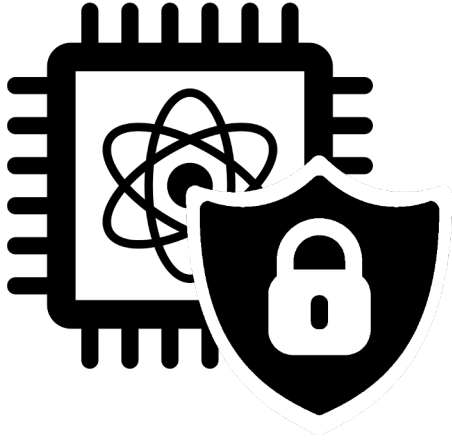




Tutorial on Security of Quantum Computing Systems

QCE Tutorial

Sept. 17, 2026

Prof. Jakub Szefer (jakub.szefer@northwestern.edu)

Electrical and Computer Engineering

Northwestern University



Computer Architecture
and Security Lab (CASLAB)



Northwestern
University

Tutorial Outline

- **1:00pm – 1:30pm: Introduction and Motivation**
- **1:30pm – 2:30pm: Crosstalk and Fault Injection in NISQ Quantum Computers**
- **2:30pm – 3:00pm: Coffee Break**
- **3:00pm – 3:45pm: Side Channels in NISQ and FTQC Quantum Computers**
- **3:45pm to 4:30pm: Trusted Execution Environments for Quantum Computers**
- **4:00pm: End**



Resource: Quantum Security Tutorials Web Page

CASLAB

Home

Our Group

Publications

Books

Tutorials

Next Tutorial in:
no upcoming tutorial is currently scheduled

Next:

Tutorial on Security of Quantum Computing Systems

September 17, 2026, co-located with QCE 2026.

Future:

Winter School on Attacks, Defenses, and Secure Design of Quantum Computing Systems

November 11, 2026, co-located with QCCS 2026.

Past:

Tutorial on Quantum Computer Architecture Security

June 27, 2026, co-located with ISCA 2026.

Tutorial on Security of Quantum Computing Systems

August 31, 2025, co-located with QCE 2025.

Tutorial on Quantum Computer Architecture Security

June 21, 2025, co-located with ISCA 2025.

© 2026 Jakub Szefer

Tutorials on Classical and Quantum Computer Architecture and Hardware Security

About: A series of tutorials on classical and quantum computer architecture and hardware security, presented by Prof. Jakub Szefer of the Computer Architecture and Security Lab (CASLAB) at Northwestern University.



**Tutorials on
Classical and Quantum
Computer Architecture and
Hardware Security**

Prof. Jakub Szefer (jakub.szefer@northwestern.edu)
Electrical and Computer Engineering
Northwestern University



Prior tutorials include: ISCA 2026, ISCA 2025, HOST 2025, ICCAD 2024, QCE 2023, HOST 2021, 2019, ACACES 2019, HOST 2019, HiPEAC 2019, and HOST 2018.



Computer Architecture
and Security Lab (CASLAB)



Northwestern
University

Resource: Quantum Computer Hardware Cybersecurity BibTeX

qc-hardware-cybersecurity-bibtexPublic

PinUnwatch4Fork7Star15

main1 Branch0 Tags

Go to fileTAdd fileCode

caslab-code

Updating research paper list with recent and older papers, mainly 202...14a4309 · now34 Commits

README.md	Updating email address in README file.	last year
refs-qc-hardware-cybersecurity-papers.bib	Updating research paper list with recent and older papers,...	now
refs-qc-hardware-cybersecurity-surveys.bib	Update refs-qc-hardware-cybersecurity-surveys.bib	11 months ago

README

About

This repository contains a BibTeX bibliography file with references to research papers pertaining to security of quantum computers. Anybody is encouraged to contact the author or make pull request to add papers they would like to be considered for inclusion in the bibliography. Papers on certain topics such as post-quantum cryptography or quantum key distribution are excluded from this bibliography so that the focus can remain only on research papers dealing with attacking and defending quantum computer systems, architectures, and hardware.

Contact

To have papers considered for addition to this list, please create a pull request or or e-mail the author at: jakub.szefer@northwestern.edu

About

List of references for quantum computer hardware cybersecurity papers.

ReadmeActivity15 stars4 watching7 forks

Releases

No releases published
[Create a new release](#)

Packages

No packages published
[Publish your first pack](#)

Contributors4

caslab-code

cnpapado Kc

chuanqixu C

yizhuo1007



Resource: YouTube Lectures



Secure Quantum Computing

Prof. Jakub Szefer
Electrical and Computer Engineering
Northwestern University

Class Session 8
Topic: Quantum Computing Basics, Part 2

Secure Quantum Computing / CE 395 / ...

by Jakub Szefer

Playlist • 11 videos • 250 views

This playlist contains video recordings of lectures from the Secure Quantum Computing course at Northwestern University. [more](#)

▶ Play all🔖🔗⋮

 Secure Quantum Computing Prof. Jakub Szefer Electrical and Computer Engineering Northwestern University Class Session 8 Topic: Quantum Computing Basics, Part 2 1:07:23	Secure Quantum Computing / Spring 2026 / Class 8 / Quantum Computing Basics, Part 2 Jakub Szefer 18 views • 2 weeks ago	⋮
 Secure Quantum Computing Prof. Jakub Szefer Electrical and Computer Engineering Northwestern University Class Session 7 Topic: Quantum Computing Basics, Part 1 53:19	Secure Quantum Computing / Spring 2026 / Class 7 / Quantum Computing Basics, Part 1 Jakub Szefer 43 views • 3 weeks ago	⋮
 Secure Quantum Computing Prof. Jakub Szefer Electrical and Computer Engineering Northwestern University Class Session 6 Topic: Threat Categories for Quantum Computing 59:45	Secure Quantum Computing / Spring 2026 / Class 6 / Threat Categories for Quantum Comp. Sys., Part 2 Jakub Szefer 173 views • 1 month ago	⋮
 Secure Quantum Computing Prof. Jakub Szefer Electrical and Computer Engineering Northwestern University Class Session 4 Topic: Threat Categories for Quantum Computing 54:28	Secure Quantum Computing / Spring 2026 / Class 4 / Threat Categories for Quantum Comp. Sys., Part 1 Jakub Szefer 90 views • 1 month ago	⋮
 Secure Quantum Computing Prof. Jakub Szefer Electrical and Computer Engineering Northwestern University Class Session 3 Topic: Computer Security Basics, Part 2 1:00:04	Secure Quantum Computing / Spring 2026 / Class 3 / Computer Security Basics, Part 2 Jakub Szefer 134 views • 1 month ago	⋮
 Secure Quantum Computing Prof. Jakub Szefer Electrical and Computer Engineering Northwestern University Class Session 2 Topic: Computer Security Basics, Part 1 1:12:52	Secure Quantum Computing / Spring 2026 / Class 2 / Computer Security Basics, Part 1 Jakub Szefer 268 views • 1 month ago	⋮
 Secure Quantum Computing Prof. Jakub Szefer Electrical and Computer Engineering Northwestern University Class Session 1 Topic: Introduction to Secure Quantum Computing 1:14:13	Secure Quantum Computing / Spring 2026 / Class 1 / Introduction to Secure Quantum Computing Jakub Szefer 371 views • 1 month ago	⋮



Resource: Upcoming QCCS Symposium 2026

[QCCS 2026](#) [Program](#) [Venue](#) [Directions](#) [Contact](#) [Prior QCCS 2025](#) [Prior QCCS 2024](#) [Prior QCCS 2023](#)

QCCS 2026 **4th Annual Quantum** **Computer Cybersecurity** **Symposium**

To be held November 11 and 12,
2026 at The Guild Lounge in
Scott Hall at Northwestern
University



Computer Architecture
and Security Lab (CASLAB)

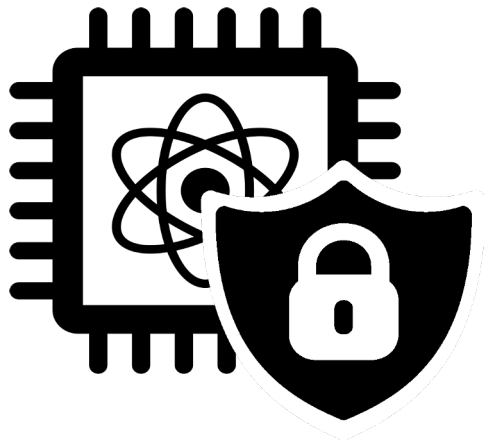


Northwestern
University

Tutorial Outline

- **1:00pm – 1:30pm: Introduction and Motivation**
 - What is Quantum Computing and Security?
 - Brief History of Quantum Computing and Motivation for Security Research
 - Introduction to Quantum Computation and Security Threats
- **1:30pm – 2:30pm: Crosstalk and Fault Injection in NISQ Quantum Computers**
- **2:30pm – 3:00pm: Coffee Break**
- **3:00pm – 3:45pm: Side Channels in NISQ and FTQC Quantum Computers**
- **3:45pm to 4:30pm: Trusted Execution Environments for Quantum Computers**
- **4:00pm: End**





What is Quantum Computer Security?

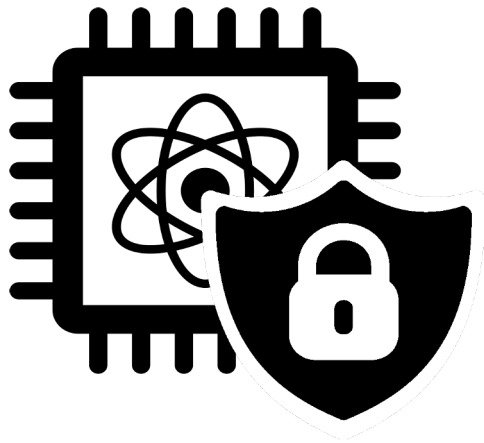


Quantum Computing + Security = ?

