Quantum Computing and U.S. Cybersecurity: A Case Study of the Breaking of RSA and Plan for Cryptographic Algorithm Transition

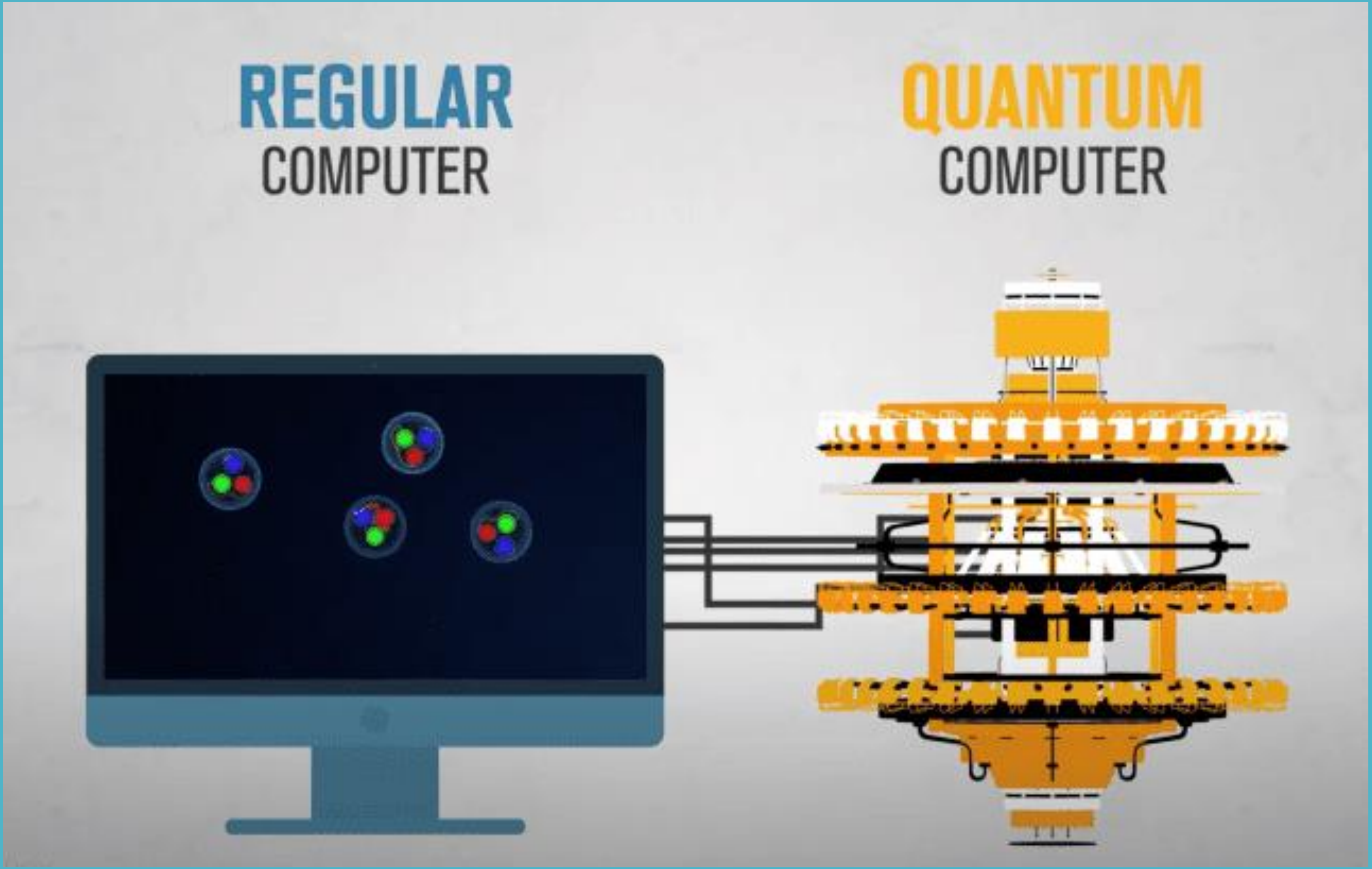
Helena Holland

Undergraduate Thesis Defense

University of Nebraska-Lincoln

Department of Mathematics

February 29, 2024


REGULAR
COMPUTER
QUANTUM
COMPUTER

Overview

Research Question: How might quantum computing technology impact American cybersecurity?

- Background
- Research Methodology
- Case Study: RSA, Shor's algorithm, American Intelligence Community's response & plan for algorithm transition
- Discussion of Results & Conclusion
- Q&A

Background

Quantum Computing

Public-key Cryptography

The Quantum Threat to Cybersecurity

How Does a Quantum Computer Work?

