

The University of Akron

IdeaExchange@UAkron

Williams Honors College, Honors Research
Projects

The Dr. Gary B. and Pamela S. Williams Honors
College

Fall 2023

The Future Between Quantum Computing and Cybersecurity

Daniel Dorazio
drd75@uakron.edu

Follow this and additional works at: https://ideaexchange.uakron.edu/honors_research_projects



Part of the [Information Security Commons](#), [Quantum Physics Commons](#), and the [Software Engineering Commons](#)

Please take a moment to share how this work helps you [through this survey](#). Your feedback will be important as we plan further development of our repository.

Recommended Citation

Dorazio, Daniel, "The Future Between Quantum Computing and Cybersecurity" (2023). *Williams Honors College, Honors Research Projects*. 1778.

https://ideaexchange.uakron.edu/honors_research_projects/1778

This Dissertation/Thesis is brought to you for free and open access by The Dr. Gary B. and Pamela S. Williams Honors College at IdeaExchange@UAkron, the institutional repository of The University of Akron in Akron, Ohio, USA. It has been accepted for inclusion in Williams Honors College, Honors Research Projects by an authorized administrator of IdeaExchange@UAkron. For more information, please contact mjon@uakron.edu, uapress@uakron.edu.

The Future Between Quantum Computing and Cybersecurity

Daniel Dorazio

The University of Akron

Williams Honors College

December 1, 2023

Abstract

Quantum computing, a novel branch of technology based on quantum theory, processes information in ways beyond the capabilities of classical computers. Traditional computers use binary digits [bits], but quantum computers use quantum binary digits [qubits] that can exist in multiple states simultaneously. Since developing the first two-qubit quantum computer in 1998, the quantum computing field has experienced rapid growth.

Cryptographic algorithms such as RSA and ECC, essential for internet security, rely on the difficulty of complex math problems that classical computers can't solve. However, the advancement of quantum technology threatens these encryption systems. Algorithms, such as Shor's, leverage the power of quantum machines to factor large numbers, a task challenging for classical computers.

Acknowledging this threat, it is important to develop and implement quantum-resistant cryptography to safeguard communication, financial systems, and national security. This study covers the past, present, and future of quantum computing and cybersecurity and their increasingly connected roles. It provides a detailed history of both fields, explores the challenges posed by quantum computing to traditional cryptographic methods, and discusses the development of new, robust cryptographic solutions to ensure security in a future where quantum computing is prevalent.

Table of Contents

1.0 Introduction.....	6
1.1 Background.....	6
1.2 Purpose of Research.....	7
2.0 Cybersecurity Essentials	8
2.1 History of Cybersecurity.....	8
2.1.1 <i>Internet Origins</i>	8
2.1.2 <i>The World Wide Web</i>	11
2.1.4 <i>Cyber Warfare</i>	13
2.1.5 <i>Revelations of Mass Surveillance</i>	14
2.1.6 <i>Modern Cybersecurity</i>	15
2.2 Cybersecurity Fundamentals.....	15
2.2.1 <i>Historical Context to Cryptography</i>	15
2.2.2 <i>Classical Cryptography</i>	16
2.2.3 <i>Evolution Towards Modern Cryptography</i>	16
2.2.4 <i>Introduction of Public Key Cryptography</i>	18
2.2.5 <i>RSA Cryptosystem</i>	19
2.2.6 <i>DES and AES</i>	20
2.2.7 <i>Elliptic Curve Cryptography</i>	21
2.2.8 <i>Non-cryptographic Cybersecurity</i>	21
3.0 Quantum Computing Essentials.....	23
3.1 History of Quantum Computing	23
3.1.1 <i>The Quantum Turing Machine</i>	23
3.1.2 <i>The First Quantum Computer</i>	24
3.1.3 <i>Advancements and Milestones</i>	24
3.1.4 <i>The Race for Quantum Supremacy</i>	25
3.1.5 <i>Modern Quantum Computing</i>	25
3.1.6 <i>Quantum Computing in Other Industries</i>	26
3.1.7 <i>Challenges and Outlook</i>	27
3.2 Quantum Computing Fundamentals	28
3.2.1 <i>Qubits</i>	28
3.2.2 <i>Entanglement</i>	29

3.2.3 Quantum Interference	29
3.2.4 Coherence and Decoherence	29
3.2.5 Quantum Error Correction	30
4.0 Quantum Threats.....	32
4.1 Shor's Algorithm	32
4.1.1 Overview	32
4.1.2 Threats to RSA and ECC.....	32
4.1.3 Time Complexity and Efficiency	33
4.2 Grover's Algorithm.....	33
4.2.1 Overview	33
4.2.2 Data Security Implications.....	34
4.3.1 Shor's Algorithm Requirements	34
4.3.2 Grover's Algorithm Requirements	34
4.3.3 Quantum Performance Trends.....	35
5.0 Quantum Resistant Cybersecurity Practices.....	36
5.1 Replacements for Private Key Cryptography	36
5.1.1 NIST Post-Quantum Cryptography Selection Process	36
5.1.2 Structured Lattices and Hash Functions.....	36
5.2 Strengthening AES	37
5.3.1 Quantum Key Distribution.....	37
5.3.2 Quantum Random Number Generation	38
6.0 Testing Hadamard Gates and Quantum Fourier Transforms via IBM Qiskit.....	39
6.1 Background.....	39
6.1.1 IBM Cloud.....	39
6.1.2 Fourier Transform and Quantum Fourier Transform [QFT]	39
6.1.3 Hadamard Gate.....	40
6.2 Creating a QTF Program with Python and Qiskit.....	40
6.3 Executing the QTF Program	42
6.4 Results.....	43
6.4.1.....	43
6.4.2 No Gates Applied	43
6.4.3 Hadamard Gate Applied	44
6.4.4 Quantum Fourier Transform Applied.....	45

6.4.5 Hadamard Gate and QFT Applied.....	46
6.5 Conclusion of Hadamard and QFT Testing.....	47
7.0 Summary.....	49

1.0 Introduction

1.1 Background

Quantum computing is the novel branch of technology that leverages the principles of quantum theory, the study of matter and energy at the most fundamental level [1], to process information in ways that classical computers cannot. While classical computers operate using bits representing binary states (0, 1), quantum computers utilize special quantum bits, or qubits, which can exist in multiple states simultaneously, offering extraordinary computational power when solving specific problems.

Since the inception of the original two-qubit quantum computer in 1998 [2], this field has experienced rapid growth, marked by significant advancements. These advancements hold the promise of solving problems currently considered intractable for even the most powerful classical computers. Many of these challenges are found in modern cryptography, which relies heavily on the presumed difficulty of nondeterministic polynomial [NP] problems—complex mathematical problems that are easily verified but nearly impossible to solve in a reasonable amount of time. Cryptographic algorithms such as RSA and ECC, used across the internet to encrypt confidential data, were developed based on the premise that these mathematical problems are impossible to solve.