There are **at least three different types of research that may be associated with quantum computer security**:

1. **Post-Quantum Cryptography (PQC)** – study and design of new cryptographic algorithms that are resistant to cryptanalysis of by quantum computers – **not part of this tutorial**
2. **Using Quantum Computers as Attackers** – most famously trying to use Shor's algorithm to break cryptographic algorithms; but also any use of quantum computer resources to perform a security attack – **not part of this tutorial**
3. **Analyzing and Defending Quantum Computers from Attacks** – finding and fixing potential security flaws with quantum computers, ensuring integrity and confidentiality of quantum computation and quantum systems – **focus of this tutorial**

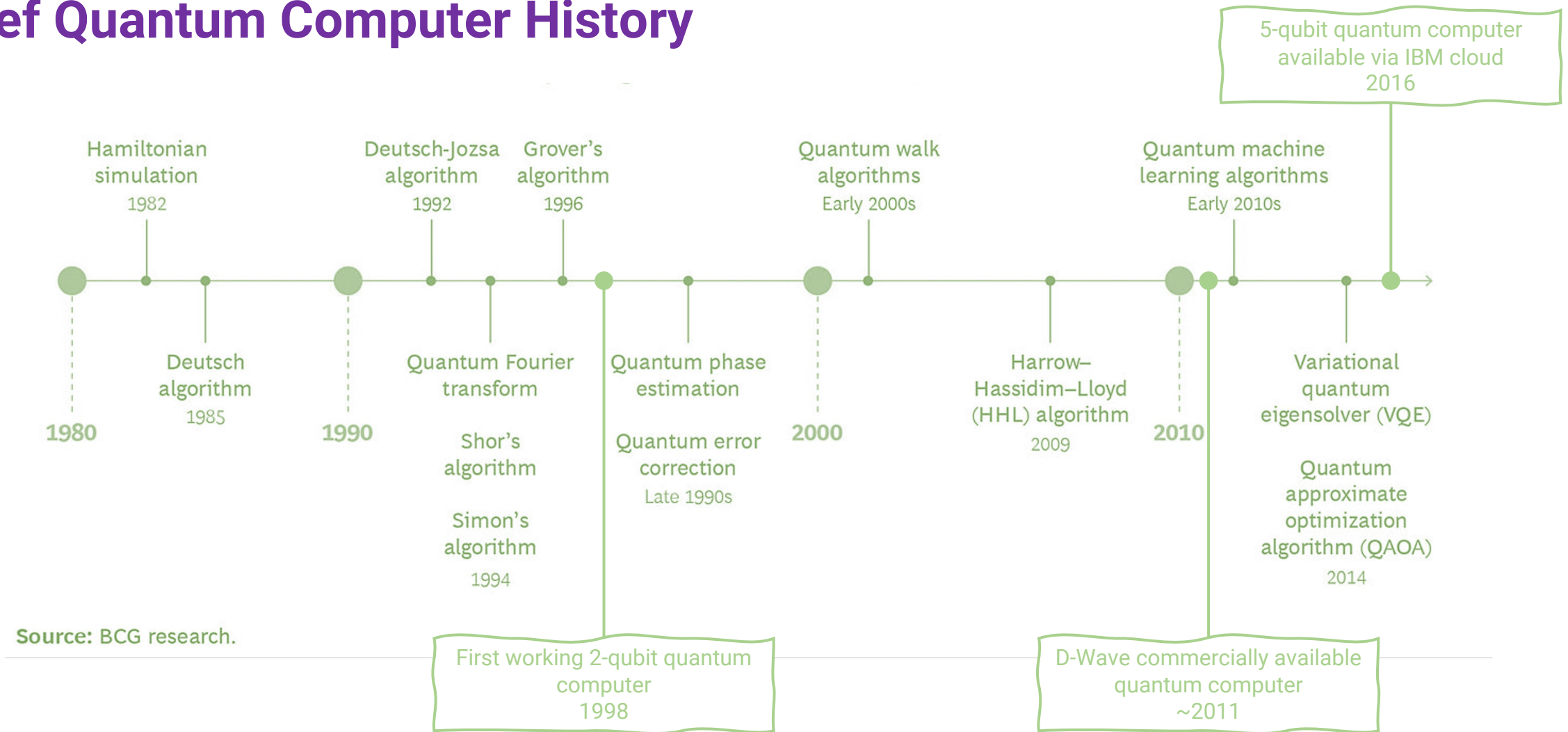




Brief Quantum Computer History and Motivation for Security Research



Brief Quantum Computer History



Why Focusing on Quantum Computing?

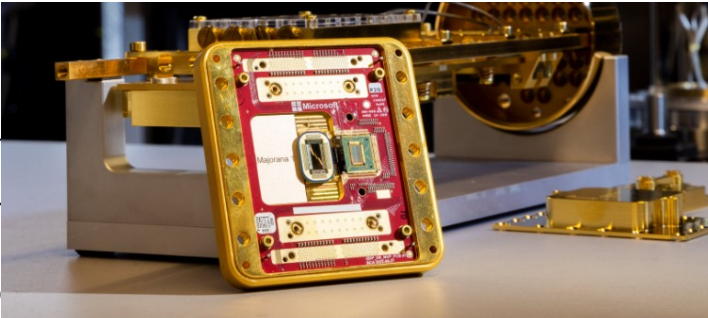
Meet Willow, our state-of-the-art quantum chip

Dec 09, 2024
6 min read

Hartmut Neven
Founder and Lead

Demonstrating quantum-centric use cases

With utility-scale quantum computing methods cannot, we see performance computing.



Microsoft's Majorana 1 chip carves path for quantum computing

Written by
Catherine Bolgar

Published
February 19,

Category
Innovation

Microsoft today [introduced Majorana 1](#), the world's first quantum chip powered by a new Topological Core architecture that it expects will realize quantum computers capable of solving meaningful, industrial-scale problems in years, not decades.

+ QPU

Amazon Web Services announces a new quantum computing chip

New 'Ocelot' chip uses scalable architecture for reducing error correction by up to 90% and accelerating the development of real-world quantum computing applications.

Accelerating Quantum Computing Research



NVIDIA Accelerated Quantum Computing Research Center

The NVIDIA Accelerated Quantum Computing Research Center (NVAQC) is where qubits go to become accelerated quantum supercomputers. By integrating partner quantum hardware with a state-of-the-art NVIDIA GB200 NVL72 system the NVAQC brings developers and academics the breakthroughs needed to scale quantum computing hardware and applications.

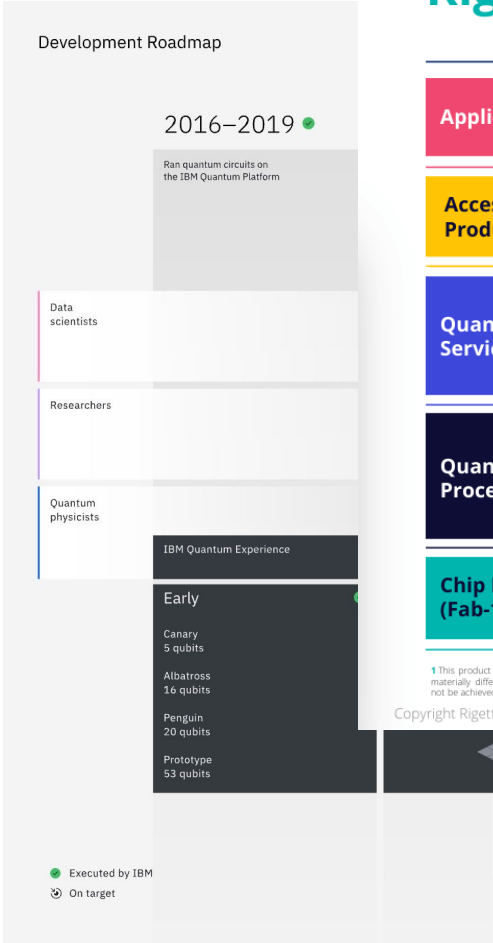


Computer Architecture
and Security Lab (CASLAB)

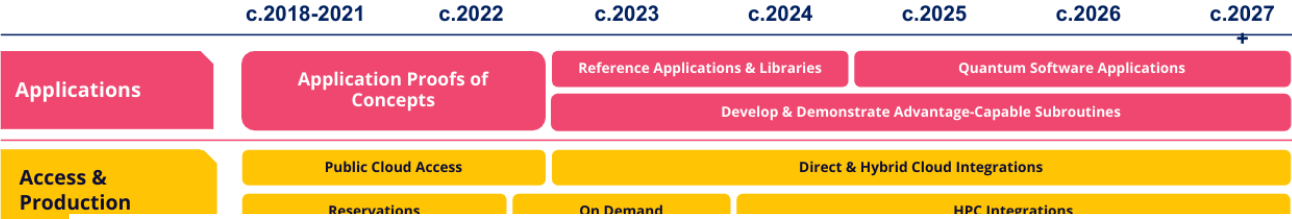


Northwestern
University

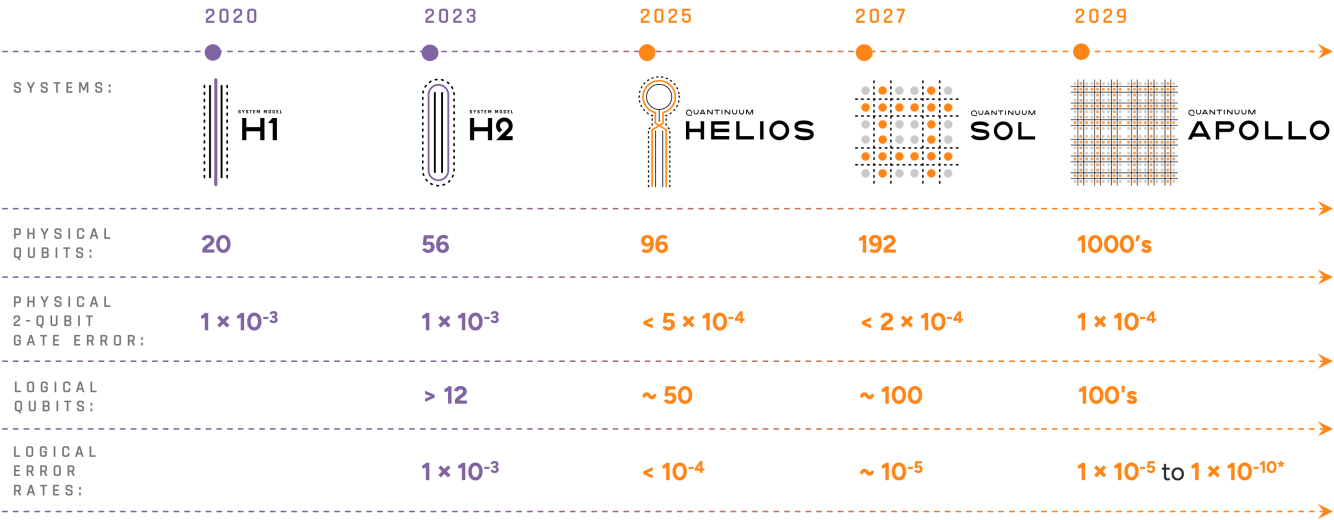
Why Focusing on Quantum Computing?



Rigetti Roadmap Aims to Reach Quantum Advantage¹



Development roadmap



© 2024 Quantinuum. All Rights Reserved.

*analysis based on recent literature in new, novel error correcting codes predict that error could be as low as $1\text{E-}10$ in Apollo (ref: arXiv:2403.16054, arXiv:2308.07915)



Why Focusing on Cloud-based Quantum Computing?