- **Computation:** input information → manipulate information → output result
- **Quantum computation:** a paradigm shift, but a purely theoretical device

Classical Bits vs **Quantum Bits (Qubits)**

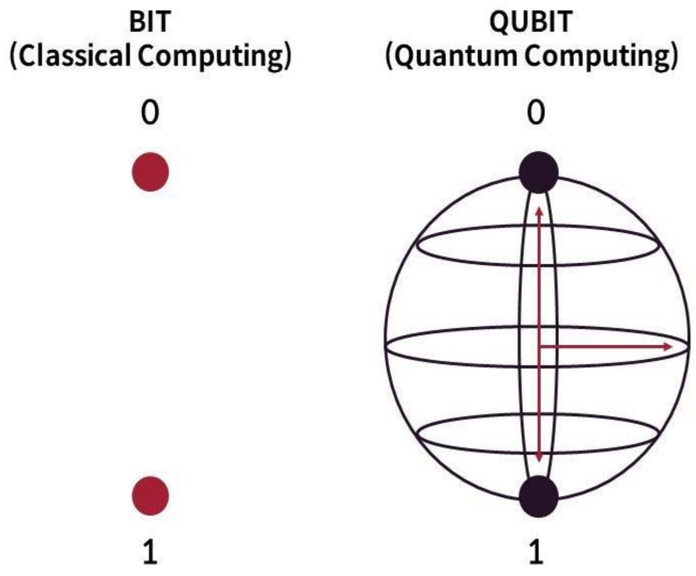
Key quantum mechanical concepts:

- superposition

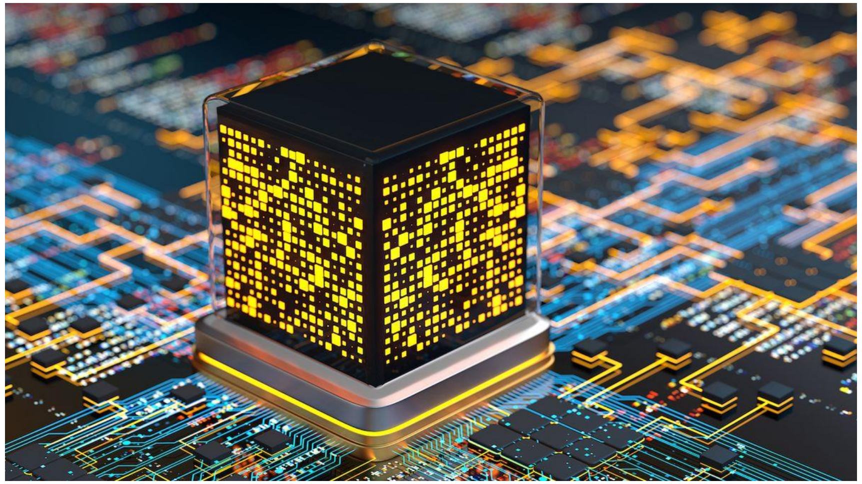
  encoding 2^n vs n states in an n-qubit system

- measurement
- entanglement

N = 3
100
110
111
101
001
011
010
000



Quantum Possibilities



- Precise sensors for biotech and defense
- Improved geospatial technologies
- Better scientific modeling, AI, machine learning, and optimization
- **Quantum Speed-Up**

  Jeopardizes modern cryptography that depends on hard problems

  Polynomial Time (Practical/Solvable): time complexity O(n^k) for some constant k

  Exponential Time (Impractical): time complexity O(2^n) for input of size n

Modern Cryptography - Encryption

Encryption: the form of cryptography that secures confidential information

The Encryption Process:

Encryption key   Decryption key

Original message (plaintext) → incomprehensible state (ciphertext) → plaintext

Key: a variable that configures the algorithm at any one time and produces a corresponding ciphertext or "unlocks" the encrypted message

Finding the key = solving a computationally difficult math problem

Public-key Cryptography

- **Symmetric** vs **Asymmetric (public-key)** encryption
- Public key (encryption), private key (decryption)

Original message (plaintext) $\xrightarrow{\text{Public key}}$ incomprehensible state (ciphertext) $\xrightarrow{\text{Private key}}$ plaintext

- Public-key algorithms/cryptosystems in use today:

    RSA

    Diffie-Hellman

    Elliptic curve cryptography

All of these public-key algorithms are dependent on the factoring or discrete logarithm problems for security.