However, as quantum technology continues to evolve, it is becoming increasingly apparent that quantum computers will soon be able to break these encryption systems using a quantum algorithm such as Shor's that can factor large integers in polynomial time. [3] Classical machines can factor large integers in sub-exponential time at best. [4] Recognizing this looming threat, developing and deploying quantum-resistant cryptography to protect the infrastructure

that supports our communication networks, financial systems, and national security is imperative.

1.2 Purpose of Research

This paper dives into the evolving domains of quantum computing and cybersecurity and explores their likely convergence as the technologies advance. Specifically, the paper aims to:

- Provide a comprehensive quantum computing and cybersecurity history, tracing their origins and significant milestones.
- Offer a basic understanding of the fundamental technology behind both fields, including the principles of quantum mechanics that underpin quantum computing and the cryptographic protocols that serve as the foundation for modern cybersecurity.
- Analyze the current state of quantum computers concerning their capability to disrupt existing cryptographic methods, particularly when referring to widely used asymmetric algorithms such as RSA or ECC.
- Explore the practical implications of quantum computing on cybersecurity strategies, including potential impacts on data privacy, secure communications, and the integrity of global information systems.
- Discuss the ongoing development of quantum-resistant algorithms and modifications to existing algorithms that will withstand the computational capabilities of quantum computers.

This study strives to bridge the gap between quantum computing and cybersecurity and intends to forecast future needs and responses in a future of quantum superiority.

2.0 Cybersecurity Essentials

2.1 History of Cybersecurity

As they are known today, computers began to emerge in the 1950s as a new method of computation and data processing. While these machines were incredibly powerful compared to anything previously available, pre-microprocessor machines were typically used independently of each other for performing calculations that once had been completed by hand or by using expensive and complex adding machines. It wasn't until the 1960s that the idea of the computer network began to materialize. [5]



Figure 1: A 1962 brochure for the IBM 1440: a "New low-cost Data Processing System" [6]

2.1.1 Internet Origins

While the World Wide Web didn't exist until the 1990s, the infrastructure that led to it began development 30 years before. In 1960, Paul Baran of the RAND Corporation developed the concept of packet switching while researching networks the US Military could use for secure and robust voice communication in the event of a nuclear war. [5] Packet switching is a fault-

tolerant method of dividing data into small pieces and transmitting them over a wire. These trim pieces are called packets; each has a header and payload. The header contains routing information about where the packet needs to be delivered, and the payload is the packet's data. [7] Before packet switching, networks were typically hardwired with no flexibility and had little to no resistance to hardware failure. Despite significant developments and advancements in network technology, packet tracing remains the foundation of modern Internet protocol.

One of the earliest examples of a working packet-switching network was ARPANET. ARPANET, or “Advanced Research Projects Agency Network,” was developed by ARPA [Advanced Research Projects Agency], a division of the Department of Defense. ARPANET aimed to connect multiple computers without needing individual terminals for each location to be connected. Using packet switching, ARPANET would allow connected terminals to send and receive information from other separate terminals on the network because of the destination in the header of data packets.

In 1971, when ARPANET was still in its infancy, an employee of the BBN research company named Bob Thomas developed a piece of software called Creeper. This software program would display the message “I’M THE CREEPER, CATCH ME IF YOU CAN” on the terminal, then transmit itself to other computers across the ARPANET running the TENEX operating system. [8] While this program was utterly harmless, affecting no data, it was the first example of a program that could replicate itself and spread across a network automatically. Programmer Ray Tomlison later developed Reaper, a self-replicating antivirus that would delete the Creeper program from infected machines across ARPANET. The Creeper and Reaper programs are colloquially known as the first virus and antivirus programs. While these programs

were created out of curiosity and experimentation, they foreshadowed what would come in the commercialized future of a giant, connected group of computers.

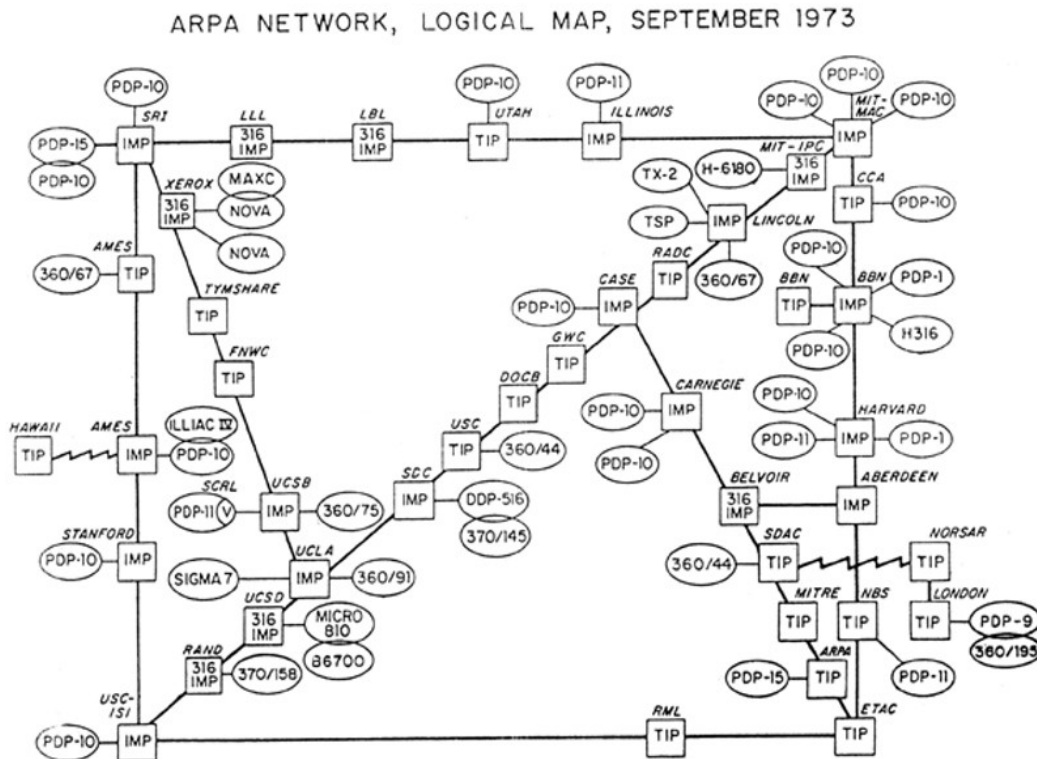


Figure 2: Logical map of ARPANET in September of 1973 [9]