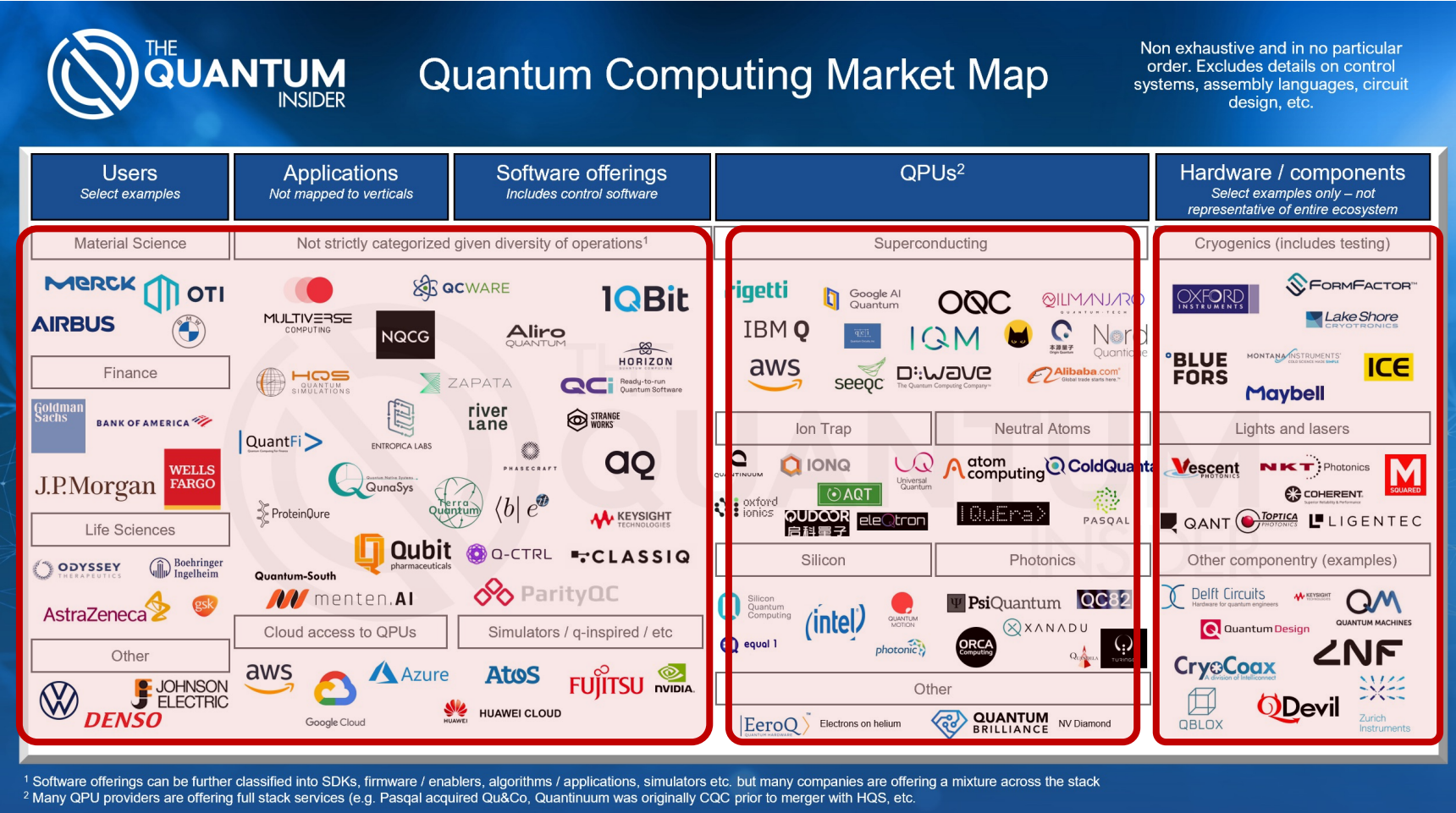
There is growing number of cloud provider that give access to quantum computers:



Presently there are almost no quantum-specific security features



There is a Large Ecosystem of Companies, and Possible Threats



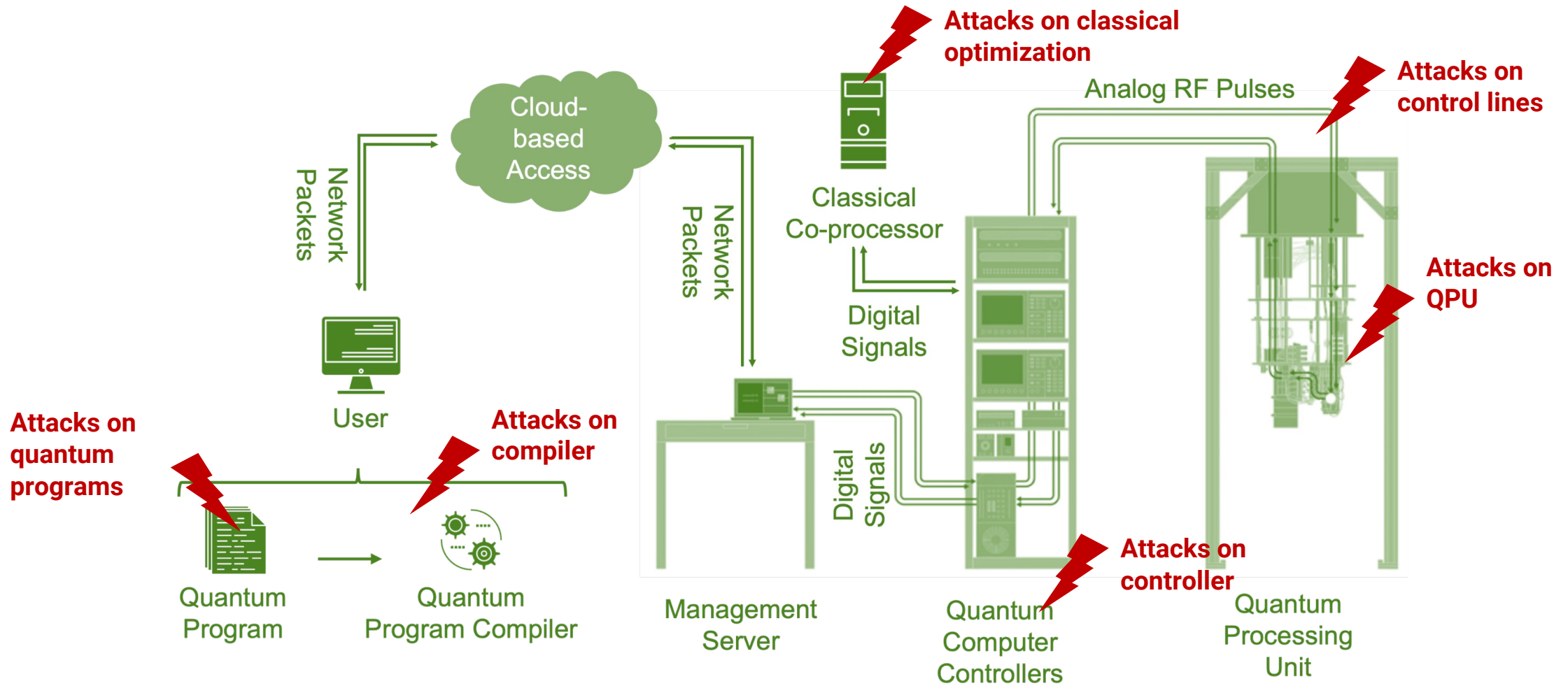
Threats to intellectual property and proprietary data

Threats to hardware IP, reverse engineering, hardware trojans, hardware attacks: standard, all classical equipment

Threats to hardware IP, reverse engineering, hardware trojans, hardware attacks: more exotic due to QPU designs



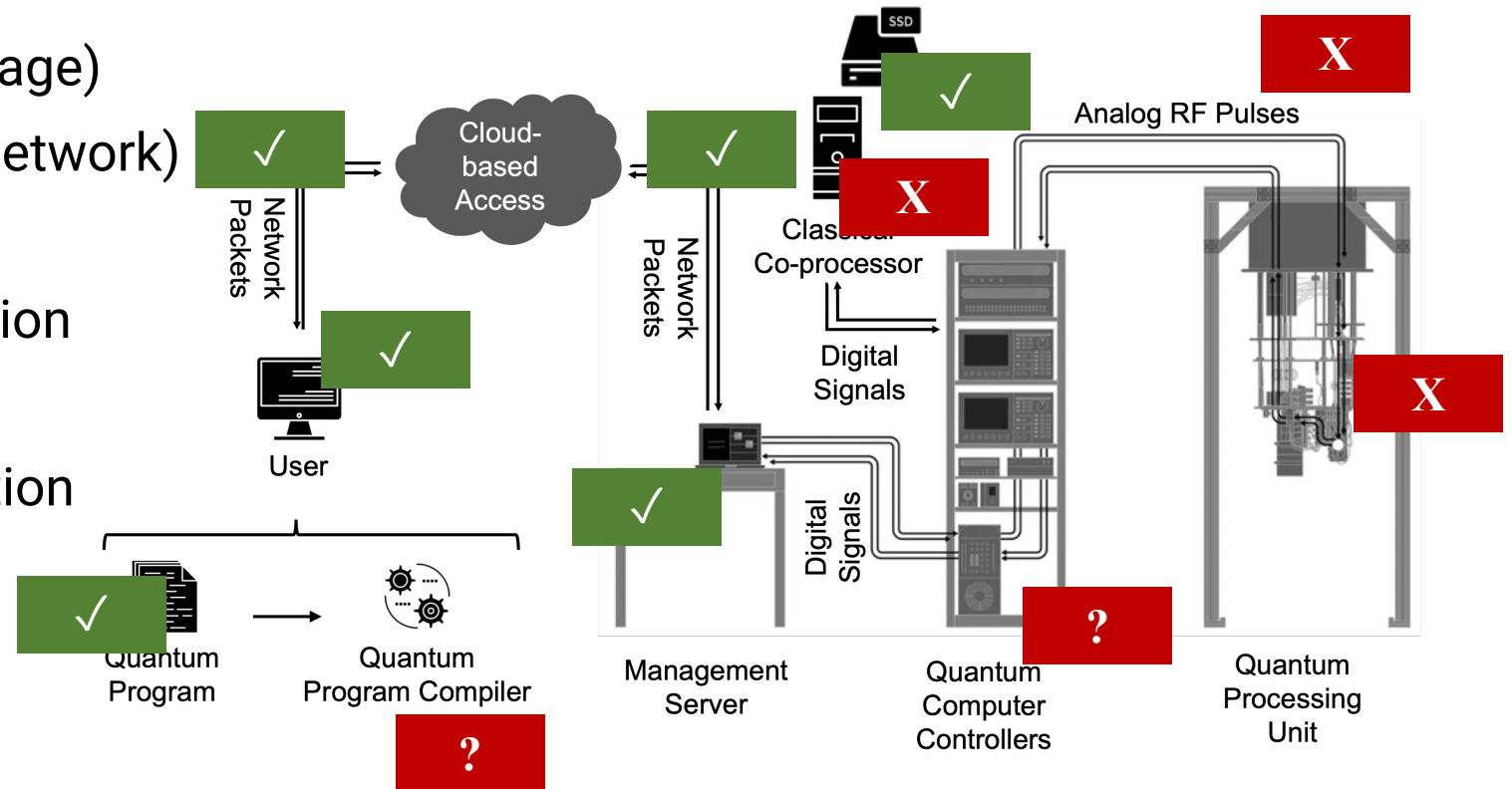
Quantum Computer Architecture, and Possible Threats



Aside: Limits of Confidentiality Protection with Ciphers

Ciphers can encrypt secret code and data, and without knowledge of the key the data cannot be decrypted.

