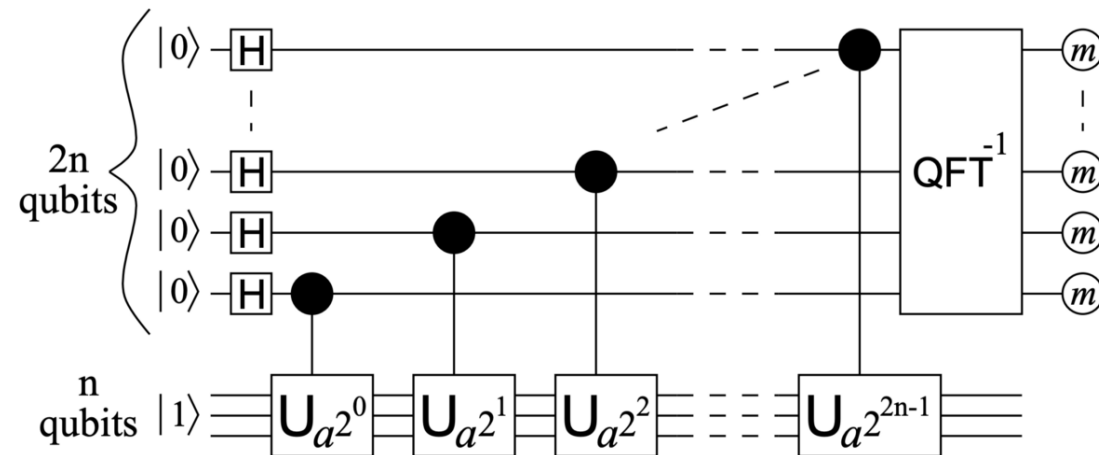


# Quantum Circuit for Shor's Algorithm

The **H gates** are used to create superposition states that enable quantum parallelism and evaluate the modular exponentiation function for many different input values simultaneously

The **inverse QFT** is needed to extract the period information encoded in the amplitudes of the input register after the QFT has been applied to the output register.

The  $|0\rangle$  state is used as the input to the H gate to create a uniform superposition of all possible input values, while the  $|1\rangle$  state is used as the input to the  $U_a$  gate to ensure that the initial state of the output register is a valid state for the modular exponentiation function.



The  $U_a$  gates perform modular exponentiation on a quantum register of  $2n$  qubits

## Impact of Shor's algorithm

---

Shor's algorithm has the potential to break the widely-used public-key cryptography systems such as RSA and Elliptic Curve Cryptography (ECC).

It would allow an attacker with a large enough quantum computer to decrypt messages that were previously considered secure.

This would compromise the confidentiality and integrity of sensitive information








## Impact of Quantum Computing on IoT Security





Quantum computing has the potential to greatly impact IoT security in both *positive* and *negative* ways.





# Positive Impact

On the positive side, quantum computing could enable **new cryptographic methods** that are more secure than current classical methods.

Quantum key distribution (QKD) can provide secure communication channels that are guaranteed to be free of eavesdropping, and **quantum-resistant encryption** algorithms can make it more difficult for attackers to compromise IoT devices.

Quantum computing can be used to **accelerate the discovery of vulnerabilities** and weaknesses in IoT devices, which can be used to enhance their security.

Quantum computing can be used to perform **more efficient security testing** and analysis, which can help identify and fix security flaws before they are exploited by attackers.



# Negative Impact

---


On the negative side, quantum computing could **break some of the current cryptographic** methods that are used to secure IoT devices.

Shor's algorithm, which runs efficiently on quantum computers, can be used to factor large integers, **breaking RSA** and other public-key cryptographic algorithms.


IoT security must evolve to include quantum-resistant cryptographic is particularly important for systems that have long lifetimes, such as embedded devices in critical infrastructure and industrial control systems.