ARPANET brought innovations and features that had never been possible, such as remote logins to another terminal, file transfers, and even the first email. [10] In 1983, ARPANET split into two networks, a military network called MILNET and a civilian network, which kept the original name. Among other reasons, MILNET was created to continue operating in the event of a disruption to a civilian network. The remaining civilian portion of ARPANET remained as a research network that continued to develop and grow into the modern Internet. The same year ARPANET was split up, it adopted the TCP/IP; this robust communications protocol was built to utilize the existing packet-switching data network with scalability and fault tolerance in mind. TCP/IP continues to be the foundation of all Internet communication today. [11]

2.1.2 The World Wide Web

In the late 1980s, the Internet was predominantly utilized by government bodies, academic institutions, and a growing number of computer hobbyists. Despite its potential, the Internet had yet to capture widespread public interest, primarily due to the high barriers to entry. Networking was rare in personal computing, and technical expertise was required to access the Internet, even with the appropriate equipment. [12]

This landscape began to shift with the innovations of Tim Berners-Lee, a British physicist who envisioned a linked information system that would come to be known as the World Wide Web. [13] Berners-Lee's creation laid the groundwork for the Internet's explosive growth, with features that were key to its widespread adoption:

- **Universality:** The Web was accessible across many hardware platforms and operating systems.
- **Hypertext:** A system of interlinked documents made information navigation significantly more intuitive than previous models.
- **Decentralization:** No single entity controlled the World Wide Web, allowing for development from anyone.
- **Simplicity:** HTML and HTTP were designed for ease of use, enabling anyone to create and publish content on the Web.
- **Multimedia Support:** While initially text-based, the Web's infrastructure could support multimedia content like images, audio, and video, enabling content-rich sites.
- **Royalty-Free Access:** Berners-Lee and CERN's decision to make Web technology freely available removed financial barriers to its adoption.
- **Open Source:** The protocols that drove the Web were made openly available, encouraging developers and the community to build upon them, leading to a proliferation of different web browsers, websites, and applications.

These foundational elements facilitated the rapid expansion of the Web, transforming it from a hobbyist landscape to an essential tool for global communication and information sharing.

2.1.3 Advent of Cybercrime

While the Internet transitioned from a hobbyist landscape to a commercial success, another transition happened concurrently. While ARPANET faced incidences of hacking, these were primarily driven by curiosity rather than malicious intent. However, as the Internet became a place for real business, commerce, and communication, bad actors found a new frontier for crime. Incidences of hacking once characterized as innocent curiosity were eclipsed by motivations of profit, vandalism, and espionage.

One of the most infamous cybercrime cases on the early Internet is the ILOVEYOU worm of May 2000. This worm was a Visual Basic script disguised as a love letter attached to an email that, when executed, would steal passwords, overwrite files, and send itself to all contacts in the victim's Windows address book. [14] This simple script caused devastating damage estimated at billions of dollars [15] as it spread to millions of computers across the globe. This event showcased the vulnerability of the systems being used and how adoption had far surpassed the security measures needed to keep incidents like this at bay.

Another devastating virus in the early 2000s was a worm called SQL Slammer. This worm exploited a vulnerability in Microsoft SQL Server 2000, allowing for arbitrary code execution through a buffer overflow attack. It is estimated that the attack infected 250,000 to 300,000 hosts, and at its peak, the number of infected hosts doubled every 8.5 seconds. The attack shook the entire Internet as it brought significant service outages and high packet loss. This incident demonstrated how necessary *timely* software updates are, as Microsoft had already pushed a patch for this specific vulnerability months before the attack. [16] [17]

While the prior two examples demonstrate attacks that imposed severe destruction on massive scales, more targeted attacks on vulnerable groups emerged. Phishing scams, where

attackers disguise themselves as reputable organizations to extract sensitive information, became rampant, resulting in soaring numbers of identity theft cases.

Attacks continued to become more complex and far outpaced attempts at mitigation. Cybercriminals developed advanced malicious software such as spyware, keyloggers, and ransomware to exploit vulnerable users. Because of the rampant spread of these types of criminal activity, the FBI founded the Internet Crime Complaint Center [18], and the FTC acted to protect consumers. This government response marked a crucial step towards formalizing efforts to combat cybercrime, signaling an increased awareness and prioritization at the federal level to address the growing threat of online criminal activity.

2.1.4 Cyber Warfare

While cybercrime was initially a domain of individual hackers or groups attacking organizations or exploiting individuals for destruction or financial gain, this paved the way for more organized and state-sponsored attacks, known as cyber warfare. The first extremely sophisticated incident of cyber warfare occurred in 2010 when malware was discovered to have infiltrated the networks of multiple Nuclear Enrichment facilities in Iran. [18]

While there had been previous instances of politically motivated attacks, such as the 2002 attack of a SCADA system at a marine terminal in Venezuela [19], the attack that became known as Stuxnet was by far the most sophisticated and destructive at the time. The Stuxnet worm only targeted Siemens SCADA [supervisory control and data acquisition] systems to modify the behavior of PLCs [programmable logic controllers] devices used in industrial environments to control manufacturing equipment. In this case, Iran was using secretly procured Siemens equipment as part of an operation to enrich uranium, which is a crucial step to creating a nuclear warhead.

The malware accessed the industrial control systems by exploiting a vulnerability in Microsoft Windows and Windows Server. Since industrial control systems are often isolated from the Internet as a security measure to prevent external attacks, the worm was also programmed to replicate through removable media such as flash drives to jump from machine to machine even without a network connection. Another characteristic unique to Stuxnet at the time was that it remained silent and undetectable on any machine not to raise any alarms—it only had adverse effects on the intended target.

The Iran incident started the current cyber warfare era, characterized by nation-states' increasing use of digital tactics to achieve their objectives. This evolution has forced governments to rethink security strategy from the ground up and emphasized the importance of national security within cyberspace. As technology continues to evolve, so does the potential for cyber warfare incidents, making it essential for governments, private industry, and individuals to remain vigilant in protecting their digital infrastructure and assets.

2.1.5 Revelations of Mass Surveillance

In the Internet era, mass surveillance has emerged as a topic of utmost importance. Mass surveillance, defined as monitoring a large portion of a population, has changed significantly in the digital age. Mass surveillance isn't a new concept. It was common in the 20th century to monitor phone conversations outbound from the US during the World Wars and the Cold War. Surveillance is often justified under the banner of national security and protection. Even so, it is highly controversial due to questions about privacy rights and the balance of power between the state and the individual.