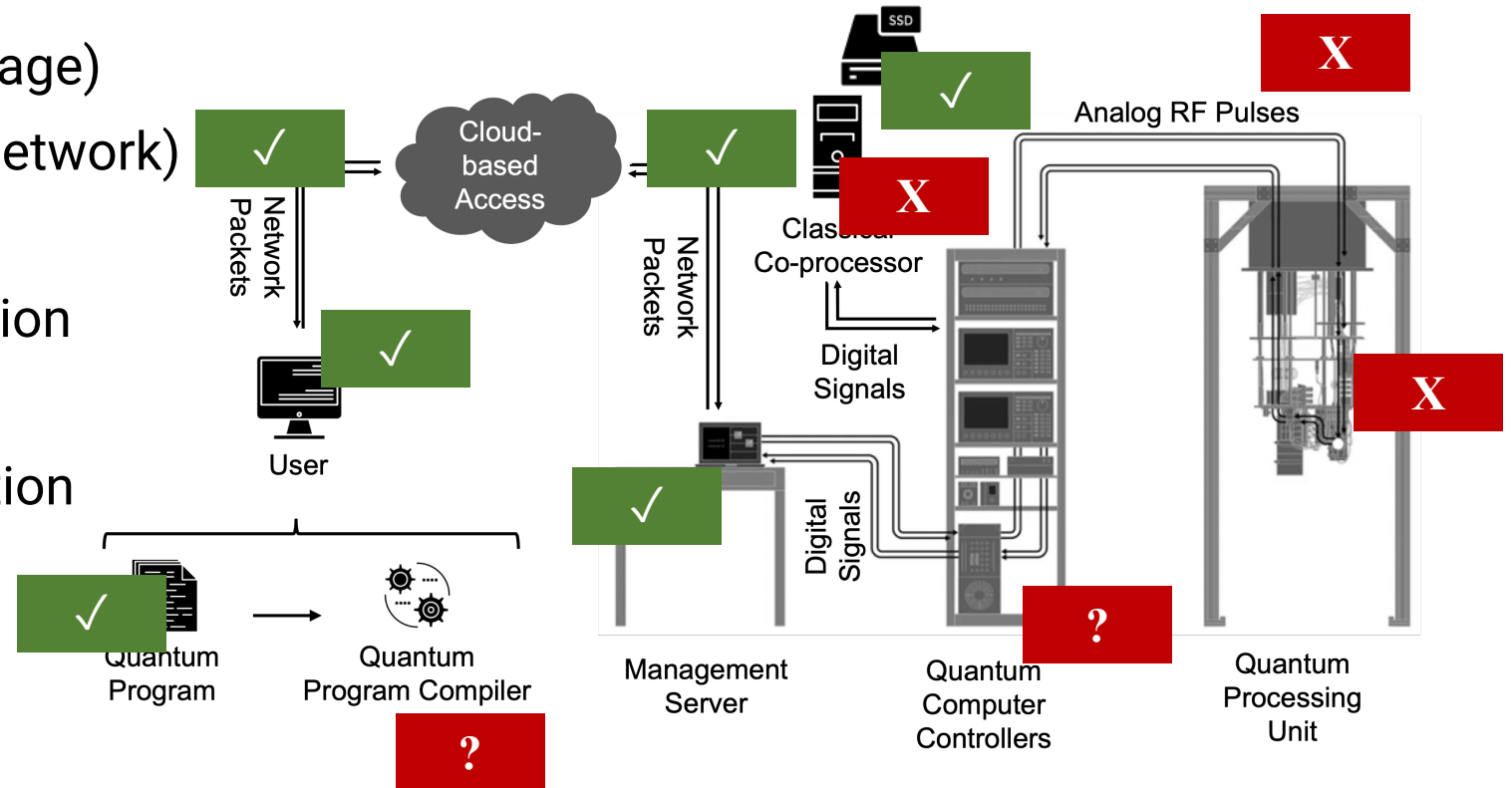
- Protect data and code at rest (storage)
- Protect data and code in transit (network)
- Cannot protect classical computation
- Cannot protect analog data
- Cannot protect quantum computation

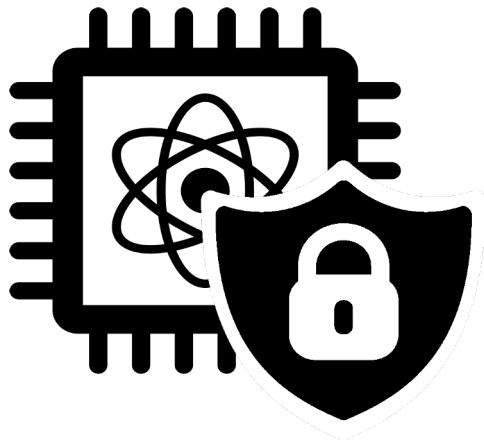


Aside: Limits of Integrity Protection with Hashes

Hashes ensure code and data cannot be modified (integrity violation) if attacker has no access to the key or the reference hash value.

- Protect data and code at rest (storage)
- Protect data and code in transit (network)
- Cannot protect classical computation
- Cannot protect analog data
- Cannot protect quantum computation



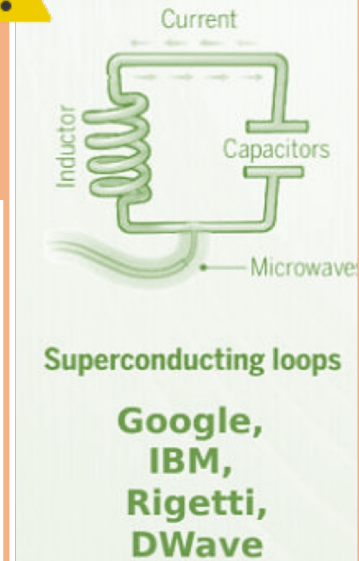


Circuit View of Quantum Computation



Qubit Types

! Tutorial will focus mostly on superconducting qubits.



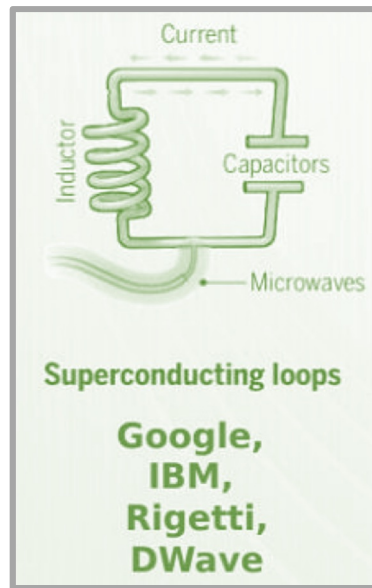
! Equally important but much less explored from security perspective

Many ways to realize a qubit:

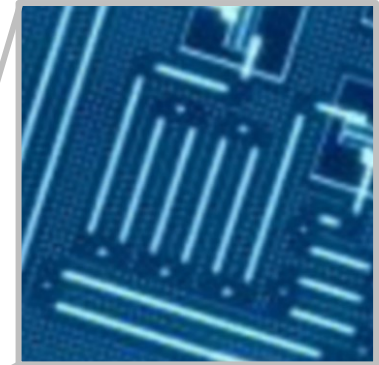
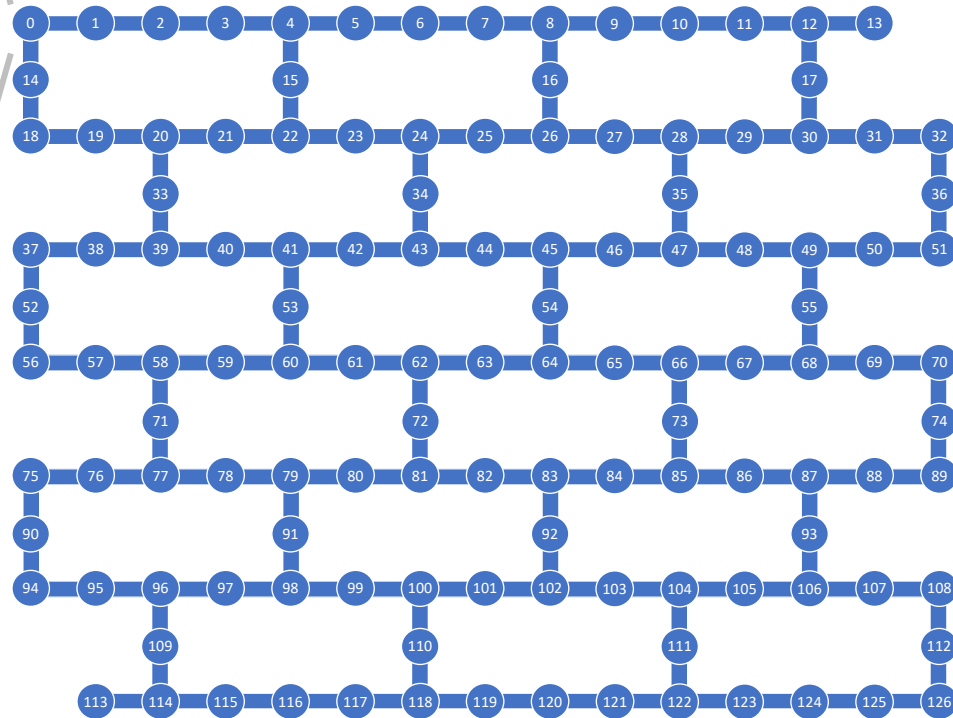
- Nuclear spin in NMR (Nuclear magnetic resonance) system: $\uparrow = |0\rangle$, $\downarrow = |1\rangle$
- Photons in a cavity: 0 photon = $|0\rangle$, 1 photon = $|1\rangle$
- Energy states of an atom: ground state $|0\rangle$, excited state $|1\rangle$
- Energy levels of transmon circuit: ground state $|0\rangle$, first excited state $|1\rangle$



Superconducting Qubit Quantum Computer Topology



A transmon qubit, used to store qubit state.



