UMBC

Emerging Software Technologies

Software system security in the era of quantum computing

IS 436 Spring 2026

Lei Zhang
leizhang@umbc.edu

AN EXAMPLE TO INTRODUCE QUANTUM MECHANICS

A gambling game: coin flip

The player calls "heads" or "tails"

The coin is flipped

The coin has landed

If the player's call matches the landing side

The player wins!

Otherwise

The house wins!

Q: What is your **chance** of winning?

Demo

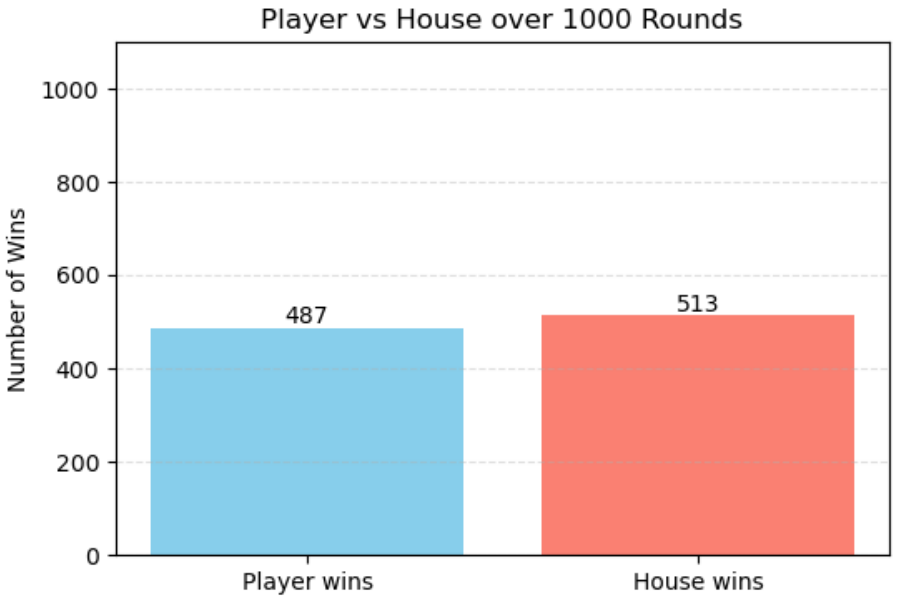
- https://github.com/zhangl64/flipcoin
- Run python classical-game.py

UMBC

All Possible Outcomes

Case	Your Call	Coin Result	Outcome
1	Heads	Heads	✅ Win
2	Heads	Tails	❌ Lose
3	Tails	Heads	❌ Lose
4	Tails	Tails	✅ Win

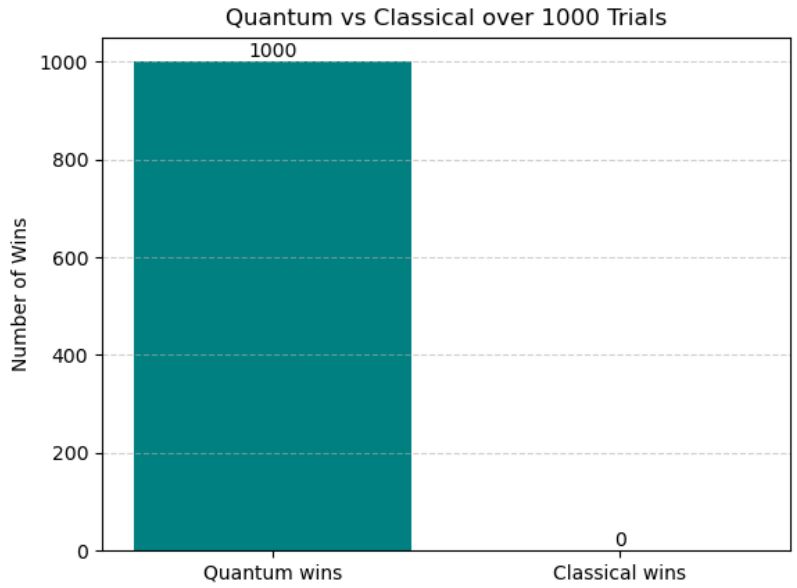
Simulate 1,000 times

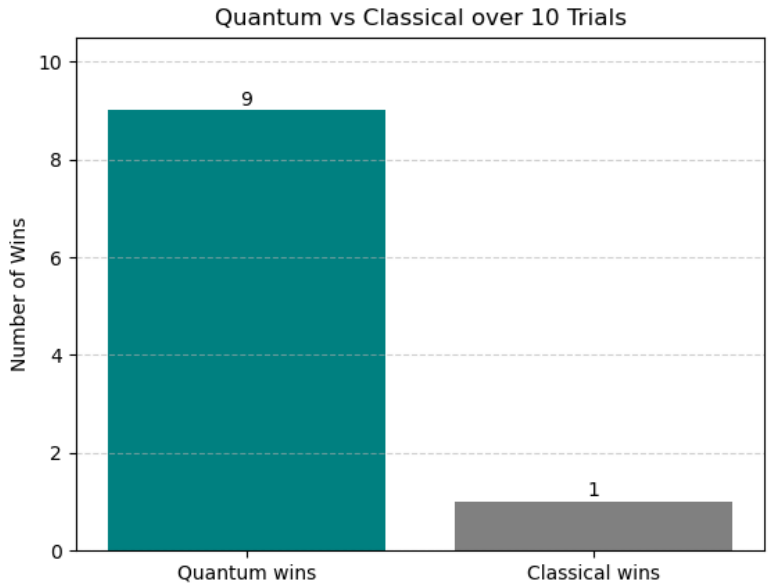


Demo on a quantum device/simulator

- https://github.com/zhangl64/flipcoin
- Run “python quantum-game.py”
- Optional: try on a real quantum computer by running ”python flipcoin-ibmq.py”

Simulate on a quantum simulator/machine





We cheat ☺!