There was a significant disclosure about mass surveillance in the US in 2013 when an IT consultant working for the NSA by the name of Edward Snowden leaked an abundance of

classified documents that detailed the US government's telephone surveillance of citizens and foreign leaders along with a vast collection of Internet records. This disclosure was sensational as the average US citizen had no idea of the scale of the intelligence collected on regular US citizens. [20]

Much has changed in the US since the revelations of 2013 in the way of government reform and pullback on mass surveillance. End-to-end encrypted [E2EE] services, specifically related to communication, have increased as a popular way to ensure conversations remain private across the Internet. The proliferation of E2EE services demonstrates that the average user prioritizes their privacy online more than ever and shows a growing awareness of how to protect it.

2.1.6 Modern Cybersecurity

In 2023, the cyber landscape continues to evolve constantly. The Internet is the crux of our economy, communication, commerce, industry, medicine, and entertainment. It is the most essential infrastructure ever created. Many new areas of the internet are certain to have cybersecurity implications, such as generative AI and cryptocurrency, the importance of the internet to the political landscape, and the spread of information through social media. Almost every aspect of our world now relies on the internet in some way, and close attention must be paid to every threat, such as quantum computing, which can potentially compromise this critical backbone of our society.

2.2 Cybersecurity Fundamentals

2.2.1 Historical Context to Cryptography

Cryptography is one of the most essential aspects of modern cybersecurity, but it has existed for thousands of years in what is now called 'classical cryptography.' Classical

cryptography was often used during war efforts, even in ancient times, to obscure information such that if it were intercepted, it couldn't be deciphered.

2.2.2 Classical Cryptography

Cryptography is defined by Merriam-Webster simply as “secret writing.” Encryption is the process of obscuring data for confidentiality. One of the earliest and most simple cryptographic algorithms that functioned this way was the Caesar Cipher. The Caesar Cipher is named after Julius Caesar, who obscured confidential messages nearly 2000 years ago by shifting letters in the alphabet by a certain number of characters. [21] Ex. If the number of characters to be shifted were three to the left, then A would become X, B would become Y, etc.

The Caesar Cipher is an early example of private key cryptography. Private key cryptography means that the encryption and decryption methods are the same. That is to say, both the encrypting and decrypting parties use a key, k , as part of an encryption function. Anyone with the private key and knowledge of the encryption function can encrypt or decrypt messages; no one without the private key would be able to encrypt or decrypt messages.

2.2.3 Evolution Towards Modern Cryptography

The path to modern cryptography saw many advancements in private key cryptography from Caesar's time to the advent of computers. One of the most pivotal developments was the Enigma cipher machine. The Enigma, which began development towards the end of World War I, played a critical role in World War II. It was utilized by Nazi Germany as an innovative method to encrypt messages before transmission. The machine resembled a typewriter but with a unique addition: an array of QWERTY lights above the keys, mirroring their layout. It also featured three windows displaying numbers from a set of rotors. Military versions used in WWII included a plugboard at the front, adding an extra layer of security by swapping different letters.



Figure 3: An Enigma machine in use, 1943 [22]

Pressing a key on the machine completed an electrical circuit through the plugboard and rotors, lighting up a corresponding letter on the keyboard above. Operators set the rotors and plugboard to specific configurations based on a military-provided Enigma calendar, ensuring that all Enigma machines shared settings on any given day, with no repeat settings. However, someone who obtained this calendar could only decrypt messages until the settings changed. [23]

A key distinction of the Enigma machine was its departure from previous encryption methods, where one character could only be encoded as a single different character. For instance, in a plaintext message like "HOPPER," the letter 'P' would always encode to the same character, say 'X.' With the Enigma, the rolling rotors meant that the first 'P' could encode as 'X.' Still, the

next 'P' could encode as any other letter, vastly complicating decryption efforts for intercepted messages.

Polish cryptanalysts initially cracked the Enigma's encryption by exploiting the machine's inability to encode a letter as itself. Later, scientist Alan Turing and his team designed the Bombe machine, significantly accelerating the decryption of daily German communications. While the Enigma machine's vulnerability led to its downfall, it remains a remarkable feat of engineering that paved the way for future advancements in cryptography.

2.2.4 Introduction of Public Key Cryptography

Modern encryption took a significant leap forward when mathematicians Whitfield Diffie and Martin Hellman introduced the concept of asymmetric public key cryptography in 1976. Public key cryptography would use different keys for encrypting and decrypting information. This was a significant shift from the traditional symmetric key cryptography, which required both parties to share the same secret key.

The Diffie-Hellman key exchange involves two participants, each generating their public/private key pair and sharing their public key with the other over an unsecured channel. They can then use their private key with the other participant's public key to generate a shared secret, which can be used for symmetric encryption over the unsecured channel. [24]

This was groundbreaking since there wasn't previously a simple solution to establishing a secure method of communication through an insecure channel. Diffie and Hellman's work would continue to serve as the foundation for the modern RSA cryptosystem.

2.2.5 RSA Cryptosystem

In 1977, three scientists, Ron Rivest, Adi Shamir, and Leonard Adleman, publicly announced the cryptosystem they had developed, named RSA. [25] This system was built upon the concepts Diffie and Hellman had established, turning it into a fully working cryptosystem.

The RSA cryptosystem used the same principle of generating a public-private key pair and sharing the public key with another participant over the insecure channel. However, instead of using the public and private keys to generate a shared secret for symmetric encryption, the public keys could be used to directly encrypt information that could only be decrypted by the person with the corresponding private key.

For example, Alice and Bob want to share secret information over an insecure channel. First, they would generate their own public/private key pair and share the public keys over the insecure channel. If Alice wants to say something to Bob secretly over the insecure channel, she will use Bob's public key to encrypt her plaintext and send the ciphertext across the insecure channel. Bob would use his private key to decrypt the ciphertext Alice had sent.

Additionally, RSA included support for digitally signing information. If Alice wanted to receive data from Bob and ensure the information had not been modified in transit, Bob could use a hashing function to create a unique hash of the plaintext, encrypt the hash using his private key, and send that to Alice along with the encrypted plaintext. Alice would proceed to decrypt the ciphertext and the digital signature Bob sent. Alice would then use the same hashing function on the plain text and compare her hash to what Bob had sent. If the hashes match, this indicates that Alice's plaintext is identical to the plaintext that Bob sent, and no changes occurred during the transmission of the messages.

RSA is a breakthrough technology that cemented public-key cryptography as a framework for establishing secure digital communication through insecure channels. The same technology, albeit with longer keys, is still used today. However, while its enduring relevance to cybersecurity persists to this day, the advent of quantum will likely mark the end of RSA as more quantum-resistant technologies become necessary.