A coupler, used for 2-qubit gate operations

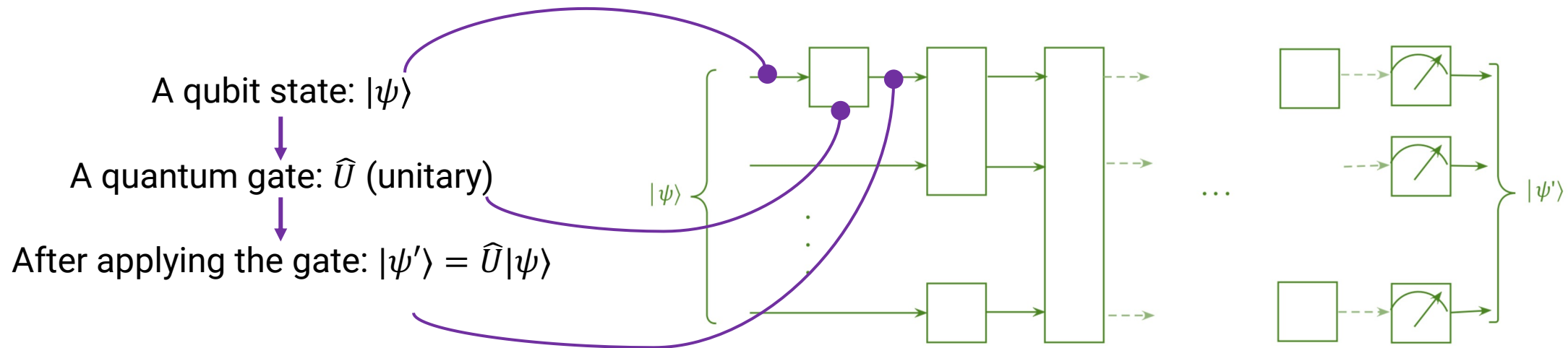
Topology of IBM Quantum Eagle R3 Processor with topology for 127 qubits.

$$|\psi\rangle = \alpha_{0\dots 00}|0\dots 00\rangle + \alpha_{0\dots 01}|001\rangle + \dots$$



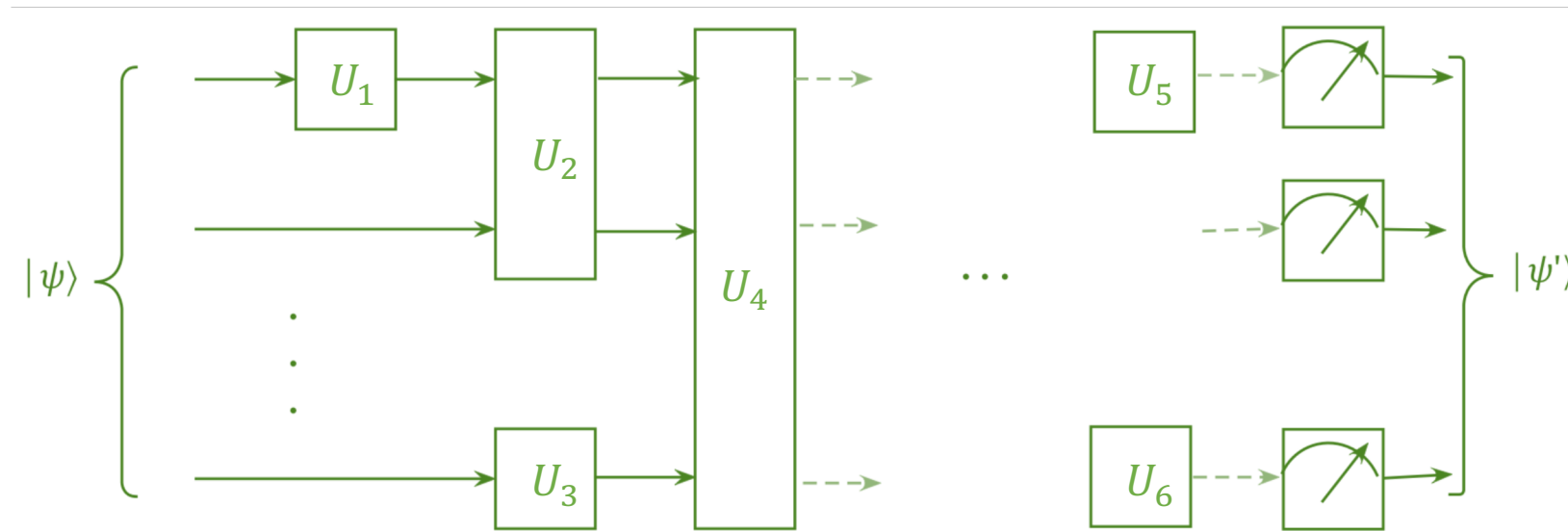
Gate Operations with Quantum Gates

- In quantum computing, **quantum gates** such as Hadamard, Pauli-X, or CNOT, are all represented by **unitary operations**
- These gates effectively apply $\hat{U}$ transformation to qubits, evolving their state over time in discrete steps:



Quantum Computing with Quantum Gates

- On a quantum computer, programs are executed by unitary evolution (quantum gates) of an input that is given by the state of the system, $|\psi\rangle$, to generate the next state, $|\psi'\rangle$.



- Since all unitary operators are invertible, we can always reverse or “uncompute” a computation on a quantum computer.

Single-Qubit Quantum Gates

Example single-qubit quantum gates:

Quantum Gate	Circuit Form	Matrix Form	Truth Table	Algebraic Properties
Identity gate (I)		$I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$	$ 0\rangle \mapsto 0\rangle$ $ 1\rangle \mapsto 1\rangle$	$X^2 = Y^2 = Z^2 = -iXYZ = I,$ $XY = -YX = iZ,$ $YZ = -ZY = iX,$ $ZX = -XZ = iY.$ $S^2 = Z$ $T^2 = S$ $TXT^\dagger = e^{-i\pi/4} SX,$ $H^2 = I,$ $X = HZH.$
Not gate (X)		$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$	$ 0\rangle \mapsto 1\rangle$ $ 1\rangle \mapsto 0\rangle$	
Y gate (Y)		$Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}$	$ 0\rangle \mapsto i\rangle$ $ 1\rangle \mapsto -i\rangle$	
Z gate (Z)		$Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$	$ 0\rangle \mapsto 0\rangle$ $ 1\rangle \mapsto - 1\rangle$	
Phase gate (S)		$S = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}$	$ 0\rangle \mapsto 0\rangle$ $ 1\rangle \mapsto i 1\rangle$	
T gate (T)		$S = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{pmatrix}$	$ 0\rangle \mapsto 0\rangle$ $ 1\rangle \mapsto e^{i\pi/4} 1\rangle$	
Hadamard gate (H)		$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$	$ 0\rangle \mapsto +\rangle$ $ 1\rangle \mapsto -\rangle$	



Two-Qubit Quantum Gates

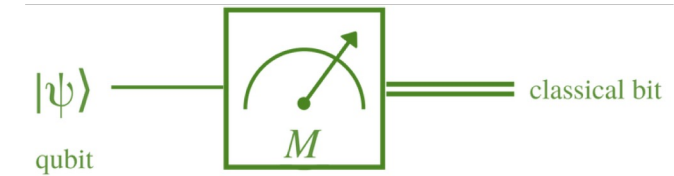
Multiple-qubit quantum gates can be expressed in a matrix form as well:

Quantum Gate	Circuit Form	Matrix Form	Truth Table
CNOT gate		$CNOT = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$	$ 00\rangle \mapsto 00\rangle$ $ 01\rangle \mapsto 01\rangle$ $ 10\rangle \mapsto 11\rangle$ $ 11\rangle \mapsto 10\rangle$
CZ gate		$CZ = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix}$	$ 00\rangle \mapsto 00\rangle$ $ 01\rangle \mapsto 01\rangle$ $ 10\rangle \mapsto 10\rangle$ $ 11\rangle \mapsto - 11\rangle$