The player calls "heads" or "tails"

The coin is in superposition

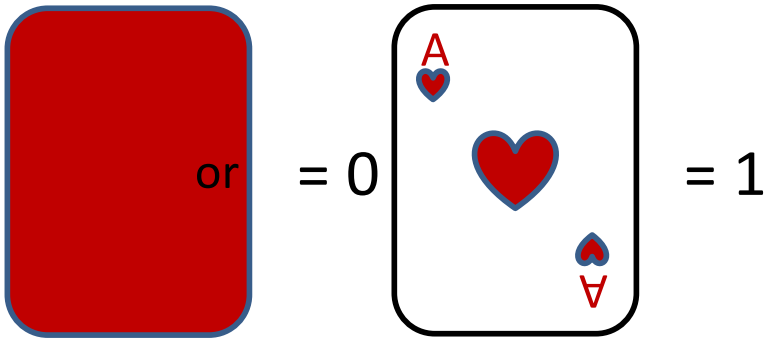
The quantum coin is flipped

The coin collapses in control

The quantum coin has landed

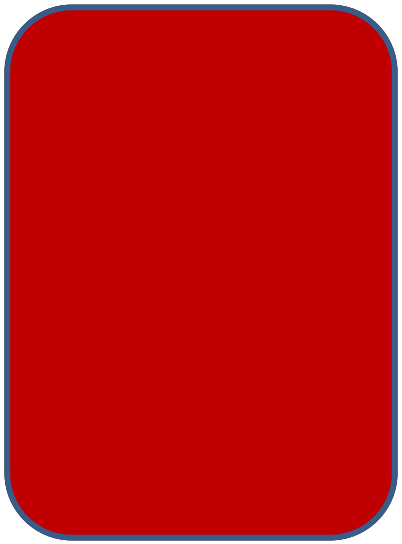
The house ALWAYS wins!

Qubits: a gentle introduction



A classical bit can only represent 0 or 1 at a time

Qubits in superposition

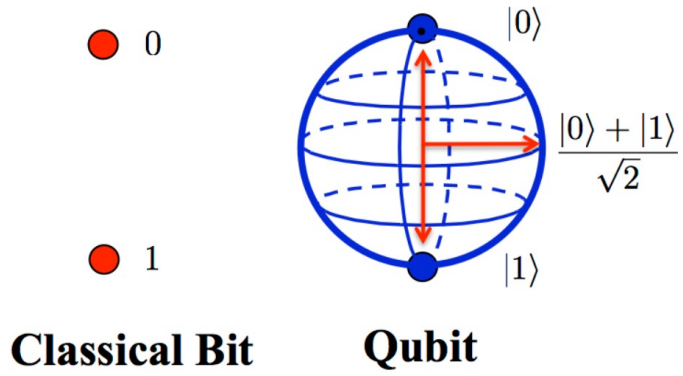


= {0, 1}

1 qubit can represent 0 and 1 at the same time, i.e., “superposition”.

Qubits

- A classical bit can take the value of 0 or 1.
  - A register of n bits can be one of 2^n states at a time.
- A qubit can be captured as a **superposition**
  - A register of n qubits can be 2^n different states.

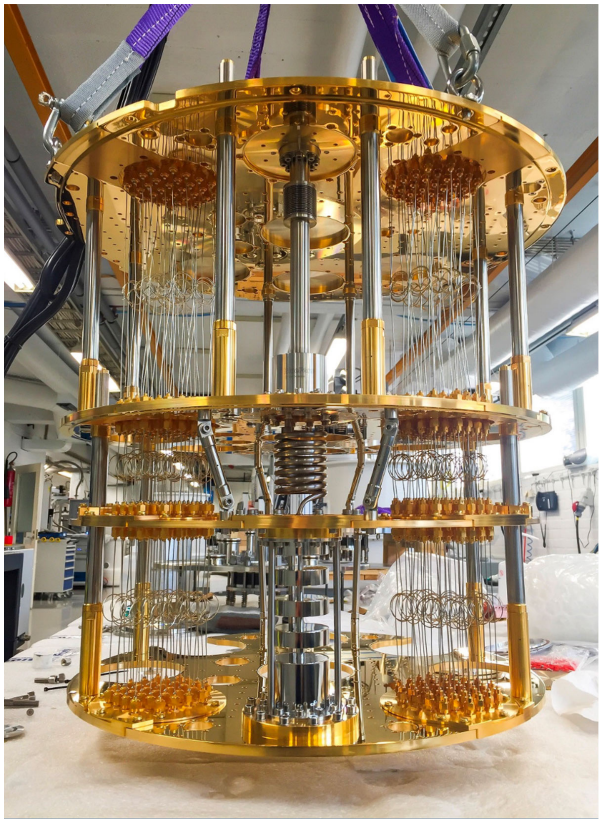


Source: Poetry in Physics

Quantum History and Current Status

Fundamentals–what is quantum computing?

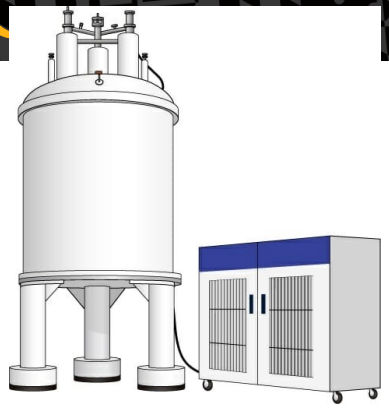
Quantum computing is the use of quantum mechanics (such as **superposition**, **entanglement**, and **interference**) to perform computation.



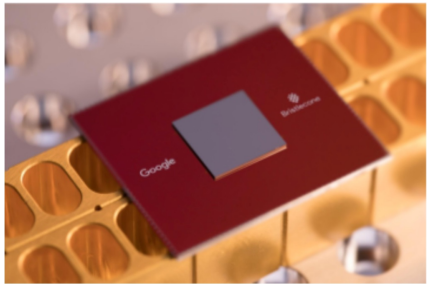
Source: Microsoft

UMBC

Timeline



First quantum computer (2-qubit)



IBM Condor: 1,121-qubit quantum computer

1982 — First quantum computing model

1998 — First quantum computer (2-qubit)