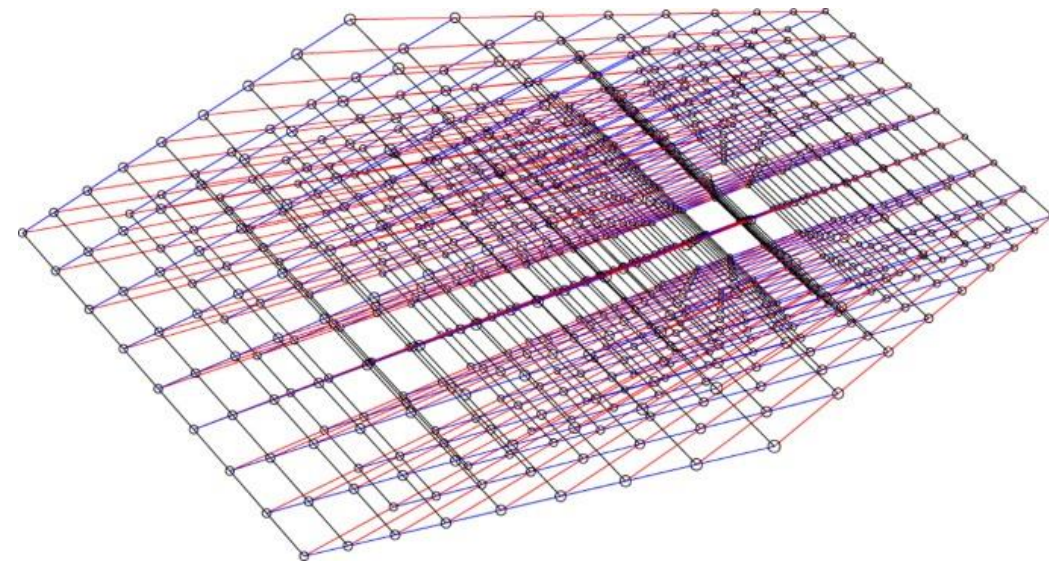
Quantum-resistant cryptographic methods are becoming increasingly important for securing IoT systems against quantum computing attacks.



Let us discover some examples of quantum-resistant cryptographic methods that are being developed specifically for IoT systems.



# Lattice-based cryptography



Lattice-based cryptography is a promising quantum-resistant cryptographic methods.

It is well-suited for IoT systems because it requires less processing power and memory.

It is a public-key cryptography based on mathematical properties of lattices in high-dimensional spaces.