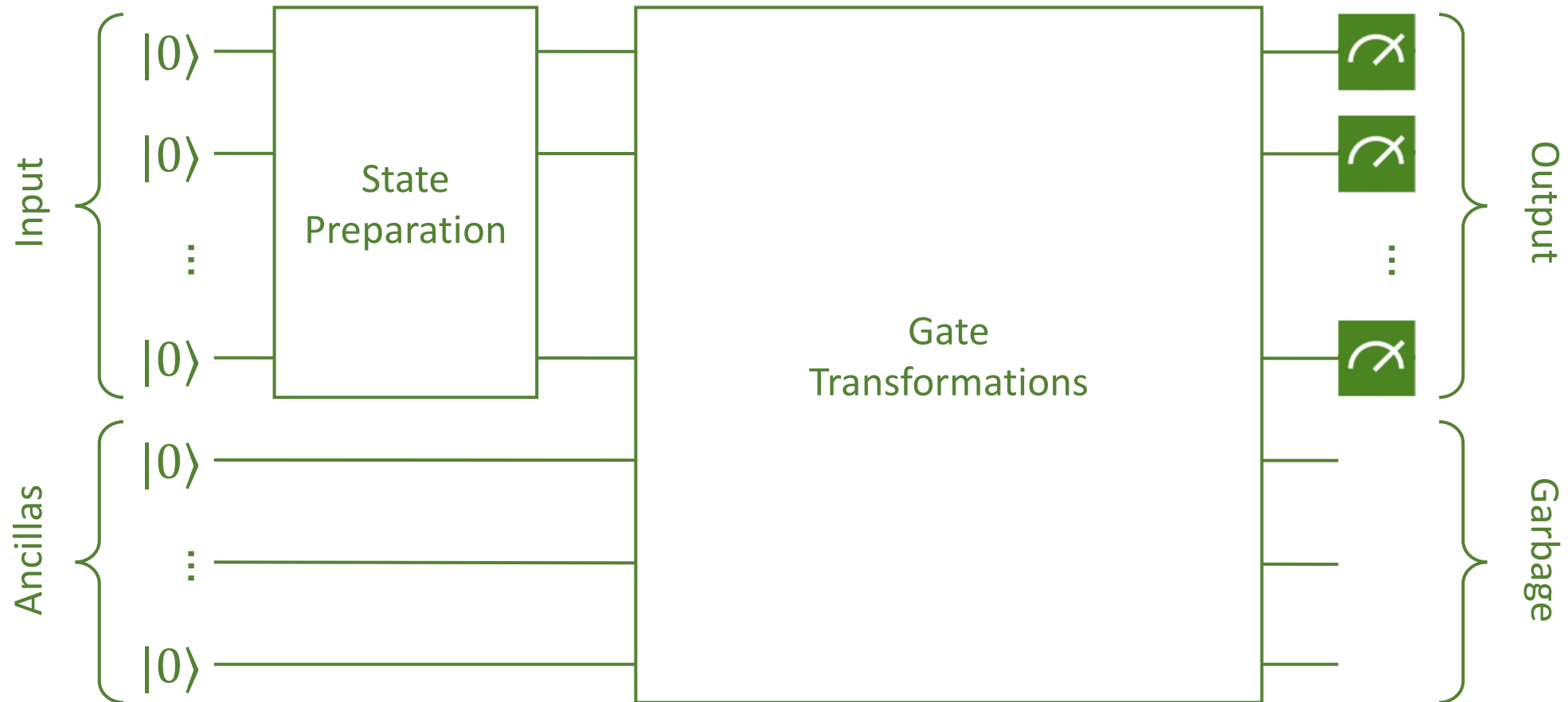


Measurement and Quantum Computation Outputs

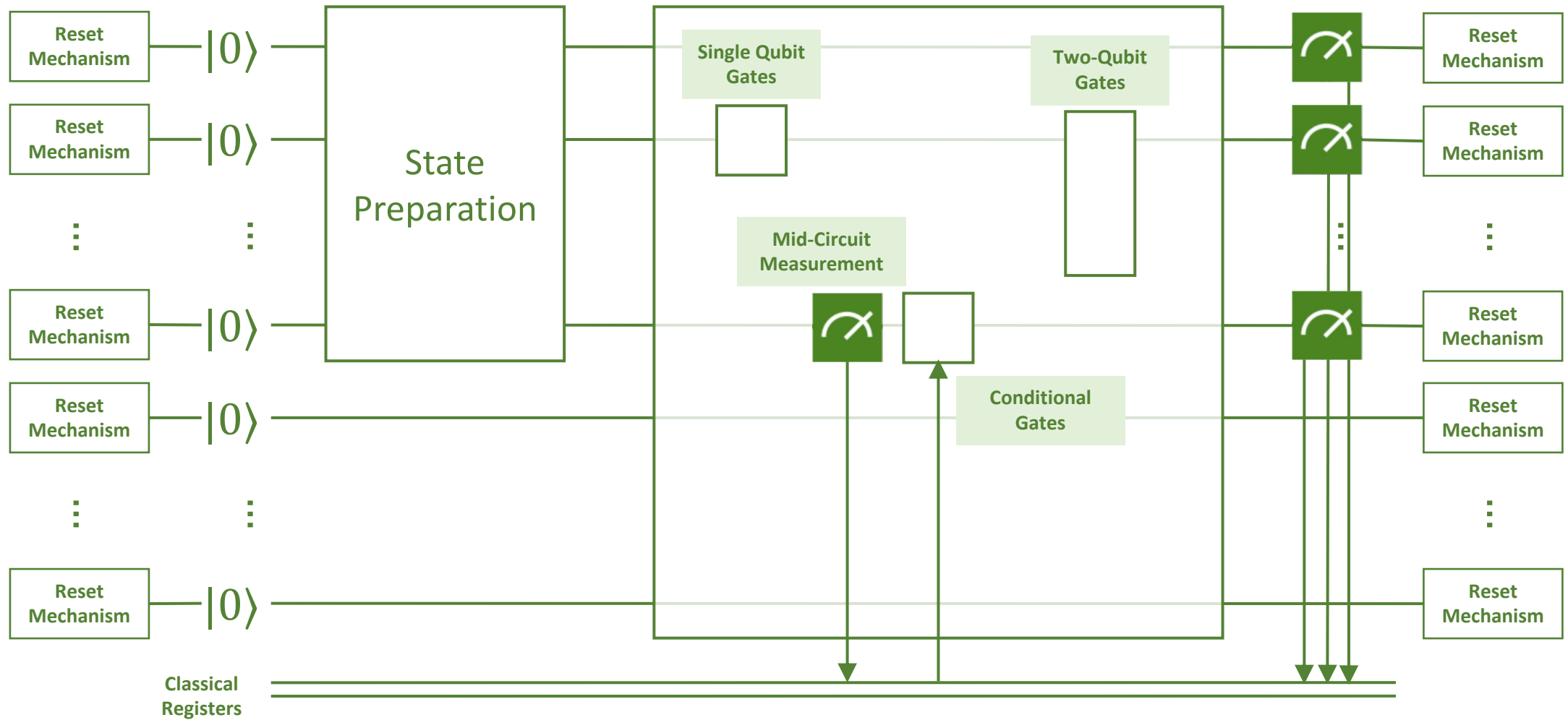
- If a quantum system were perfectly isolated, it would maintain coherence indefinitely, but it would be impossible to manipulate or read it out.
- A quantum measurement is a destructive process.
- When a quantum system is measured, the wave function $|\psi\rangle$ collapses to a new state according to probabilistic rules.
- If $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ after measurement, either $|\psi\rangle = |0\rangle$ or $|\psi\rangle = |1\rangle$, and these possibilities occur with probabilities of $|\alpha|^2$ and $|\beta|^2$ with $|\alpha|^2 + |\beta|^2 = 1$.
- A quantum measurement never produces another quantum state $|\psi\rangle = \alpha'|0\rangle + \beta'|1\rangle$



Putting It Together: A Typical NISQ Quantum Circuit



A Typical Quantum Circuit with More Details



Imperfections, Noise and Errors in Quantum Circuits

Some sources of noise and errors in NISQ quantum computers:

- **Imperfect gate operations** – variations in control pulses (timing, amplitude, phase) lead to gate operations not executing fully as expected
- **Crosstalk** – manipulating one qubit can affect another qubit
- **Leakage Errors** – qubits can transition into states outside the logical $|0\rangle$ and $|1\rangle$ space, e.g., in multi-level systems like superconducting qubits
- **Readout errors** – Imperfect measurement processes cause incorrect detection of qubit states



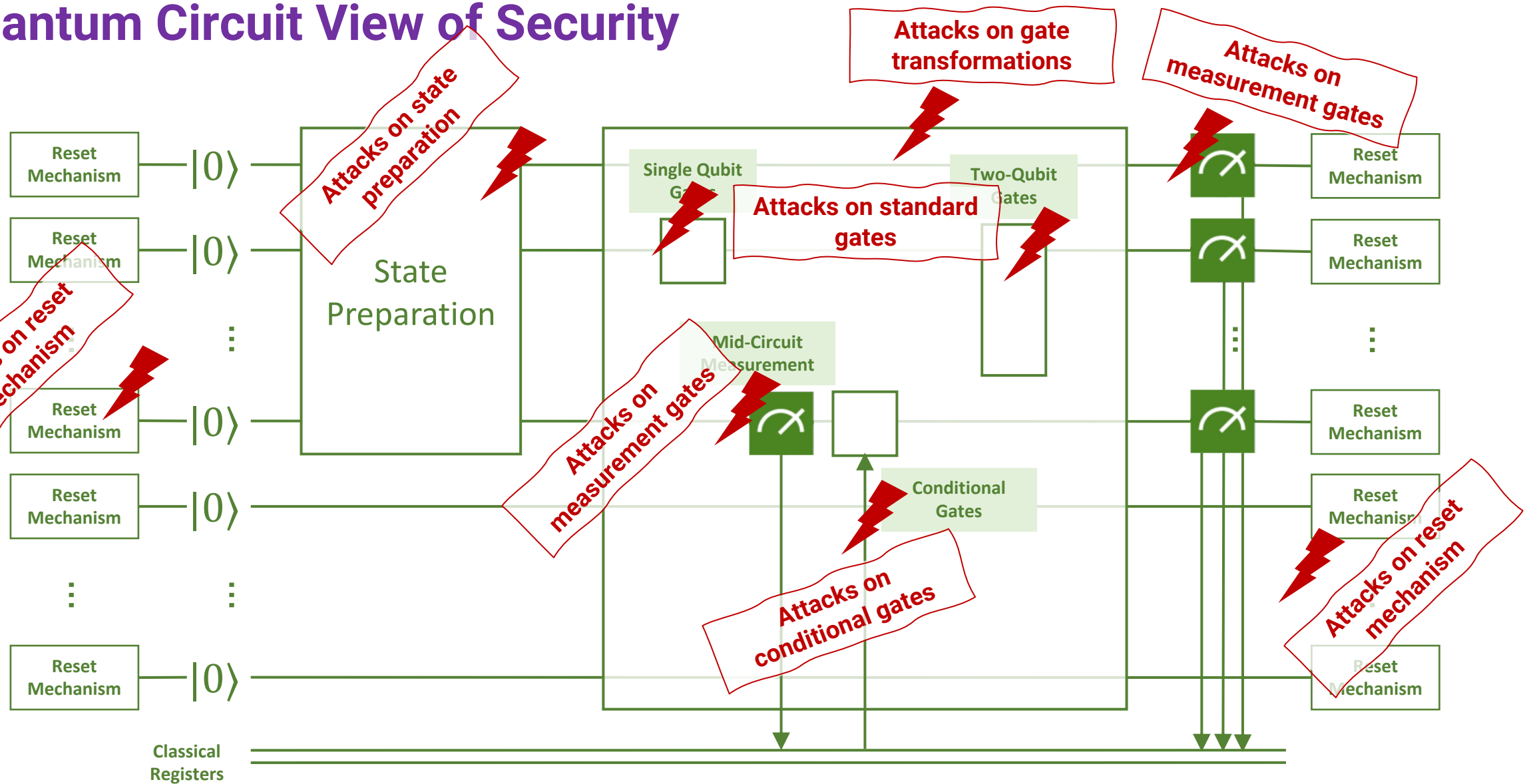
Security Implications of Noise and Errors in Quantum Circuits

Attackers can leverage the noise and errors in today's NISQ quantum computers:

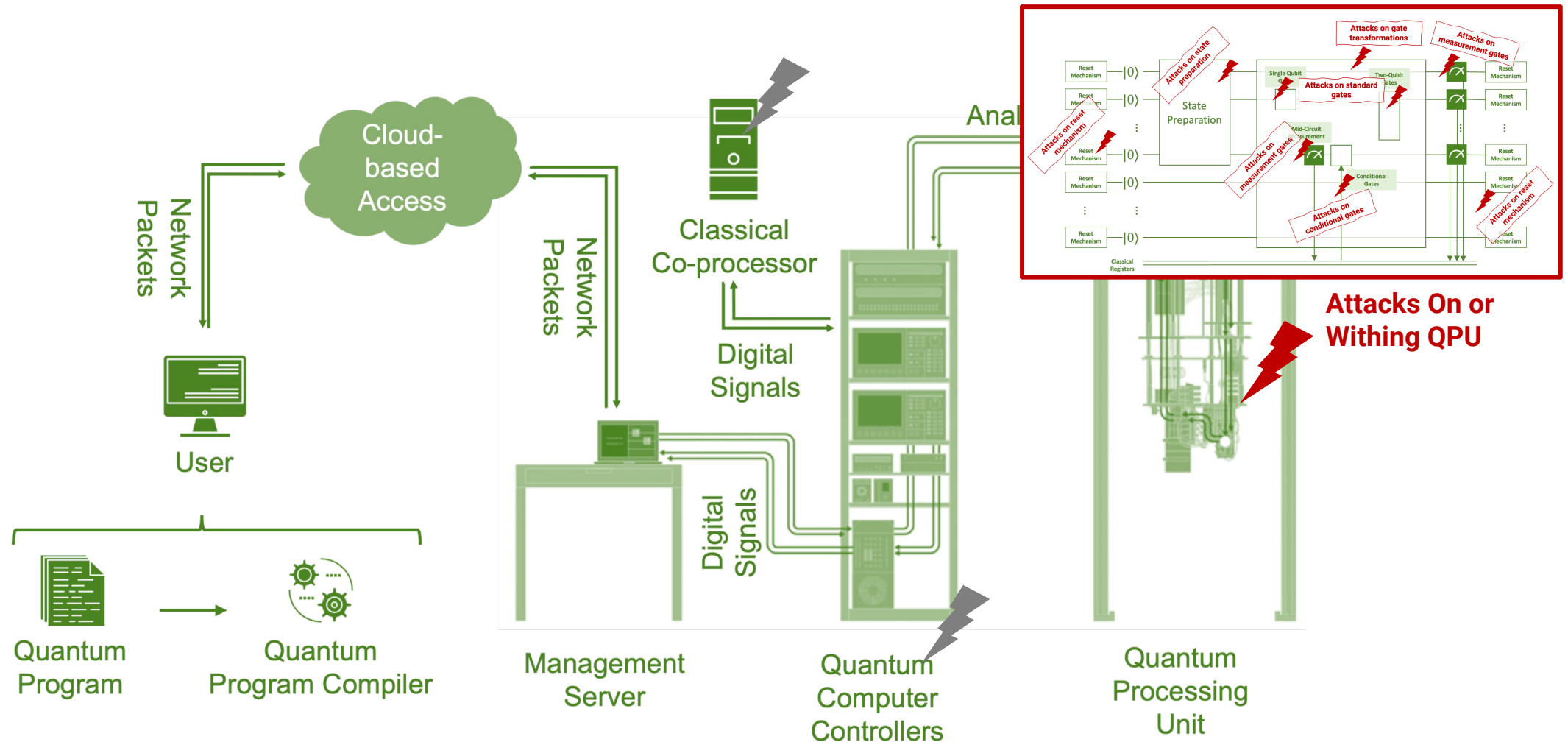
- **Imperfect gate operations** – side channels or fault injection
- **Crosstalk** – side channels or fault injection
- **Leakage Errors** – fault injection
- **Readout errors** – side channels or fault injection