2.2.6 DES and AES

The same year RSA cryptography was debuted, the US agency previously known as the National Bureau of Standards (now NIST) established the Data Encryption Standard (DES) as a standardized way to encrypt and protect unclassified government traffic. [26] DES was a 64-bit symmetric cryptographic system that became ubiquitous in the early Internet age. Endorsed by the US government, it was used for encryption in online banking and early SSL versions, the HTTPS protocol's foundation.

Despite its widespread use, DES had a significant flaw—it was vulnerable to brute-force attacks, with a key length of 64 bits, 56 of which are used for encryption. DES had 2^{56} possible keys, or about 72 quadrillion. While a personal computer couldn't brute-force a 56-bit encryption key in a reasonable amount of time, the Electronic Frontier Foundation (EFF) developed a machine that, in 1998, could brute-force a key in just 56 hours. [27] This machine, named the EFF DES Cracker, cost \$250,000 — a hefty sum, yet easily within reach for many large organizations. This vulnerability signaled the need for a more secure form of encryption. As a result, a modified version called Triple DES, or 3DES, was developed. 3DES enhanced security using the DES algorithm in three stages with three different keys.

However, 3DES was only a temporary solution. The need for a new encryption standard led to the development of the Advanced Encryption Standard, or AES. After a five-year selection

process, during which 15 competing encryption standards were evaluated, NIST announced AES in 2001. [28] AES represented a substantial improvement over DES and 3DES, offering 128, 192, or 256-bit encryption options, compared to the 56-bit key of DES and the effective 112-bit key length of 3DES. [29]

Twenty-two years after its introduction, AES remains the de-facto symmetric key algorithm and is used in almost every aspect of the Internet and computing. Disk and file encryption, HTTPS and TLS, VPNs, Wi-Fi security including WPA2 and WPA3, financial transactions, and cloud computing all rely on AES specifically as a secure method of transmitting data. If AES were compromised, there would be devastating consequences. There have been some minor developments to cracking AES encryption in less than brute force time, but none to date have threatened the security of AES in any significant way.

2.2.7 Elliptic Curve Cryptography

Elliptic Curve Cryptography is a modern public key cryptosystem that became popular in the early 2000s based on the algebraic structure of elliptic curves over finite fields rather than the multiplication of large prime numbers. ECC can offer the same levels of encryption as RSA with much smaller key sizes, which helps optimize efficiency and energy consumption. [30]

ECC is widely used on the modern Internet for various applications such as TLS for web browsing, cryptocurrency wallets, and mobile device encryption. ECC is prevalent in applications with limited resources, such as embedded systems or mobile devices, due to its higher efficiency than RSA. [30]

2.2.8 Non-cryptographic Cybersecurity

While most of this study's focus on cybersecurity relates to cryptography, it is important to note the vast array of security measures outside of cryptography that protect information

systems. Non-cryptographic security methods encompass various techniques and technologies, including network security protocols, intrusion detection systems, comprehensive security policies, and staff training.

Network security protocols such as firewalls and routing are critical to controlling data access and preventing unauthorized access. Intrusion detection systems exist to identify potential security breaches by monitoring hosts and the network for suspicious activity. Comprehensive security policies and training are vital for establishing a robust security culture within the organization, addressing the human element of cybersecurity.

The culmination of these measures forms a multi-layer strategy to protect the integrity and confidentiality of data in a constantly changing landscape where cryptographic security alone isn't sufficient.

3.0 Quantum Computing Essentials

3.1 History of Quantum Computing

3.1.1 The Quantum Turing Machine

The concept of the quantum computer dates back to 1980 when physicist Paul Benioff proposed a quantum mechanical model of the Turing machine, a device theorized by scientist Alan Turing in 1936. A Turing machine consists of a tape of infinite length divided into sections where each section can contain a character or be blank. A device can then look at one space on the tape at a time. Based on predetermined instructions or states, the device can read the tape, change or erase the character on the tape, and then move any number of spaces to the right or left. The machine will follow its predetermined instructions until an end state is reached. Before the instructions are executed, the tape contains the input information, and after the process is over, the tape contains the solution to the problem.

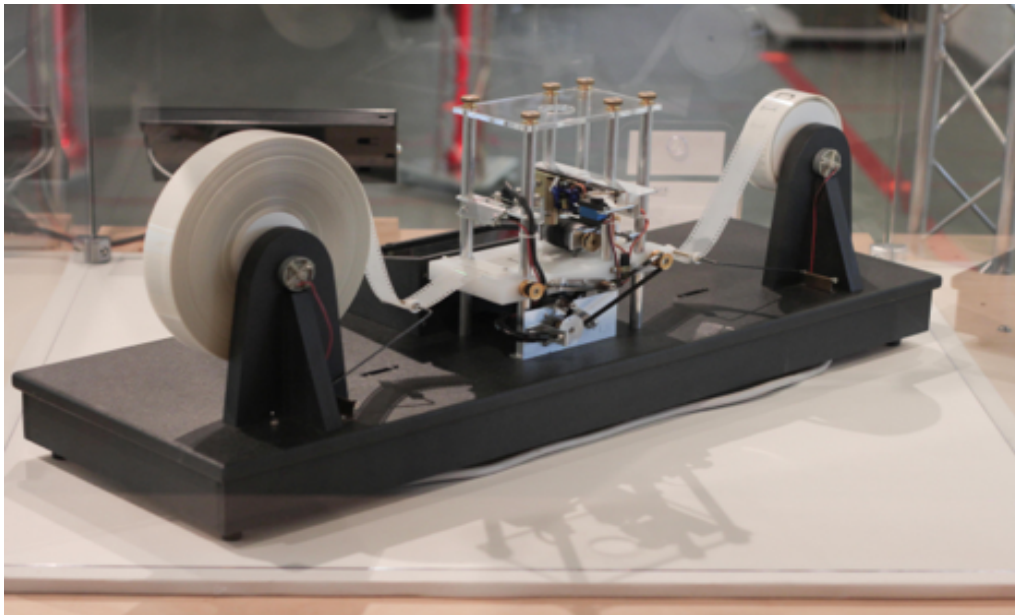


Figure 4: A model of a Turing Machine [31]

Benioff's quantum model of the Turing machine revolutionized this concept by introducing qubits to this simplified computer model, which leverages quantum theory to solve

more than one calculation at a time. Shortly after Benioff's initial proposition, theoretical physicist Richard Feynman proposed that the quantum computer could solve problems that classical computers couldn't. [32] Feynman later developed the method of modeling quantum circuits known as quantum circuit notation. [33]

3.1.2 The First Quantum Computer

In 1998, Isaac Chuang, Neil Gershandfeld, and Mark Kublinec created the first implementation of a working quantum computer that could be loaded with a state, run computations in fewer steps than a classical computer, and read out the final state. [2] This machine consisted of two qubits and was the first full implementation of its kind. While the quantum computer couldn't produce any meaningful results or solve problems that classical computers couldn't already calculate, this was still a breakthrough and proof that this technology could one day become ubiquitous with further research and development.