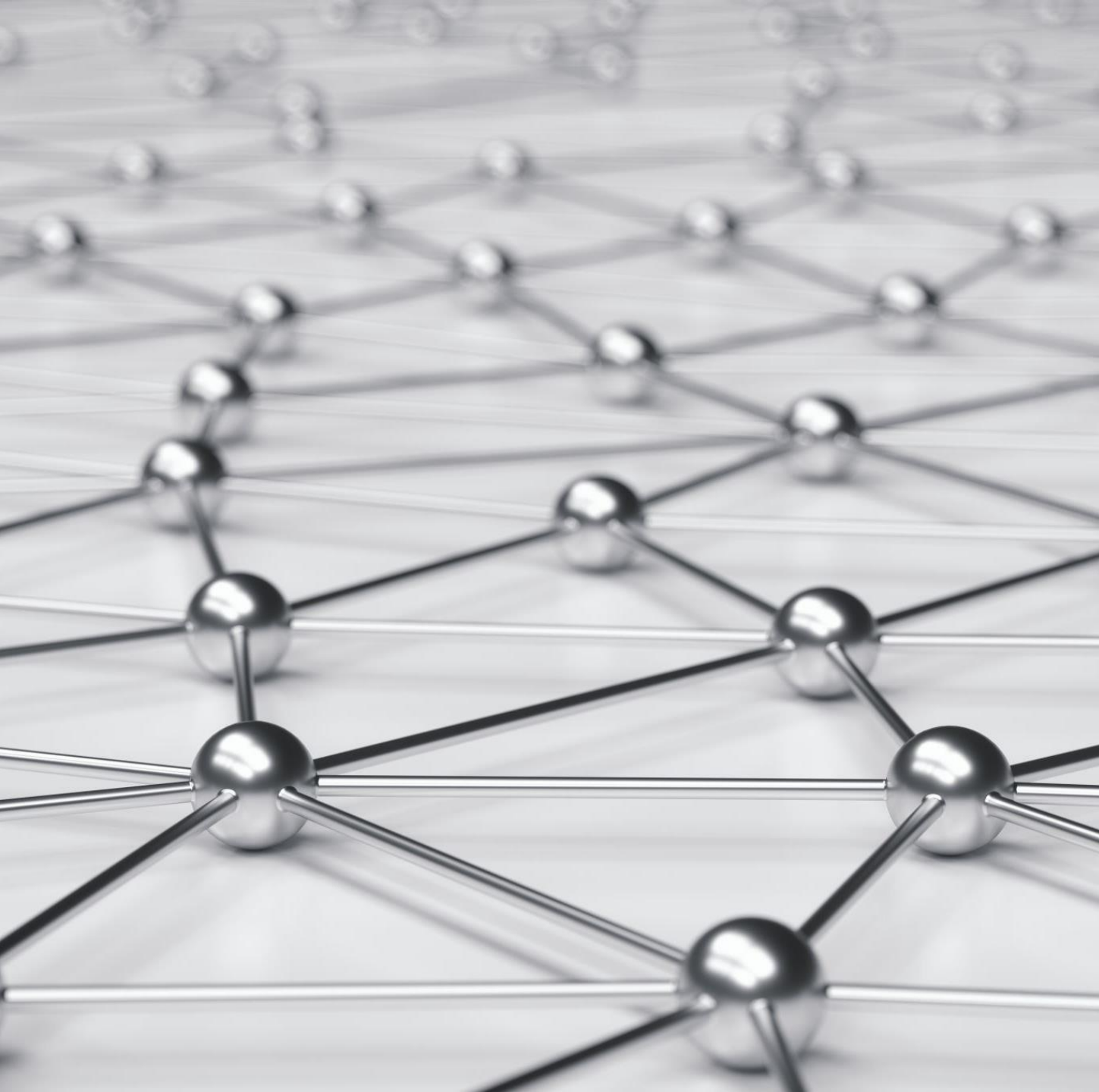
It can be used for secure communication and authentication in IoT systems.

It can be used for key exchange, digital signatures, and encryption.

The security of the system is based on the hardness of the Shortest Vector Problem (SVP) or the Closest Vector Problem (CVP) in high-dimensional lattices.

These problems are considered computationally intractable for classical and quantum computers (NP-Hard).





# Learning with Errors (LWE)

One popular lattice-based cryptographic algorithm is the Learning with Errors (LWE) problem.

A secret key is used to generate a set of public keys that are noisy versions of the secret key.

The security of the system is based on the difficulty of distinguishing the noisy public keys from random noise.



# Lattice-based cryptography Pros & Cons

**Pros:** Low computational and memory requirements, which makes it well-suited for resource-constrained devices such as IoT devices. It also provides strong security guarantees against both classical and quantum attacks.

**Cons:** The size of the keys required for lattice-based cryptography is relatively large compared to other cryptographic methods, which can be a challenge for resource-constrained devices. Additionally, the speed of lattice-based cryptographic operations can be slower than other cryptographic methods, which can impact the performance of systems that use lattice-based cryptography.



# Code-based cryptography

---

Code-based cryptography is another promising quantum-resistant cryptographic method for IoT systems.

It is based on the theory of error-correcting codes, and it can be used for digital signatures, key exchange, and encryption.

In code-based cryptography, the security of the system is based on the hardness of decoding random linear error-correcting codes.

The private key in a code-based cryptographic system is a generator matrix for the code, while the public key is a matrix that is derived from the private key using a random permutation.

The security of the system is based on the difficulty of finding the private key from the public key.