Quantum Circuit View of Security



Crosstalk and Fault Injection in NISQ Quantum Computers



Aside: Security Treats to Fault-Tolerant Quantum Computing (FTQC)

NISQ



FTQC

(Noisy Intermediate-Scale Quantum)

refers to today's quantum computers, which have roughly 100s – 1000s of qubits but are too error-prone for long computations. They have some promise in quantum chemistry, optimization, quantum machine learning or quantum simulation.

(Fault-Tolerant Quantum Computing)

uses error-correcting codes to protect logical qubits from noise, enabling arbitrarily long computations at the cost of need for 100s to 1000s of physical qubits per logical qubit. FTQC can run any quantum circuit, but not every circuit may run efficiently.



Logical Qubits and Quantum Error Correction

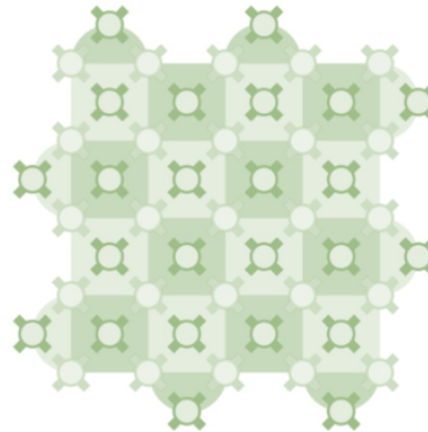
To make qubits reliable FTQC quantum computers group many physical qubits into one logical qubit and apply quantum error correction to detect and correct errors.

A surface code is popular error correction code used in FTQC:

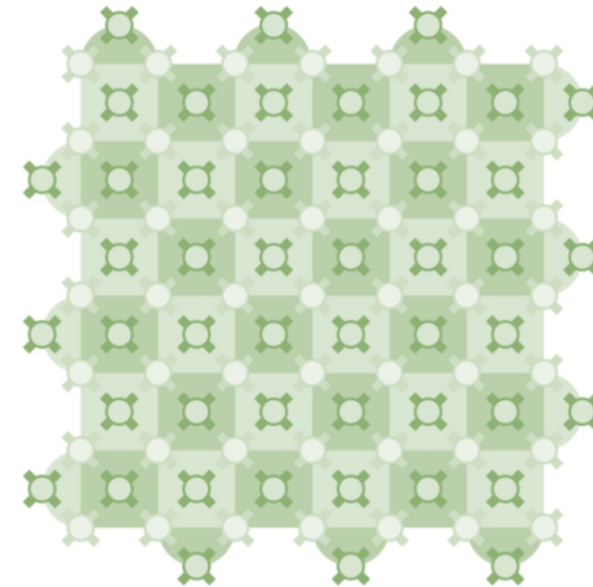
A patch that
encodes one
logical qubit.



3x3



5x5



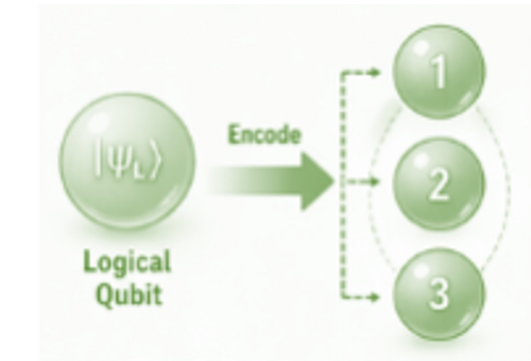
7x7

Data qubits
and
Ancilla (or
stabilizer)
qubits make
up a patch



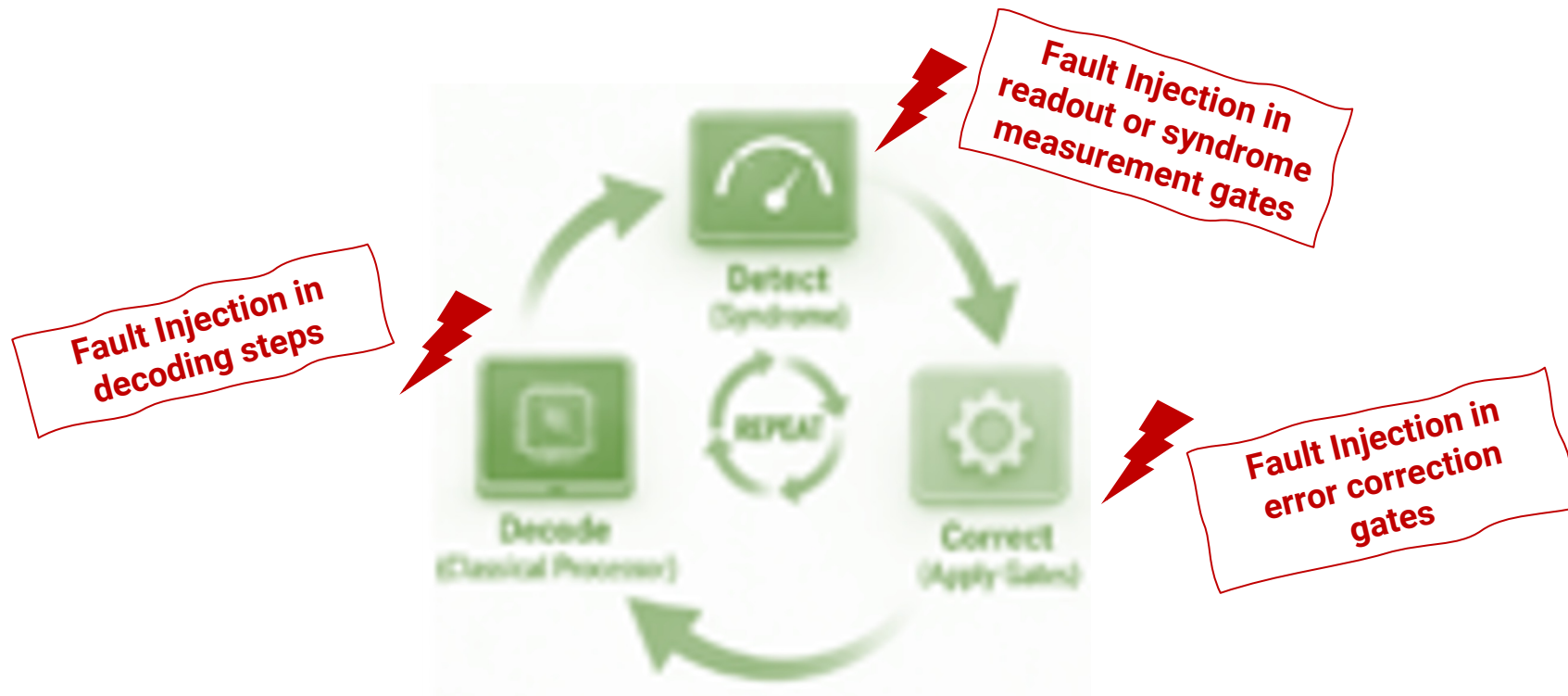
Error Correction Steps

1. **One logical qubit is encoded into many physical qubits**
2. **(If computing, quantum logical gates are applied to the physical qubits in a patch)**
3. **Ancilla qubits are measured to check correlation** and obtain syndrome measurement
4. **Classical computer analyzes syndrome to detect most likely error**, typically X or Z quantum gate is then used to flip the qubit where the error occurred
5. **Repeat the cycle:** Detect – Decode – Correct



Error Correction Cycle and Threats

Error correction cycle is vulnerable to various possible threats:



Imperfections, Noise and Errors in FTQC Quantum Circuits

Some sources of noise and errors in FTQC quantum computers:

- **Physical qubit and gate problems** -- same as in NISQ, physical gates and qubits are noisy and not perfectly execute
- **Decoding algorithm** -- decoding algorithm and error correction code may not account for certain types of errors
- **Failures in classical hardware** – faults in classical decoder can exist and affect execution



Security Implications of Noise and Errors in FTQC Quantum Circuits

Attackers can leverage imperfections in FTQC quantum computers:

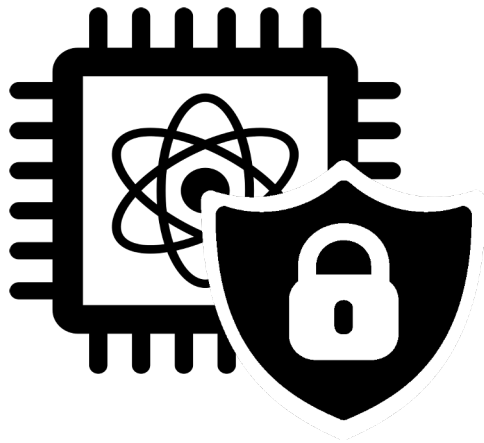
- **Physical qubit and gate problems** -- fault injection and side channels
- **Decoding algorithm** -- fault injection and side channels
- **Classical hardware** – fault injections and side channels



Tutorial Outline

- **1:00pm – 1:30pm: Introduction and Motivation**
 - What is Quantum Computing and Security?
 - Brief History of Quantum Computing and Motivation for Security Research
 - Introduction to Quantum Computation and Security Threats
- **1:30pm – 2:30pm: Crosstalk and Fault Injection in NISQ Quantum Computers**
- **2:30pm – 3:00pm: Coffee Break**
- **3:00pm – 3:45pm: Side Channels in NISQ and FTQC Quantum Computers**
- **3:45pm to 4:30pm: Trusted Execution Environments for Quantum Computers**
- **4:00pm: End**





Short Aside: Using Quantum Computers as Attackers



Public Key Cryptography

Public key cryptography uses different encryption (or signing) and decryption (or verification) keys:

- No need for a shared secret key between sender and receiver.
- Anybody with knowledge of public encryption key can encrypt and send message to receiver who has corresponding secret key for decrypting the message.
- Anybody with knowledge of public verification key can verify digital signatures signed by the party with the corresponding secret signing key.