3.1.3 Advancements and Milestones

Following the debut of the first quantum computer, the field saw a wave of advancements and breakthroughs, driving the technology further. The quality of qubits has seen notable improvements, including longer coherence times, enhanced techniques for noise reduction, sharper control over quantum states, and improved error correction. One of the biggest hurdles in quantum computing is dealing with interference, which can ruin the accuracy of the results. Despite improving noise reduction methods, quantum error correction remains essential to refine and validate the outcomes produced by quantum circuits. Nowadays, quantum computing adopts a fault-tolerant approach, acknowledging that qubits can fail and planning for these contingencies to ensure reliable operations.

3.1.4 *The Race for Quantum Supremacy*

The years following the debut of the first quantum computer were marked by a shared objective: to build a machine capable of achieving quantum supremacy. This term refers to the ability of a quantum computer to solve a problem that is practically infeasible for classical computers. In 2019, Google claimed to have reached this milestone with its Sycamore processor.

In the 2019 *Nature* article titled “Quantum Supremacy Using a Programmable Superconducting Processor,” Google's engineers detailed how they used their 54-qubit Sycamore processor to perform a calculation that would be impossible for classical computers. The task involved random quantum circuit sampling—a task in which quantum computers have a significant advantage. Google asserted that this task would take the world's fastest supercomputer about ten thousand years to complete, whereas their quantum processor achieved the correct result in just 200 seconds. [34]

However, this achievement, particularly Google's claim regarding the time comparison with classical supercomputers, has been met with skepticism. Critics, including IBM, considered the ten-thousand-year estimate to be an exaggeration. A 2022 publication revealed that the same problem solved by Google's quantum machine could be completed in 15 hours using a cluster of 512 GPUs. Furthermore, it suggested that the task might take only a few dozen seconds on the latest, most advanced hardware available. [35]

3.1.5 *Modern Quantum Computing*

Currently, there are an abundance of companies that are publicly working on quantum computing. IBM, Google, Microsoft, and Amazon are all heavily invested in developing quantum hardware since it promises to be lucrative if the technology reaches a point of practicality. All four companies offer quantum infrastructure in the cloud, allowing users to

program quantum experiments using the APIs and frameworks each respective company has developed. Each company operates some quantum computers using their latest processor developments, allowing their cloud platform users to run code on quantum hardware and simulators.

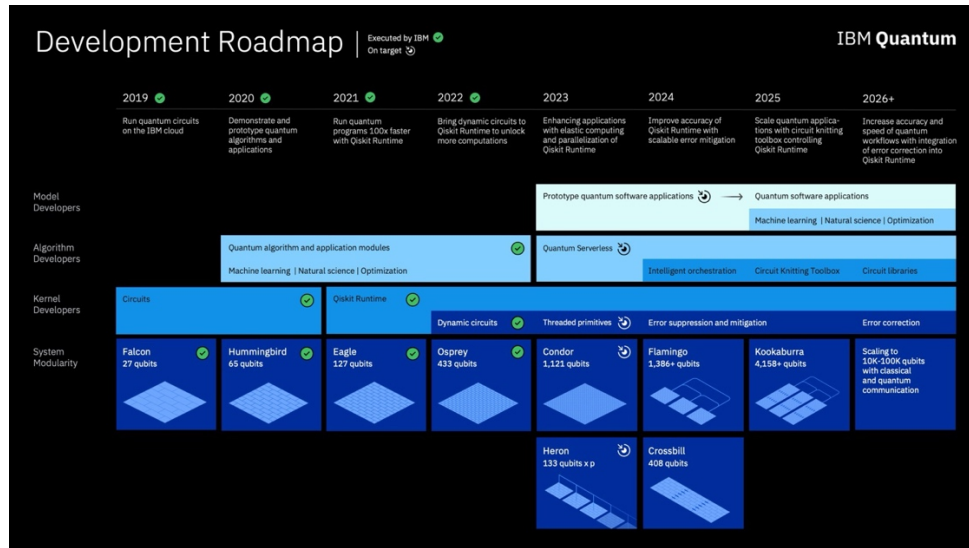


Figure 5: IBM Quantum Roadmap 2022 [36]

The amount of effort these companies dedicate to establishing themselves in the quantum field demonstrates their desire to lead the industry as the technology matures. Their commitment signals not only their belief that this technology is a major part of the future of computing but also that there is significant market potential for being one of the first companies to offer quantum services to willing customers.

3.1.6 Quantum Computing in Other Industries

Many companies, specifically those in healthcare and material science, have closely monitored the quantum industry. The promised power of future quantum computers could prove to run programs that deal with folding or material synthesis, tasks that require immense processing power on classical computers, much more efficiently.

In 2023, Cleveland Clinic and IBM unveiled the first quantum computer dedicated to healthcare research. With its 127 qubit ‘Eagle’ processor, an IBM Quantum System One is now installed at the Cleveland Clinic main campus. It will be used as part of the accelerator project Cleveland Clinic and IBM announced two years prior, intending to expedite the pace of biomedical research. [37]



Figure 6: The IBM Quantum System One at Cleveland Clinic’s Main Campus [37]

Cleveland Clinic is likely the first of many private organizations, whose chief focus is not quantum computing, to begin investing in quantum technology. However, it is likely that adoption will only start to accelerate once there is more clear evidence of quantum supremacy.

3.1.7 Challenges and Outlook

Even with the proliferation of quantum web services and the beginnings of quantum in private industry, there are still significant challenges the quantum industry will need to rectify

before it becomes a mainstay. Quantum is said to be in its Noisy Intermediate Scale Quantum [NISQ] era. This is a term coined by physicist John Preskill in 2018. Preskill stated that while quantum computing is making significant advancements, and 100 qubit quantum computers would soon become standard tools for exploring ‘many-body’ quantum physics, this grade of quantum machine will not be able to achieve proper noise reduction and error correction required to perform the tasks that quantum computers have promised to deliver. His paper aimed to manage expectations that quantum computing must still mature in many ways; while NISQ quantum computers likely won’t be the ones achieving quantum supremacy, they are a stepping stone toward something much bigger. [38]

3.2 Quantum Computing Fundamentals

3.2.1 Qubits

In classical computing, binary digits [bits] are used as the foundation for storing information. A single bit can hold a value of 0 or 1. What makes quantum computers fundamentally different is the usage of quantum binary digits [qubits]. Like regular bits, qubits can hold a value of 0 or 1, but they can also have a third state in which it is simultaneously in both of the first two states – this is known as superposition. When a qubit is in the superposition state, it has a particular unknown probability of being 0 and a probability of being 1. The superposition is resolved only once the quantum bit is measured and the actual state is known. Unlike a classical system where measurement reveals the pre-existing state (i.e., a 0 or 1), measurement in a quantum system actively determines the system's state. This concept is often referred to as the collapse of the quantum state.