The Quantum Threat to Cybersecurity

A quantum computer can solve both the factoring and discrete logarithm problems in polynomial time using **Shor's algorithm** (1994), rendering all forms of public-key cryptography vulnerable as soon as a quantum computer is built.

Research Methodology

Research Question: How might quantum computing technology impact American cybersecurity?

Case Study Method:

- RSA & Shor’s Algorithm
- The plan for migration to post-quantum cryptography
  - Quantum-Resistant Algorithm Standardization Process
  - National Security Memorandum 10 (NSM – 10)
  - SWOT Analysis of Algorithm Transition Plan

Case Study

RSA and Shor’s Algorithm

The American Intelligence Community’s Response

RSA

- Developed in 1977 by cryptologists Rivest, Shamir, and Adleman (RSA)
- Secures online financial transactions, web browsers, email services, VPNs
- RSA relies on the **factoring problem**

Find odd prime numbers p and q such that a large number n = pq

Cracking RSA = factoring a large number n into two primes (possible for a quantum computer)

RSA

Definition 3.1: ($\mathbb{Z}$ denotes the set of all integers.) The numbers a, b $\in \mathbb{Z}$ are congruent modulo N, written $a \equiv b \pmod{N}$, if $N \mid a - b$.

Algorithm 3.1: RSA Key Establishment and Encryption

Input: plaintext b

1. Pick at random two primes, p and q.
2. Compute $n = p*q$.
3. Choose a value e such that $1 < e < (p - 1)(q - 1)$ and $\gcd(e, (p - 1)(q - 1)) = 1$.
4. Publish the public key (e,n).
5. Compute $d \equiv e^{-1} \pmod{(p - 1)(q - 1)}$, the private key.
6. To encrypt a message b, a user computes $y = b^e \pmod{n}$, the encrypted message.

Output: ciphertext y

Algorithm 3.2: RSA Decryption

Input: private key d; ciphertext $y = b^e \pmod{n}$

1. Compute $b = y^d \pmod{n}$ to recover the original message.

Output: plaintext message b

RSA Algorithm Example: n = 3*11

Suppose you are the party designated to hold the private key:

Encryption: original message **b = 2**.

- Choose two odd primes, **p = 3** and **q = 11**. Then **n = 33** and **(p – 1)(q – 1) = 20**.
- Choose a value **e = 7** such that 1 < e < 20 and gcd(e, 20) = 1.
- Compute a value **d = 3** such that d ≡ e^{-1} (mod 20). (3*7 ≡ 1 (mod 20))
- The public key is (e,n) = (7,33).
- To encrypt b = 2, a party calculates **y** = b^e (mod n) = 2^7 (mod 33) = **29**.

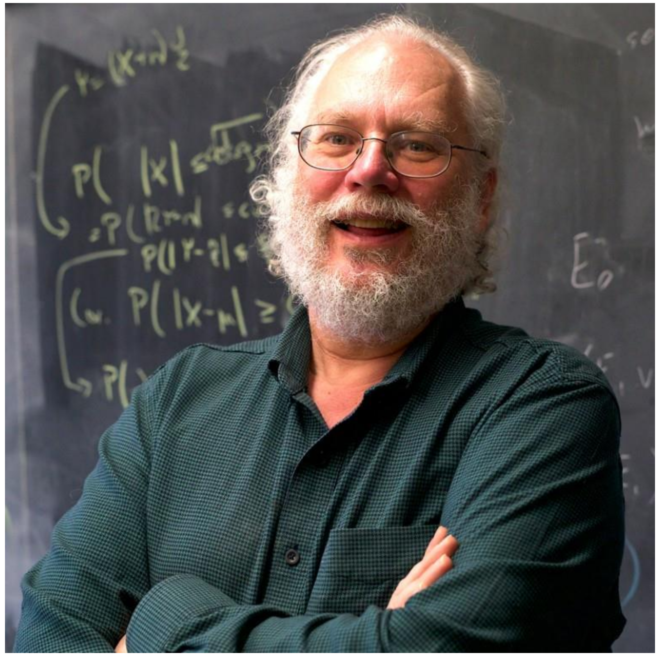
Decryption: We want to decrypt the ciphertext **y = 29** to recover the original message **b = 2**.

- Using the private key **d = 3**, calculate b = b^{ed} (mod n) = y^d (mod n) = 29^3 (mod 33) = 2.
- The original message, b = 2, has been uncovered.

Standard RSA key sizes are 1024-bit, 2048-bit, or 4096-bit, making n = pq computationally difficult to factor.