# McEliece cryptosystem

One popular code-based cryptographic algorithm is the McEliece cryptosystem.

It was first proposed by Robert McEliece in 1978 and considered as one of the most promising candidates for post-quantum cryptography.

A random binary Goppa code is used to generate the private key, while the public key is derived from the private key using a random permutation.

The security of the McEliece cryptosystem is based on the hardness of decoding random linear error-correcting codes.





# Code-based cryptography Pros & Cons

Pros: Relatively low computational and memory requirements, which makes it well-suited for resource-constrained devices such as IoT devices. Code-based cryptography has been extensively studied and has a well-established theoretical foundation.

Cons: The size of the keys required for code-based cryptography can be relatively large compared to other cryptographic methods, which can be a challenge for resource-constrained devices. The speed of code-based cryptographic operations can be slower than other cryptographic methods, which can impact the performance of systems that use code-based cryptography.

# Hash-based cryptography

---

Hash-based cryptography is a simple and efficient quantum-resistant method that can be used for IoT systems.

It is also known as one-time signature (OTS) or Merkle signature.

A hash function is used to create a one-way function that maps a message of arbitrary length to a fixed-length output. This output is then used as the digital signature of the message.

To verify the signature, the receiver computes the hash of the original message and compares it with the digital signature received. If the two match, the signature is considered valid.

The security of hash-based cryptography is based on the collision resistance of the underlying hash function.

# Hash-based cryptography Pros & Cons

**Pros:** Relatively low computational and memory requirements, which makes it well-suited for resource-constrained devices such as IoT devices. The size of the keys used in hash-based cryptography is relatively small compared to other cryptographic methods.

**Cons:** Limited number of signatures that can be created using a given key pair. This means that the private key must be frequently changed to avoid exhausting the number of available signatures. Hash-based cryptography is vulnerable to preimage attacks, where an attacker can find an input that hashes to a given output.

# Quantum Key Distribution for IoT

QKD can be used to generate and distribute secure cryptographic keys in IoT systems.

In IoT systems, QKD can be used to securely exchange keys between devices and gateways, which can then be used to encrypt and decrypt data transmitted between them.





## QKD Pros & Cons

**Pros:** It provides a high degree of privacy, since any attempt to eavesdrop on the quantum communication channel will be detected by the two parties.

**Cons:** The quantum communication channel must be carefully engineered to ensure that the photons can be transmitted without being disturbed by environmental factors such as temperature fluctuations or vibrations. QKD can be relatively slow compared to other cryptographic methods, which can impact the performance of IoT systems.