3.2.2 Entanglement

Another pillar of quantum computing is the concept of entanglement. Entanglement is when two particles are connected across any distance. These two particles share a quantum state; if one changes, the other will change predictably. A significant implication of entanglement in modern quantum computing is the notion that entanglement can be used for error correction in qubits. [39] When two particles are entangled, a measurement of one of the particles will instantly determine the other particle's state, defying the principle that nothing can travel faster than the speed of light.

3.2.3 Quantum Interference

Interference is a crucial mechanism for controlling quantum states. It works by superimposing multiple quantum states, which amplifies paths leading to the correct answer and cancels paths leading to wrong answers. [40] This concept stems from the wave-like nature of quantum particles, where waves can constructively and destructively interfere with computation. Constructive interference strengthens the probability of correct answers and destructive interference. Interference is critical for running Shor's or Grover's algorithms and is a beneficial phenomenon in quantum computing. Interference shouldn't be confused with decoherence, which disrupts quantum states and degrades computational performance.

3.2.4 Coherence and Decoherence

Coherence in quantum computing refers to the ability of a quantum system to maintain quantum states over time. Stability in superposition and entanglement are critical for a quantum system. Coherence is measured by how long a quantum state can remain undisturbed before it decoheres because of interference from the outside world, one of the most significant challenges

in the quantum computing landscape. The longer the coherence in a quantum machine, the more calculations and operations can be performed on the qubits that can be executed reliably.

Quantum computing systems are susceptible to environmental noise, electromagnetic disturbances, and temperature fluctuations; these adversities cause decoherence, where the quantum states degrade, and accurate computation cannot occur.

The two primary errors due to decoherence are bit flip and phase flip. [41] A bit-flip error in quantum computing is analogous to a bit-flip error in classical computing, where a bit state changes from 0 to 1 or vice versa. A phase flip is a more complex error that changes how the phases of relative states constructively or destructively interfere with each other. This can significantly disrupt the flow of the quantum system and ruin results.

3.2.5 Quantum Error Correction

Because mitigating decoherence is so challenging, much research and development has been put into fighting the causes of decoherence and accounting for errors caused by decoherence through Quantum Error Correction [QEC]. QEC is responsible for dealing with the multitude of inaccuracies that result from the error-prone nature of quantum state manipulation. One of the nuances of QEC that makes it so difficult is the need to measure qubit indirectly. Since measurement collapses the quantum state of a qubit, sophisticated techniques have been developed to distribute the quantum state of one qubit to multiple other qubits, which provides redundancy to correct errors indirectly.

In 1995, Peter Shor published an article detailing a scheme for quantum error correction that utilizes nine other qubits to recover quantum information if one of the qubits decoheres. [42] This was a breakthrough as there had not previously been a method to preserve quantum information through decoherence. It is impossible to copy a qubit's state without measuring it.

Shor's way cleverly circumvents measuring the qubit's state by spreading information among the other qubits without copying the quantum state.

4.0 Quantum Threats

4.1 Shor's Algorithm

4.1.1 Overview

Introduced in 1994 by Peter Shor, Shor's algorithm represents a significant milestone for quantum computing since it could harness the unique advantages of quantum computing to complete a task that can't be done on a classical computer. Shor's algorithm can efficiently solve integer factorization problems.

In classical computing, calculating the factors of large numbers is a computationally expensive and time-consuming endeavor since no algorithm can efficiently solve these problems that can run on a binary system. Shor's algorithm leverages the principles of superposition and entanglement found in quantum computers to calculate possibilities simultaneously.

4.1.2 Threats to RSA and ECC

Public-key cryptography systems such as RSA and ECC, which are used ubiquitously throughout our internet-connected world, are based on the fundamental concept that products of large primes are challenging to factor. In RSA, the public key 'N' is calculated as the product of two large prime numbers, 'p' and 'q'. RSA is secure because it is impossible to factor the public key. If an attacker were to factor the public key hypothetically, they could then calculate the 'p' and 'q' values to calculate the private key. Once an attacker has the private key, they can decrypt any message encrypted with the public key, and the system is compromised. Shor's algorithm can also be adapted to solve discrete logarithm problems and prove effective against ECC systems. [3]

4.1.3 Time Complexity and Efficiency

In classical computing, no method exists to factorize large integers in polynomial time. The General Number Field Sieve is the most efficient algorithm in sub-exponential time. [4] While it is the best algorithm for the job in classical computing, it still takes a prohibitively long time to factorize long numbers such as those used in the RSA cryptosystem. Currently, the integer factorization problem is classified as NP, which implies that solutions to the problem can be easily verified, but they still need to be solved.

On the other hand, Shor's algorithm can solve these problems in polynomial time when run on a suitable quantum computer. This enables a dramatic reduction in time complexity and a leap in efficiency. Shor's algorithm leverages the computational advantages of quantum machines to process an abundance of calculations simultaneously, a feat impossible of a classical computer. This algorithm exemplifies how a quantum computer could reach quantum superiority with the proper hardware and changed how the world must think about cryptography.

4.2 Grover's Algorithm

4.2.1 Overview

Developed by Lov Grover in 1996, Grover's algorithm is another landmark in quantum computing that showcases how a quantum computer can leap forward in speed and efficiency over classical machines. The algorithm is designed to search through databases efficiently.

In classical computing, searching an unsorted database has a linear time complexity; each item must be checked individually until the desired item is found. Grover's algorithm leverages a quantum computer's superposition and entanglement properties to examine multiple possibilities simultaneously, dramatically speeding up the search process. [43]

4.2.2 Data Security Implications

While Grover's algorithm is mainly designed for fast, unsorted database searching, it can also weaken the security of symmetric encryption algorithms that rely on the difficulty of searching an ample key space. Simply put, guessing the key is the only way to break AES. Classical computers must check every key until they discover it. This is a time-consuming process in large key spaces. An AES 128-bit key has 2^{128} possible keys in the key space, so up to 2^{128} operations are required [$O(N)$ where N is the key length].

However, the number of operations needed is effectively halved using Grover's algorithm, offering a quadratic speedup over classical searching. A 128-bit AES would require 2^{64} operations on a quantum computer. While this is still a significant number of computations, remember that DES, a 64-bit encryption system, was broken using the EFF DES Cracker in 1998, and classical machines have only become more powerful.