18 years

2016 — IBM: 5-qubit quantum computer

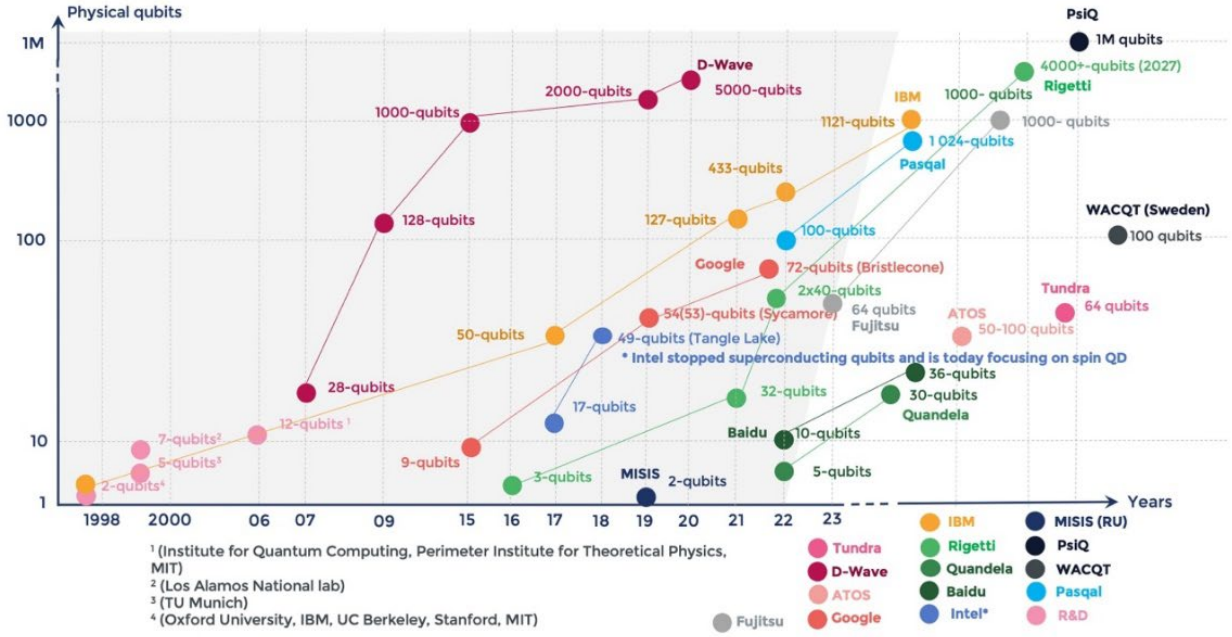
9 years

2025 — IBM Condor: 1,121-qubit quantum computer

? — Quantum Advantage

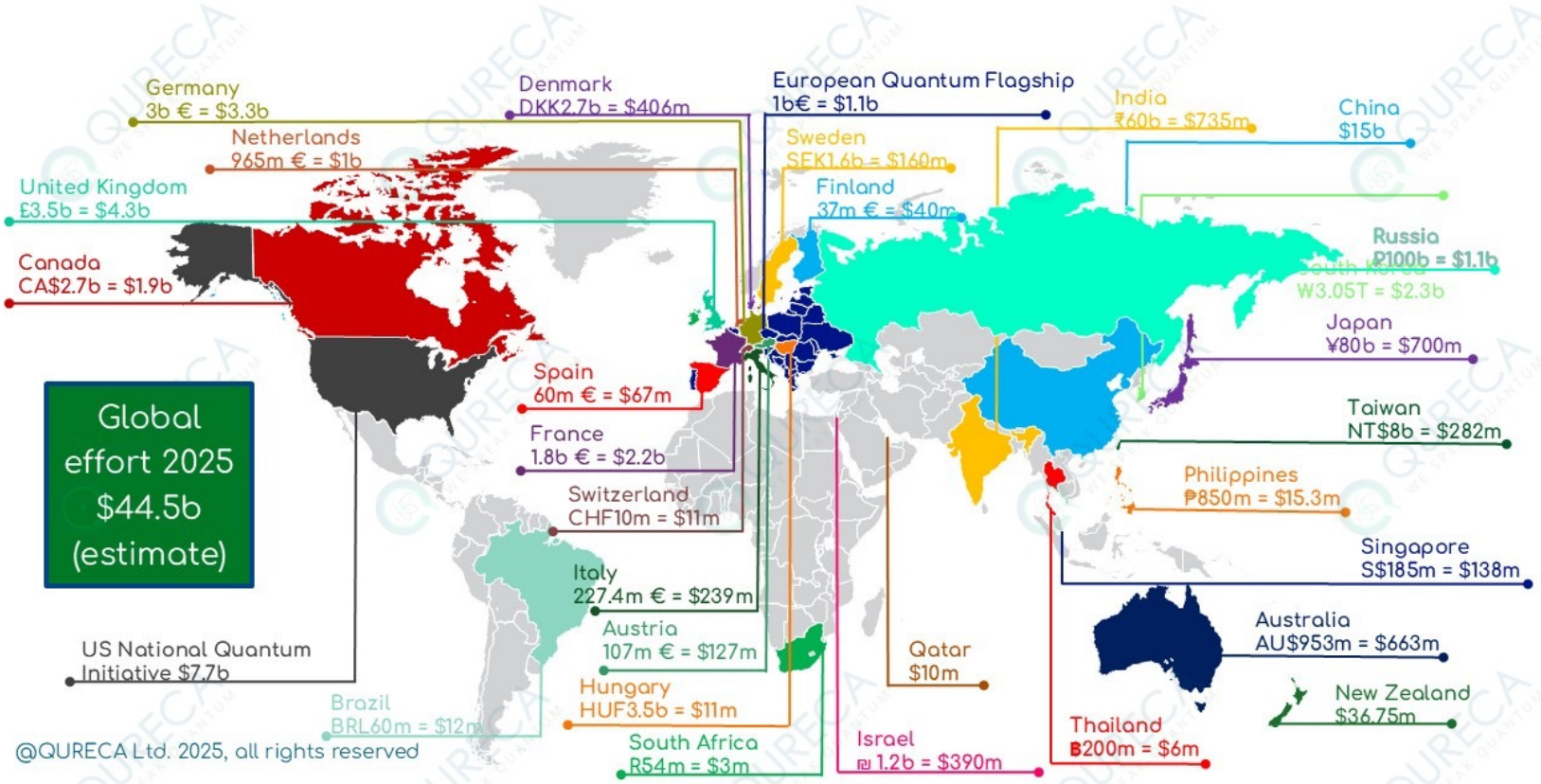
UMBC

Current state of quantum computing

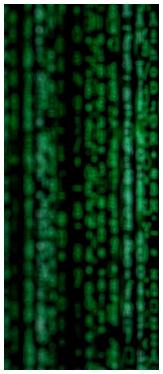


Quantum computing is an emerging technology with incredible potential, but it also comes with major challenges.

UMBC


Germany
3b € = $3.3b
Denmark
DKK2.7b = $406m
European Quantum Flagship
1b€ = $1.1b
India
₹60b = $735m
China
$15b
Netherlands
965m € = $1b
Sweden
SEK1.6b = $160m
United Kingdom
£3.5b = $4.3b
Finland
37m € = $40m
Russia
₽100b = $1.1b
Canada
CA$2.7b = $1.9b
South Korea
W3.05T = $2.3b
Japan
¥80b = $700m
Spain
60m € = $67m
Taiwan
NT$8b = $282m
France
1.8b € = $2.2b
Philippines
₱850m = $15.3m
Switzerland
CHF10m = $11m
Singapore
S$185m = $138m
Global effort 2025 $44.5b (estimate)
Italy
227.4m € = $239m
Australia
AU$953m = $663m
Austria
107m € = $127m
Qatar
$10m
US National Quantum Initiative $7.7b
Hungary
HUF3.5b = $11m
New Zealand
$36.75m
Brazil
BRL60m = $12m
Thailand
฿200m = $6m
South Africa
R54m = $3m
Israel
₪ 1.2b = $390m
@QURECA Ltd. 2025, all rights reserved

Quantum Advantage



Applications:

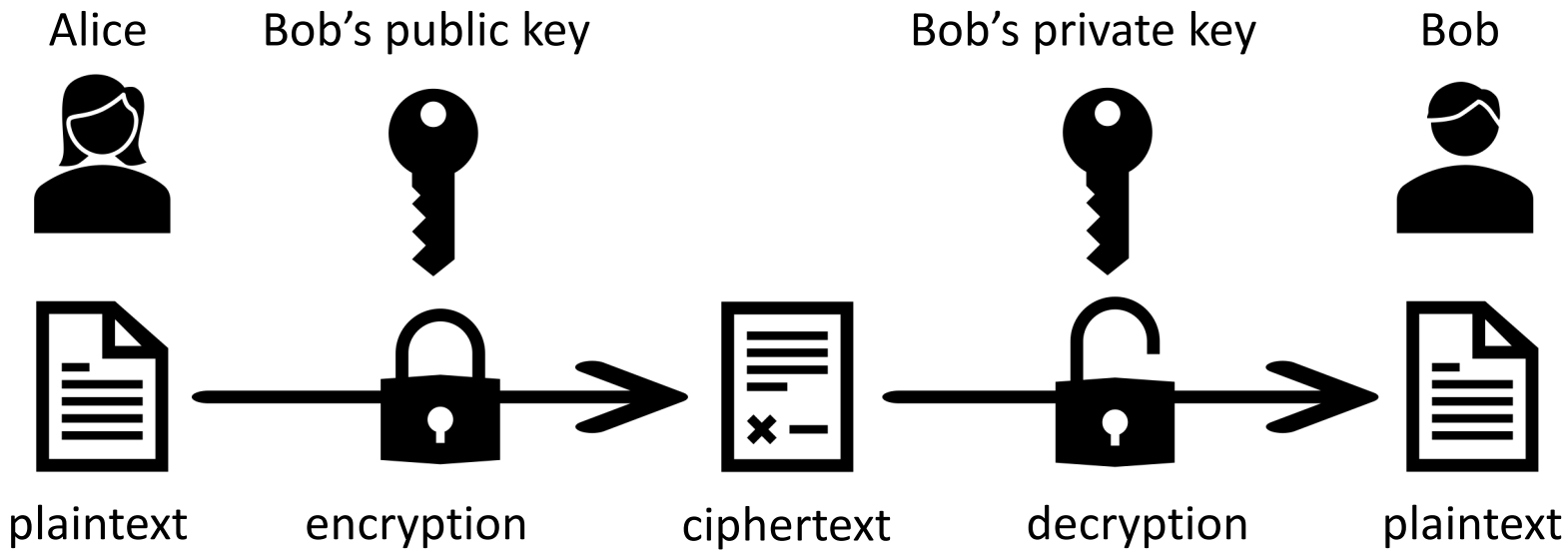
- **Cybersecurity**
- Artificial Intelligence
- Computational Chemistry
- Drug Design & Development
- …

Classical Crypto

Review

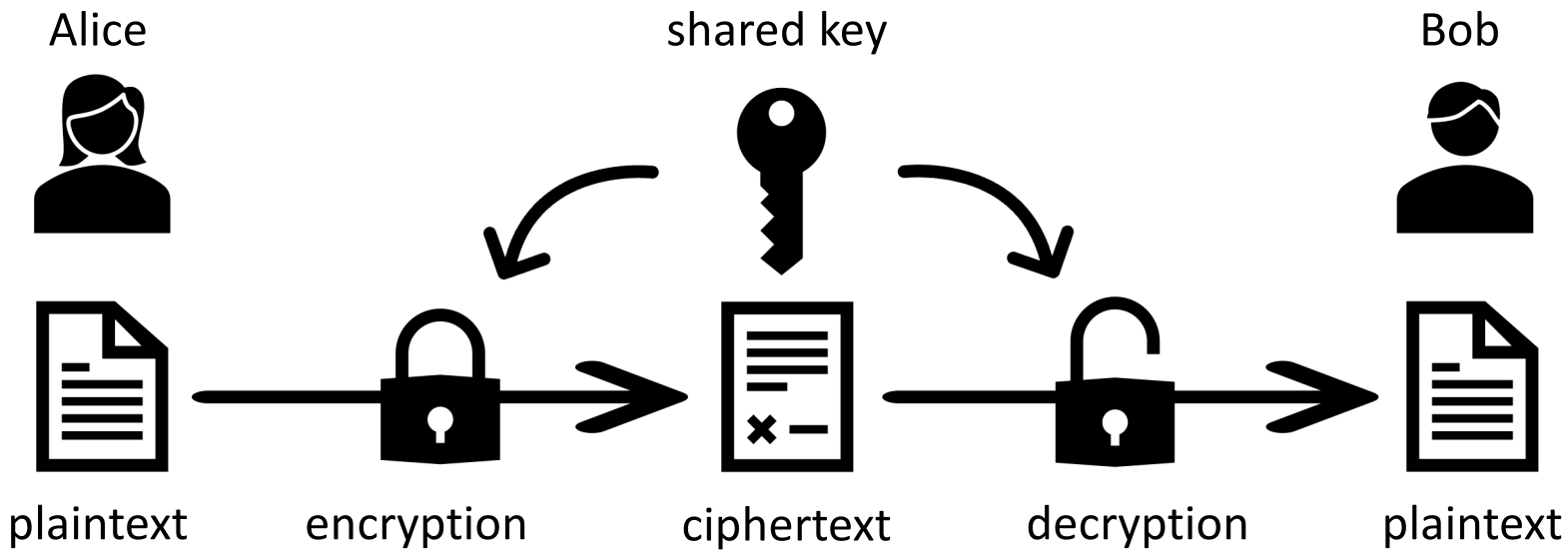
- What is public-key (asymmetric) encryption?
- Can you name some asymmetric algorithms?
- What is private-key (symmetric) encryption?
- Can you name some symmetric algorithms?

Public key encryption (RSA, ECC and DH)



Asymmetric (public key) cryptography

Private key encryption (AES and 3DES)