# Isogeny-based cryptography

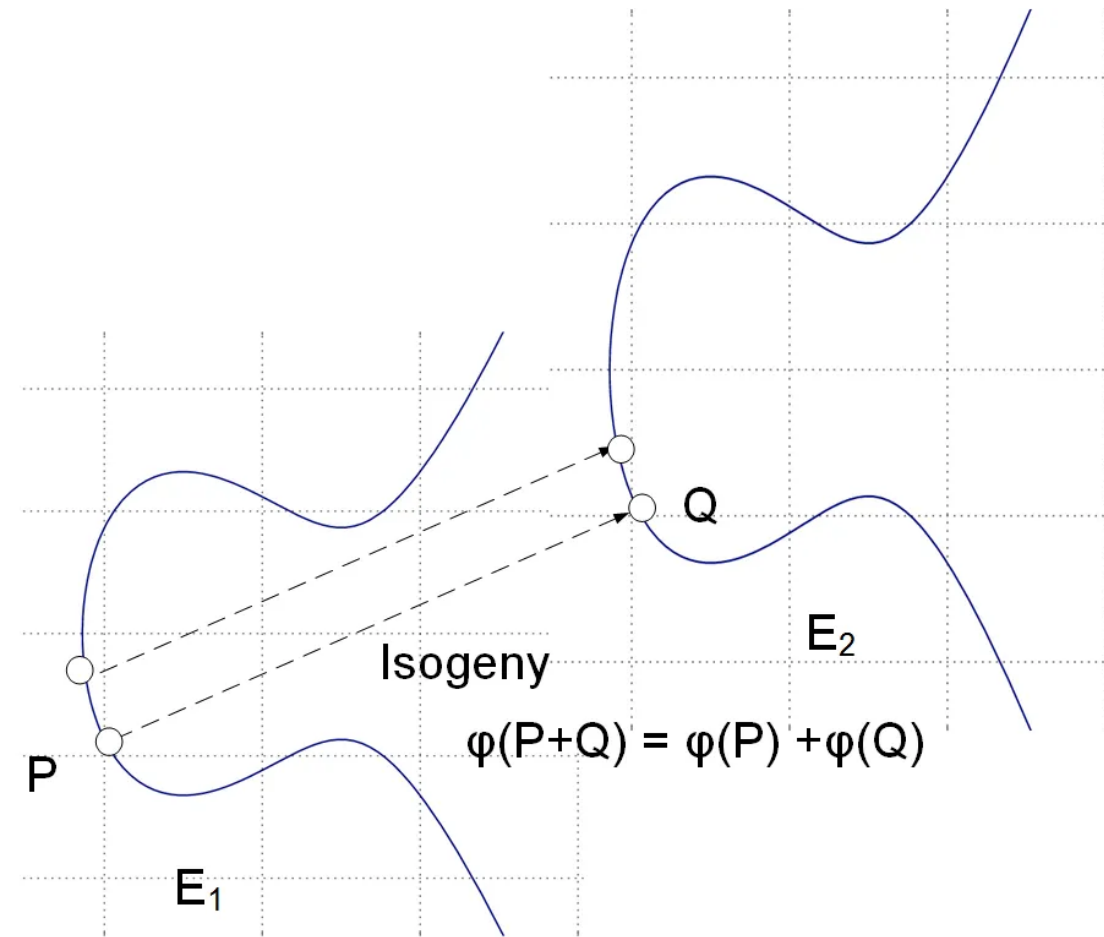
Isogeny-based cryptography is a promising quantum-resistant cryptographic method that is based on the mathematical properties of elliptic curves.

It can be used for key exchange and digital signatures in IoT systems.

Isogenies are mathematical functions that map one elliptic curve to another, while preserving certain algebraic properties.

The private key consists of a secret isogeny that is used to derive a public key, which can then be used to encrypt messages or authenticate digital signatures.

The security of the system is based on the difficulty of computing the secret isogenies between two elliptic curves from the public key.





# Isogeny-based cryptography Pros & Cons

**Pros:** Relatively small key sizes compared to other post-quantum cryptographic methods. This makes it well-suited for resource-constrained devices such as IoT devices.

**Cons:** computational complexity of the isogenies between elliptic curves. This can make the system relatively slow compared to other cryptographic methods. The security of isogeny-based cryptography is still an active area of research, and there may be unknown attacks that could be used to break the system.



The background of the slide is a complex, multi-layered digital graphic. It features a central, glowing, multi-faceted geometric shape, possibly a cube or a complex polyhedron, rendered in vibrant colors like purple, blue, and orange. This central shape is surrounded by a dense field of smaller, semi-transparent geometric shapes (squares, circles, triangles) and patterns, creating a sense of depth and complexity. The overall color palette is rich and varied, with a dark background that makes the bright colors stand out.

# Conclusion

---

Quantum computing could be used to strengthen the security of IoT systems, while on the other hand, it could pose new security risks to IoT systems.

Quantum computing could be used to develop new cryptographic systems that are resistant to quantum attacks.

Quantum computing could be used to develop new algorithms for analyzing data collected from IoT devices.

Quantum computers can be used to crack the symmetric-key cryptography used to secure the communication between IoT devices and the cloud.

Quantum computers can be used to tamper with the data collected from IoT devices.

Quantum resistance is a very hot research topic.





Děkuji