Shor’s Algorithm

A crowning achievement of the last century, developed by AT&T researcher Peter Shor in his 1994 paper “Polynomial –Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer”



Shor's Algorithm for Prime Factorization

A post-processing shortcut after finding the order r

Algorithm 3.3: Shor's Algorithm for Prime Factorization[51]

Input: n from the RSA public key

1. Pick $a \in \mathbb{Z}$ at random such that $\gcd(a,n) = 1$ and $1 < a < n$.
2. Find $r =$ order of $[a]_n$ (using the quantum part of Shor's algorithm).
3. **Case 1:** r is odd.
   (i) The algorithm FAILS. Return to step 1 and choose another a.

   Case 2: r is even.

   1. Compute $\gcd(n, a^{r/2} - 1)$ using the Euclidean algorithm.
      (i) **Case 1:** $n > g = \gcd(n, a^{r/2} - 1) > 1$.
         1. The algorithm SUCCEEDS and terminates. A non-trivial factor of n, g, has been found.

      (ii) **Case 2:** $\gcd(n, a^{r/2} - 1) = 1$.
         1. The algorithm FAILS. Return to step 1 and choose another a.

Output: g, a nontrivial factor of n

Example: RSA and Shor's Algorithm

RSA ENCRYPTION

n = 33, p = 3, q = 11, d = 3

Public key (e,n) = (7,33)

Original message: b = 2

Ciphertext: y = 2^7 (mod 33) = 29.

RSA DECRYPTION KEY (q = 11)

→ d ≡ e^{-1} (mod(p - 1)(q - 1))

7d ≡ 1 (mod 20)

d = 3

SHOR'S ALGORITHM

- Pick a = 2 [gcd(2,33) = 1]:

  r = $|[2]_{33}|$ = 10

  r is even

  gcd(33, $2^{10/2}$ – 1) = gcd(33, 31) = 1. FAIL.

- Pick a = 4:

  r = 5. r is odd. FAIL.

- Pick a = 5:

  r = 10

  r is even

  gcd(33, $5^{10/2}$ – 1) = gcd(33, 3124) = 11. **SUCCESS.**

The U.S. IC's Response

The **NSA** officially called for a transition to quantum-resistant cryptography in 2015.

- Quantum-resistant algorithm development and standardization
  - The National Institute of Standards and Technology (**NIST**)

- Executing a successful transition project across national systems
  - NSM-10

Goal = transition national security systems and critical infrastructures by **2035**

SWOT Analysis of Algorithm Transition Plan

STRENGTHS	WEAKNESSES	OPPORTUNITIES	THREATS
Crypto-agility emphasis	Uncertain timing of standards and execution	Facilitate future adaptations (crypto-agility)	Adversary plans to steal vulnerable, encrypted data before re-processing
QIS R&D	Diverse infrastructures require individualized solutions	Increased QIS awareness	Large-scale disruption
Ongoing algorithm standardization	Minimal records of cryptography use/function	Organization of cryptography use/function and security standards	Negatively affecting system security or business functions
Collaboration across domains	Vulnerable to stealing encrypted data before re-processing	Stronger relationships between government, industry, standards bodies	U.S. solution export risks

Discussion

Research Question: How might quantum computing technology impact American cybersecurity?

Key Threats to Cybersecurity

Key Opportunities for Cybersecurity

Quantum Threats to Cybersecurity

DIRECT	INDIRECT	CONSEQUENCE
The destruction of RSA and public-key cryptography	Post-quantum migration entails large-scale disruption that may weaken security during the transition process (likely to continue)	Failure to transition would undermine military and civilian communications, critical control systems, online financial transactions
	Incentivizes the stealing of U.S. solutions and vulnerable, encrypted information before re-processing	Motivates system attacks, adversary exploitation of information, decreased competition within industry
	Losing the quantum race	

Quantum Opportunities for Cybersecurity

DIRECT	INDIRECT	CONSEQUENCE
Extremely secure encryption and better system performance through QIS	The process of transitioning towards quantum-resistant cryptography forces the organization of cyberspace	More efficient cryptographic transitions in the future
	Increased crypto-agility, automation, and system security going forward	Organization, documentation, and automation strengthen cybersecurity
	NSM-10 mandates may lead to QIS advancement through government, academia, and industry partnerships	Advancement in QIS and cryptography

Conclusion

The impact of quantum computing depends largely on the success of the transition project but will make obsolete all forms of currently-employed public-key cryptography and introduce large-scale change and disruption across American digital systems.

- Strengths and weaknesses of the case study method
- Topics for further research
- Research contribution