Symmetric (private key) cryptography

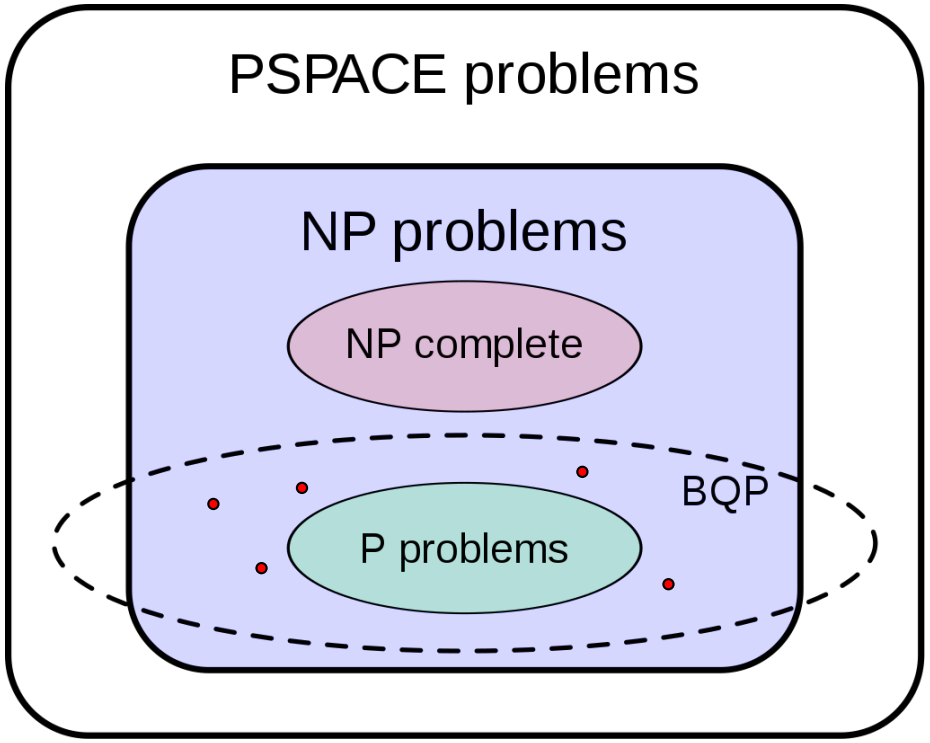
Public- vs Private-key encryption

1. **Speed:** Symmetric encryption is generally faster, as it requires less computational power.
2. **# of keys:**
   - Symmetric: single shared key;
   - Asymmetric: a pair of keys.
3. **Key distribution:**
   - Symmetric: secure key distribution is crucial, as the same key is used for both encryption and decryption;
   - Asymmetric: simple key distribution, as only the public key needs to be shared.
4. **Applications:**
   - Symmetric: secure communication and file encryption;
   - Asymmetric: digital signature and key exchange.

User authentication

Quantum Crypto

Relationship between quantum computing and classical computational complexity



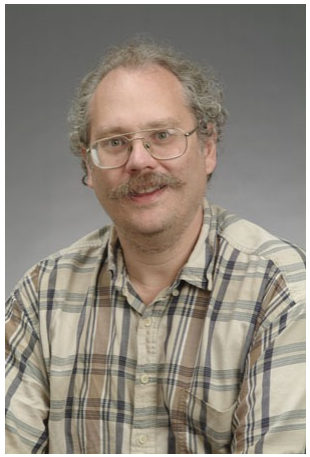
BQP: bounded-error quantum polynomial time

Source: Michael Nielsen and Isaac Chuang (2000). Quantum Computation and Quantum Information.

Integer factorization & Shor's algorithm

1459067680075833232301869393490706352924018723753571643995818710198734
3879900535893836957140267014980212181808629246742282815702292207674690
6543401224889672472407926969987100581290103199317858753663710862357656
5105078837142971156373427889114635351027120327651665184117268598379886
7211183720508552634661874005 3

- Problem: given an integer N, find its prime factors (**integer factorization**), e.g., $15 = 3 \times 5$.
- RSA scheme (public key -> private key)
- The best classical algorithm has complexity $O(e^{1.9(logN)^{1/3}(loglogN)^{2/3}})$ – sub-exponential
- Shor's algorithm can solve it in BQP time $O((logN)^2(loglogN)(logloglogN))$



Peter Shor

UMBC

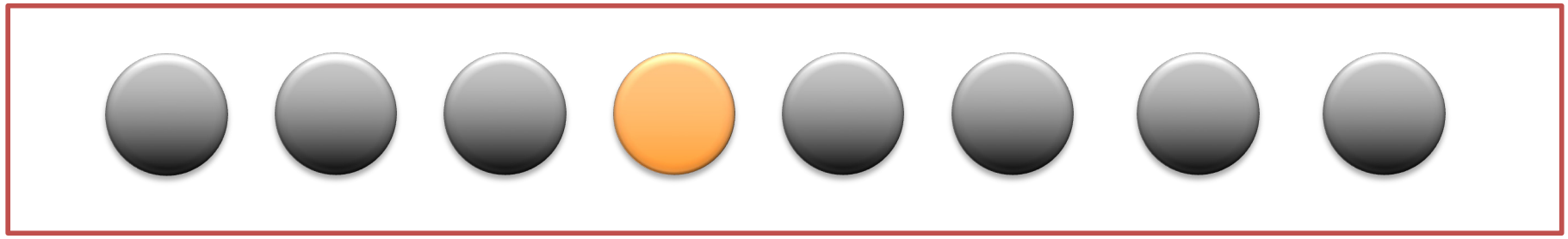
Practice

- https://github.com/zhangl64/qiskit-shor/blob/main/prime_factorization.py
- Download the code and test it with multiple numbers
  1. 15 = 3 x 5
  2. 50,000,167,000,051= 50,000,017 x 1,000,003
  3. 167,700,407,018,119 = 50,000,017 x 3,354,007
  4. 6,763,011,116,220,131 = 68,778,473 x 98,330,347(**finish at home**)
- Command: time python prime_factorization
- Windows, use
  - time /t
  - python your_script.py
  - time /t
- 11,346,317 = 3,431 x 3,307
  = 47 x 73 x 3,307

Shor's algorithm in steps

1. Given N, check that N is not a prime; otherwise, stop
2. Choose a random integer $1 < a < N$
3. Compute $b = gcd(a, N)$, if $b > 1$, return b and stop; otherwise,
4. **Find the order of $a \bmod N$, that is $r > 0$ such that $a^r \equiv 1 \bmod N$**
5. If r is odd, go to Step 2; otherwise,
6. Compute
   1. $x = a^{\frac{r}{2}} + 1 \bmod N$
   2. $y = a^{\frac{r}{2}} - 1 \bmod N$
7. If $x = 0$, go to Step 2; if $y = 0$, take $r = \frac{r}{2}$ and go to Step 5
8. Compute $p = gcd(x, N)$ and $q = gcd(y, N)$. At least one of them will be a non-trivial factor of N

Grover's algorithm, brute-force



- Complexity: O(N)

UMBC

Grover's algorithm, simplified

1. Represent each ball as a quantum state. How many qubits do we need for 8 balls?
2. Put all qubits in superposition:
   - (1/2) * (|000⟩ + |001⟩ + |010⟩ + |011⟩ + |100⟩ + |101⟩ + |110⟩ + |111⟩)
   - Each state has a probability of 1/8.
3. Apply the oracle function, which flips the sign of the state that represents the golden ball:
   - (1/2) * (|000⟩ + |001⟩ + |010⟩ + |011⟩ + |100⟩ **- |101⟩** + |110⟩ + |111⟩)
4. Apply the Grover operator, which amplifies the amplitude of the state for the golden ball.
5. Repeat Step 4 for about two more times, which will give you the state that represents the golden ball with high probability.