Shor's Algorithm

Shor's algorithm is famous for its potential to break widely used cryptographic systems based on **integer factorization** (like RSA) and **discrete logarithms** (used in DSA, ECDSA, and DH):

- Shor's use for **integer factorization** threatens RSA-based signatures.
- Shor's use for **discrete logarithms** threatens DSA, ECDSA, and ElGamal-based systems.

Shor's algorithm works with sufficient number of noise-less qubits, which is a big limitation and not possible on today's quantum computers yet, although there are constant advances.



Parameters and Security of the RSA Algorithm

RSA is a family of public-key cryptosystems that can be used for public-key encapsulation and digital signatures:

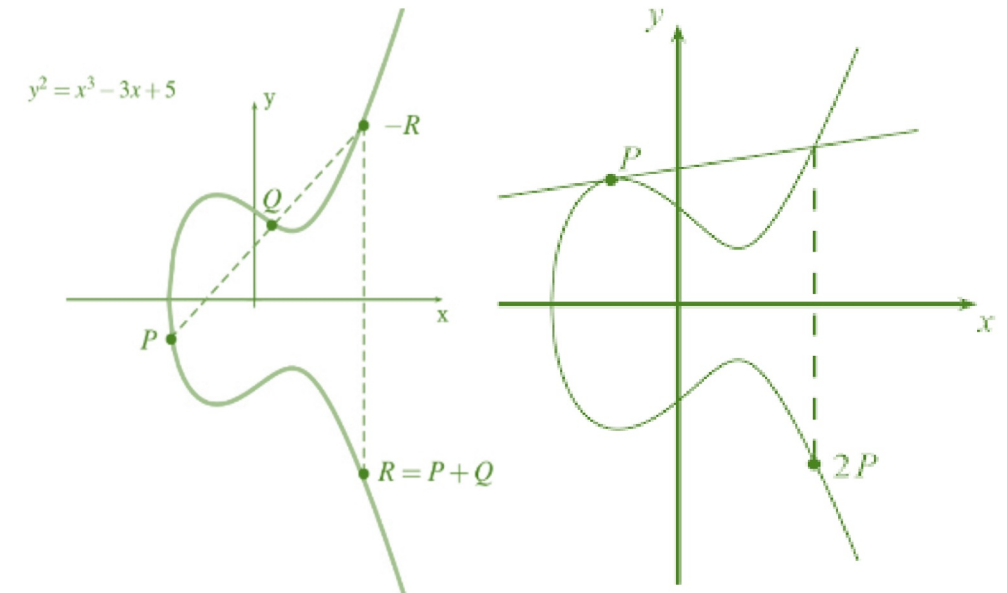
- The security of RSA is based on the difficulty of factoring the product of two large prime numbers.
- The keys in RSA are based on modulus $n = p * q$ where p and q are the primes. **The p and q are secret while n is part of the public key.**
- RSA can have different key sizes, typically from RSA-2048 with 2048-bit keys to RSA-4096 with 4096-bit keys, here the number of key bits N is the size of n .



Parameters and Security of the ECC Algorithm

ECC (Elliptic Curve Cryptography) is a family of public-key cryptosystems that can be used for public-key encapsulation and digital signatures as well:

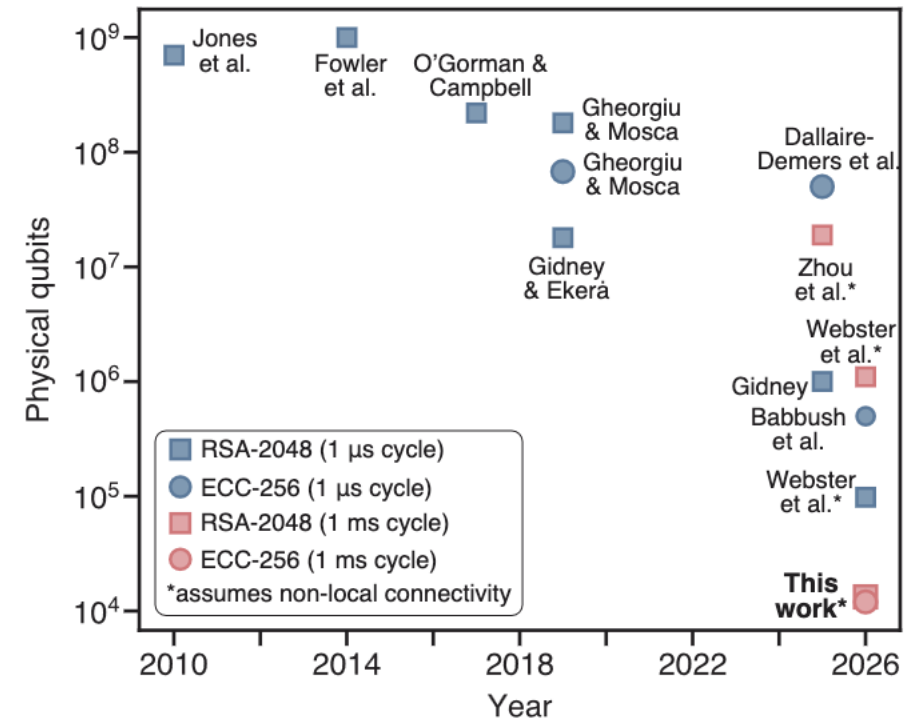
- The security of ECC is based on the difficulty of the Elliptic Curve Discrete Logarithm Problem (ECDLP).
- The keys in ECC are based on a curve equation, typically of the form $y^2 = x^3 + ax + b$. **The private key is a secret random integer k while the public key is a generator $G(x,y)$ and point on the curve derived from it $Q(x,y) = G(x,y) * k$**
- ECC can achieve the same level of security as RSA with much smaller key sizes; for example, a 256-bit ECC key provides security equivalent to an RSA-3072 key.



Quantum Needs to Break RSA and ECC with Shor's Algorithm

Current noisy (NISQ) quantum computers are too noisy to run Shor's algorithm correctly, and **Fault-Tolerant Quantum Computers (FTQC) with logical qubits are needed for Shor's:**

- In FTQC each logical qubit is built using many physical qubits, about 100 to 1000 physical qubits are needed per logical qubit to get very low error rates
- If $N=2048$ then you need quantum computer with $\sim 400,000$ to $\sim 4,000,000$ physical qubits
- But there are advances every year in improving factoring and Shor's



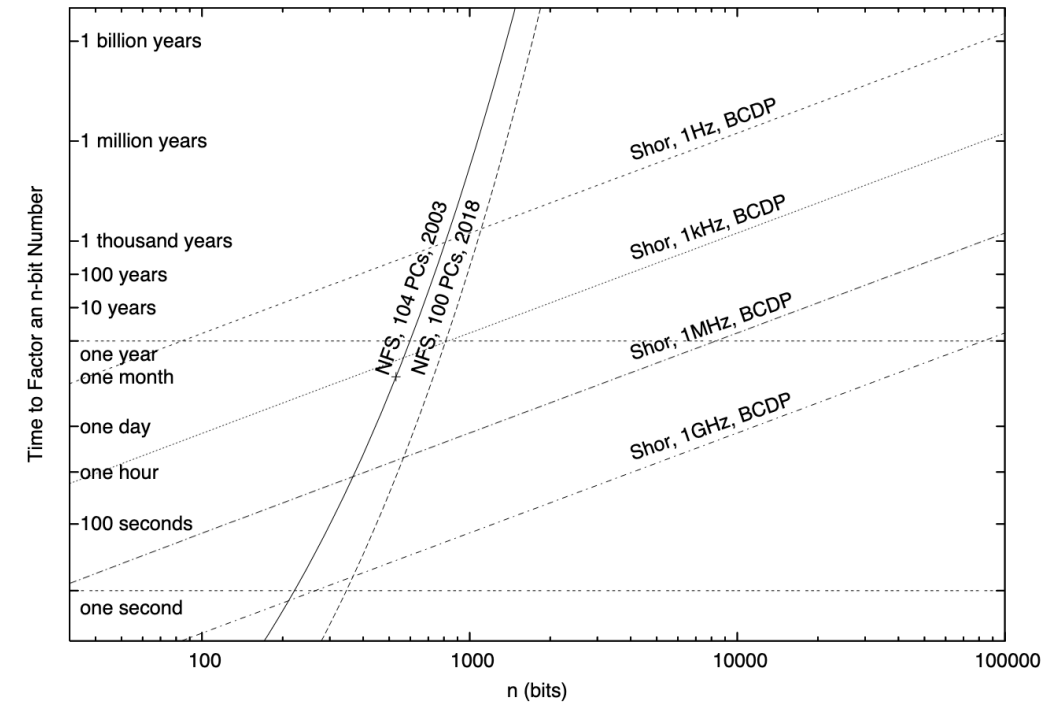
Cain et al., 2026



Quantum Needs to Break RSA and ECC with Shor's Algorithm

In addition to the large qubit count, **Shor's algorithm has a very large depth, requiring possibly long execution time to do the factoring:**

- The depth of the Shor's circuit is about $O(N^3)$, for $N=2048$ that means the circuit depth is about 8^9 gates to execute
- Assuming 1us gate cycle (1MHz) that is about 2 hours while assuming 1ms (1kHz) gate cycle time that is about 99 days (Yamaguchi et al., 2023)
- Typical logical qubit cycle times:
 - Physical gate times are about 50ns
 - Logical gate times are about 5ms



Van Meter et al., 2005



New Types of Public Key Algorithms

To come ahead of the threat to RSA and existing algorithms, NIST (National Institute of Standards and Technology) has been running various processes (competitions) to find new algorithms.

- Find algorithms for key encapsulation and digital signatures that are not known to be affected by Shor's or other algorithms
- Public process to ensure transparency, long (few years) public review process
- Standardized algorithms will replace RSA, ECC, etc.
- **These are called the Post-Quantum Cryptographic (PQC) algorithms, many are already standardized and government and industry are pushing to switch to PQC by 2029 or 2030.**



Take Away Message

Although Shor's algorithm poses a threat to classical computers, it requires a lot of quantum computer resources. In the meantime, new PQC algorithms are being deployed today that are not known to be vulnerable to attacks by Shor's algorithm

Hence, hopefully, more research (and this tutorial) will focus on security of quantum computers themselves, as threats of Shor's algorithm are understood and being anticipated