- Complexity: $O(\sqrt{N})$

Motivation

The effective security strength of key encryption algorithms

Encryption type	Encryption algorithm	Key size (bits)	Effective security level on CCs (bits)	Effective security level on QCs (bits)
Public key	RSA 1024	1024	80	0
	RSA 2048	2048	112	0
	ECC 256	256	128	0
	ECC 384	384	256	0
Private key	AES 128	128	128	64
	AES 256	256	256	128

Shor's algorithm & Grover's algorithm on QCs

UMBC

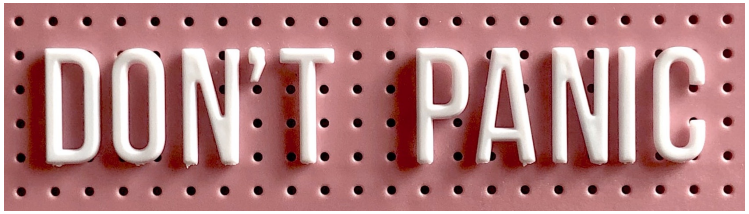
Factoring integers with sublinear resources on a superconducting quantum processor

Bao Yan,[1,2,*] Ziqi Tan,[3,*] Shijie Wei,[4,*] Haocong Jiang,[5] Weilong Wang,[1] Hong Wang,[1] Lan Luo,[1] Qianheng Duan,[1] Yiting Liu,[1] Wenhao Shi,[1] Yangyang Fei,[1] Xiangdong Meng,[1] Yu Han,[1] Zheng Shan,[1] Jiachen Chen,[3] Xuhao Zhu,[3] Chuanyu Zhang,[3] Feitong Jin,[3] Hekang Li,[3] Chao Song,[3] Zhen Wang,[3,†] Zhi Ma,[1,‡] H. Wang,[3] and Gui-Lu Long[2,4,6,7,§]

[1]*State Key Laboratory of Mathematical Engineering and Advanced Computing, Zhengzhou 450001, China*
[2]*State Key Laboratory of Low-Dimensional Quantum Physics and Department of Physics, Tsinghua University, Beijing 100084, China*
[3]*School of Physics, ZJU-Hangzhou Global Scientific and Technological Innovation Center, Interdisciplinary Center for Quantum Information, and Zhejiang Province Key Laboratory of Quantum Technology and Device, Zhejiang University, Hangzhou 310000, China*
[4]*Beijing Academy of Quantum Information Sciences, Beijing 100193, China*
[5]*Institute of Information Technology, Information Engineering University, Zhengzhou 450001, China*
[6]*Beijing National Research Center for Information Science and Technology and School of Information Tsinghua University, Beijing 100084, China*
[7]*Frontier Science Center for Quantum Information, Beijing 100084, China*

Shor's algorithm has seriously challenged information security based on public key cryptosystems. However, to break the widely used RSA-2048 scheme, one needs millions of physical qubits, which is far beyond current technical capabilities. Here, we report a universal quantum algorithm for integer factorization by combining the classical lattice reduction with a quantum approximate optimization algorithm (QAOA). The number of qubits required is $O(\log N/\log\log N)$, which is sublinear in the bit length of the integer N, making it the most qubit-saving factorization algorithm to date. We demonstrate the algorithm experimentally by factoring integers up to 48 bits with 10 superconducting qubits, the largest integer factored on a quantum device. We estimate that a quantum circuit with 372 physical qubits and a depth of thousands is necessary to challenge RSA-2048 using our algorithm. Our study shows great promise in expediting the application of current noisy quantum computers, and paves the way to factor large integers of realistic cryptographic significance.

Now or future?



- If it was true, are you ready?
- Take action now: replace public-key encryption with quantum-safe ones

Making your software quantum safe

May 4, 2022

National Security Memo (NSM-10) on Mitigating Risks to Quantum Attacks

Sep 7, 2022

NSA: Commercial National Security Algorithm Suite 2.0 (CNSA 2.0)

Lei Zhang

Nov 18, 2022

OMB: Migrating to Post-Quantum Cryptography (PQC)

Dec 21, 2022

Law H.R.7535 Quantum Computing Cybersecurity Preparedness Act

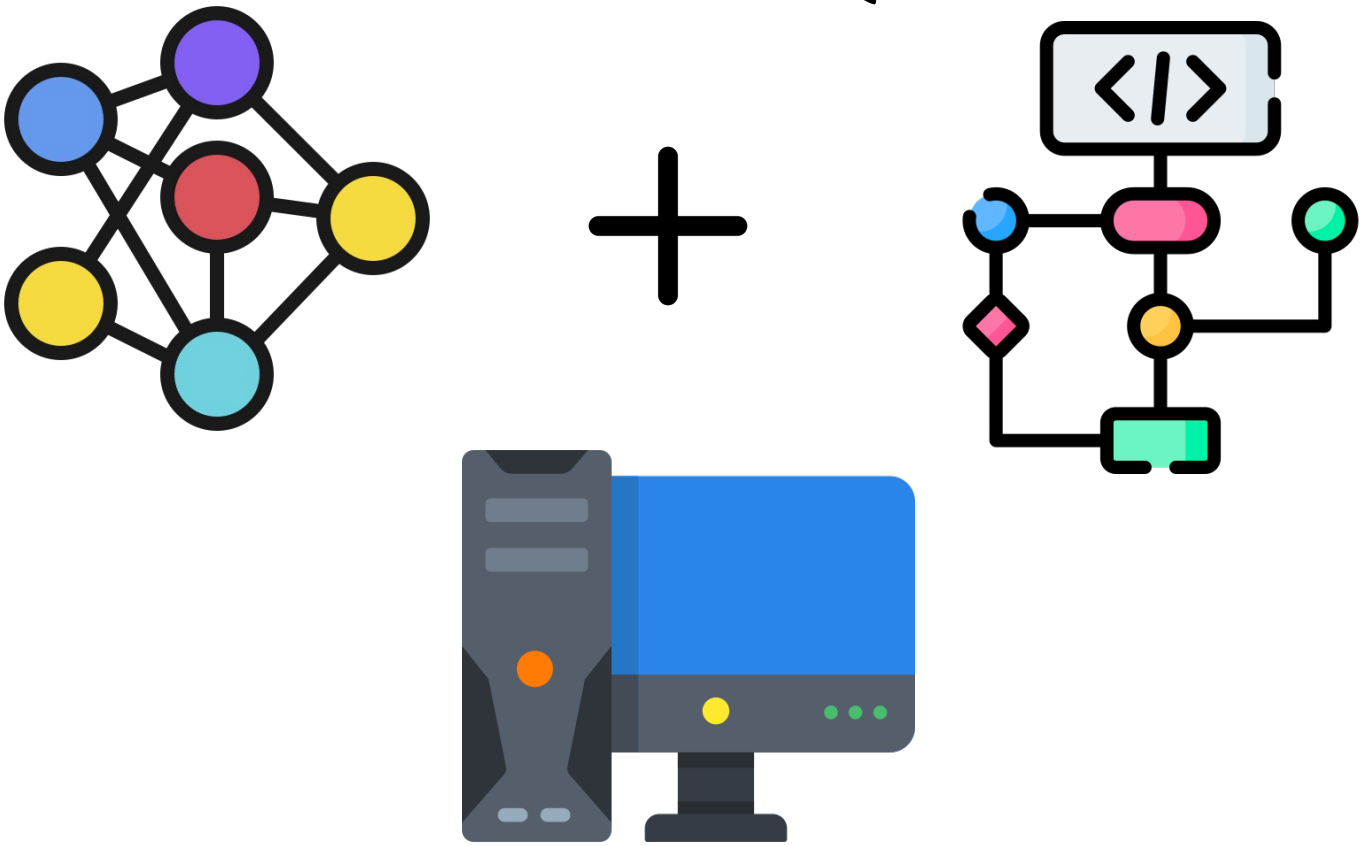
2024--2033

NIST PQC Standard; Migration to PQC

Icons credits: Xmind and Flaticon

UMBC

What is PQC?



How to migrate to PQC?

1. Find Public-Key Encryption (PKE)

2. Replace PKE with PQC

UMBC

PQC Migration Research

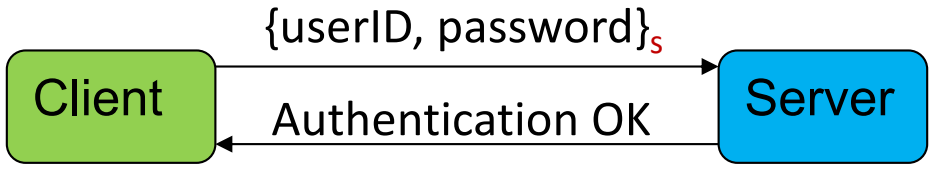
Case study on IBM Db2

Motivations

- Db2 complies with various security standards to protect sensitive data
- It is a mature product (released in 1987)
- It is under active development, with new features being added regularly
- Db2 codebase consists of tens of millions of lines of C/C++ code

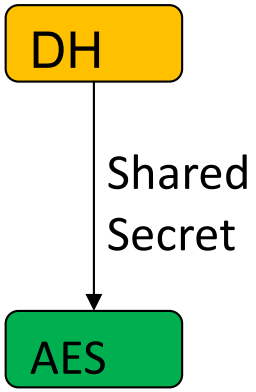
User authentication in Db2

- Db2 consists of a Db2 server and data server clients
- The first step of accessing a Db2 database is authentication, which is used to verify a user's identity
- Db2 requires two items to authenticate a user,
  - i.e., a user ID and a password



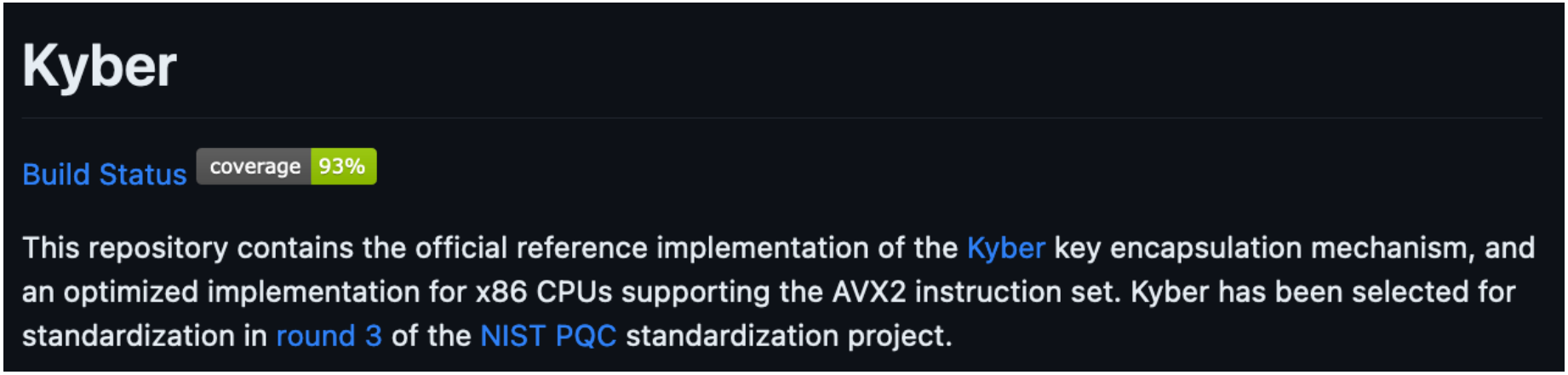
Transport Layer Security (TLS)

- To keep in-transit data safe, Db2 builds an encrypted communication using TLS protocol
- TLS uses public- and private-key encryptions
- Db2 adopts public-key encryption (Diffie-Hellman) to exchange a shared secret between the user and Db2
- The key is then used to encrypt subsequent communications with private-key encryption (AES)



UMBC

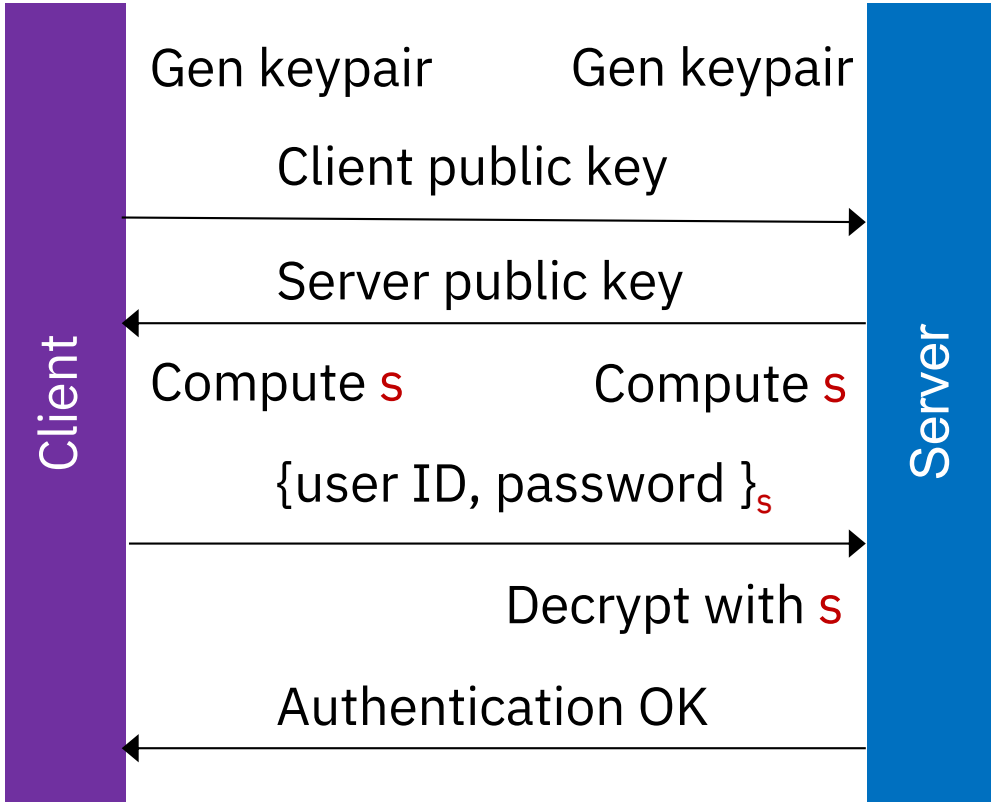
PQC: Kyber



- https://github.com/pq-crystals/kyber

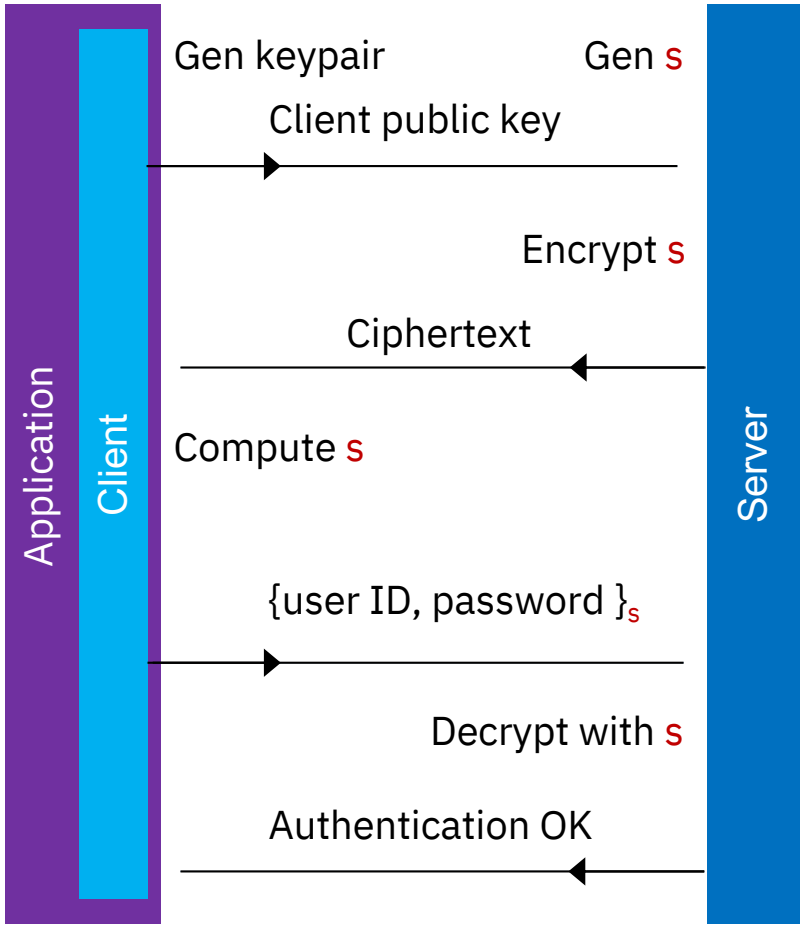
Target: Diffie-Hellman (DH) in user authentication

1. Client and server "agree" on a shared secret (s) using DH
2. Th s becomes the key of AES
3. Client encrypts user ID and password using s
4. Server decrypts user ID and password using the same s



Solution: Kyber

- Kyber is based on learning-with-errors (LWE) problem over lattice
- **A×s+e=t, NP-hard**
- Security equivalent to AES-128, AES-192, and AES-256

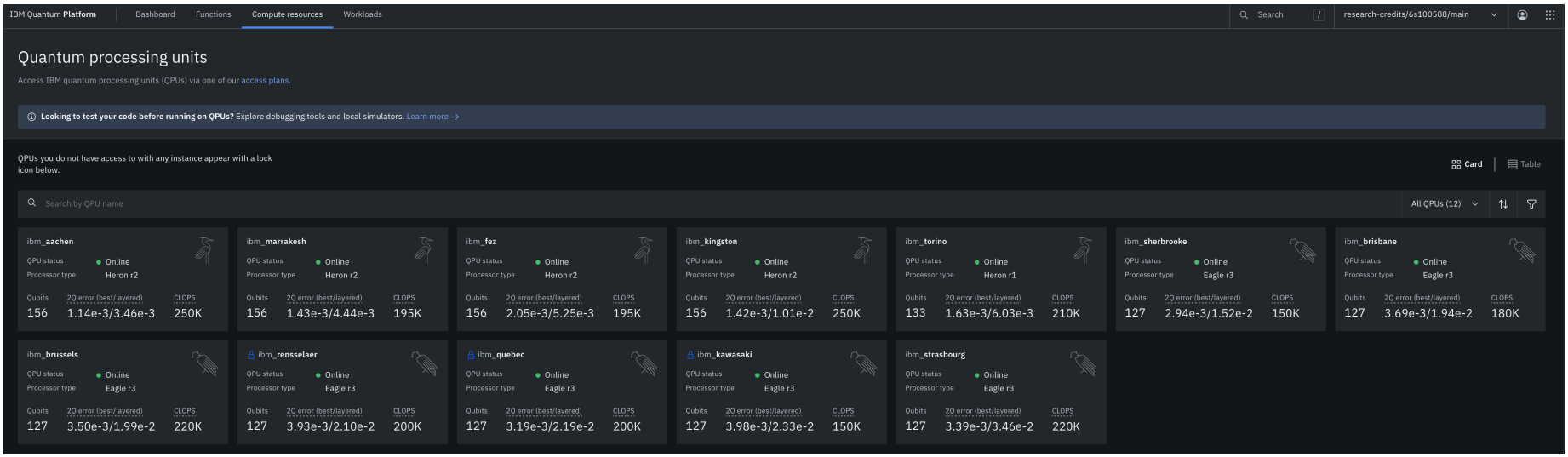


Summary

- Problem: Quantum threats
  - Public-key, replace
  - Private-key, key size
- Solution: PQC
  - Classical algorithms on classical computers to replace existing public-key encryption

UMBC

IBM quantum systems



Kahoot!

- No need to sign up
- Any mobile devices with Internet
  - Phone, laptop, etc
- Just type the web link in your browser: **www.kahoot.it**
- Join with PIN on the screen

End