4.3 Timelines

The amount of time before RSA, ECC, and AES are severely threatened by quantum computing heavily depends on how quickly quantum technology develops.

4.3.1 Shor's Algorithm Requirements

To compute the two factors of the public key in RSA, Shor's algorithm requires $2n$ logical qubits where n is the key length. For example, to crack a 2048-bit RSA, 4096 logical qubits would be needed. However, the number of logical qubits is theoretical; in reality, many physical qubits would be required for each logical qubit for proper error correction.

4.3.2 Grover's Algorithm Requirements

Unlike Shor's algorithm, Grover's algorithm has a different qubit requirement, which requires enough qubits to represent every possibility in the key space. This is simply n qubits in a

2^n space. For example, to find an AES 128-bit key, a 128 logical-qubit quantum processor would be needed. Adding more qubits to a quantum computer over the number of bits in the key isn't necessary and can even degrade performance. Speed can increase linearly by dividing the problem into multiple parts and having multiple quantum computers, each running the algorithm on their dedicated key section. A linear speedup on top of the quadratic speedup offered by Grover's algorithm is an ideal scenario, as configuring multiple quantum computers in parallel could have performance impacts.

4.3.3 Quantum Performance Trends

While quantum computers have groundbreaking potential, they currently face significant limitations, especially regarding error rates and fidelity. NISQ quantum computers cannot accurately run Shor's and Grover's because of the lack of fault-tolerant quantum computing capabilities. The most considerable challenge is maintaining coherence long enough to perform computations.

Despite these challenges, IBM, Google, and many other companies continue to push forward and develop their quantum technologies, focusing on improving qubit counts, fidelity, and error correction. The timeframe for fault-tolerant quantum machines to reach the precision needed to run Shor's or Grover's algorithms is unclear. However, Google's reported achievement of supremacy and IBM's steady progress following their roadmap show that significant strides are still being made in the industry. The developments from these companies aren't simply incremental; they represent significant strides toward realizing the full potential of quantum computing.

5.0 Quantum Resistant Cybersecurity Practices

5.1 Replacements for Private Key Cryptography

Public-key cryptosystems such as RSA and ECC are particularly vulnerable to quantum attacks compared to symmetric systems. As fault-tolerant quantum computers become a reality, these asymmetric systems must be replaced with quantum-resistant algorithms.

5.1.1 NIST Post-Quantum Cryptography Selection Process

NIST is currently in the years-long process of selecting a replacement quantum-resistant cryptography system similar to how they chose AES to replace DES.

There are currently four algorithms that will likely be the future of quantum-resistant public key cryptography systems, which NIST expects to be finalized as part of the post-quantum standard:

- CRYSTALS-Kyber
- CRYSTALS-Dilithium
- FALCON
- SPHINCS+

NIST states that CRYSTALS-Kyber will be used as the algorithm for general encryption, and the other three will be used for digital signatures. CRYSTALS-Dilithium will be the primary algorithm for digital signatures, and FALCON will be for more minor signatures. SPHINCS+ is a slower algorithm but uses hashes instead of structured lattices like the other three selections.

[44]

5.1.2 Structured Lattices and Hash Functions

Structured lattice problems are a promising approach to developing quantum-resistant algorithms. Rather than traditional cryptography systems, which rely on number-theoretic problems, lattice-based cryptography is based on problems in lattice theory that are believed to

be quantum-resistant. Lattice cryptography relies on the computational difficulty of problems such as the Shortest Vector Problem and Closest Vector Problem, computation challenges in which quantum computers have no mathematical advantage. [45]

The SPHINCS+ cryptography system takes a different approach and leverages the inherent properties of hash functions. Hash functions are designed to be one-way and impossible to reverse. Even with quantum computing, reverse engineering a hash function to obtain the input is a significant challenge.

5.2 Strengthening AES

While less susceptible to quantum attacks than asymmetric cryptography, AES still has associated risks if a powerful enough quantum machine that could run Grover's algorithm accurately and efficiently were to be developed. It has been widely suggested that doubling the encryption key size, specifically from AES 128-bit to 256-bit, will be sufficient for the time being. A quantum computer wouldn't be able to crack AES 256 for the same reason a classical computer cannot break AES 128 – the key space is too large to brute force, even with the advantage of searching keys twice as fast on a quantum machine.

5.3 Other Quantum Enabled Concepts

In addition to developing quantum-resistant algorithms, other quantum-enabled concepts that promise to leverage the principles of quantum mechanics to enhance security throughout the cybersecurity landscape have emerged.

5.3.1 *Quantum Key Distribution*

Quantum key distribution [QDK] is a secure communication method that uses quantum mechanics to exchange cryptographic keys to protect against eavesdropping.

QDK uses superposition and entanglement to encode information, and any attempt to intercept the information will alter the quantum states of the information, which will signal the presence of an eavesdropper. QDK uses no computational difficulty to encrypt and decrypt information, so it resists brute force and quantum attacks. QDK could replace key exchanges currently performed by public-key cryptosystems such as RSA and ECC. [46]

QDK requires physical infrastructure, typically in the form of fiber optic networks and quantum transmitters and receivers used to communicate quantum states in photons. There would be challenges to integrating QDK as a mainstream infrastructure, chiefly due to the cost of building out fiber networks. Implementing QDK over existing TCP/IP fiber networks with additional equipment is possible. Currently, NIST recommends against using QDK for protection in national Security Systems. It believes post-quantum cryptography will be a more cost-effective and easily maintained solution. [47]

5.3.2 Quantum Random Number Generation

Quantum Random Number Generation [QRNG] is the concept that leverages the randomness found in quantum phenomena to generate truly random numbers. This would be a significant breakthrough in cryptography since algorithms in classical random number generation are often predictable and can be reverse-engineered.

Multiple types of quantum phenomena are used in QRNG, including photon polarization, superposition, and radioactive decay. Each of these aspects can be measured to convert their values into digital data, and that data is output as a stream of random numbers that can be further processed.

6.0 Testing Hadamard Gates and Quantum Fourier Transforms via IBM Qiskit

6.1 Background

6.1.1 IBM Cloud

IBM Cloud is one of the largest cloud service providers, along with Amazon Web Services and Microsoft Azure. However, it is also the foremost cloud-based quantum computing platform. IBM maintains Qiskit, a comprehensive framework that provides developers, mathematicians, and businesses a toolkit to create applications to leverage quantum hardware for bleeding-edge workflows and applications in various fields.

IBM manufactures quantum processors and computing systems and hosts several quantum simulators and real quantum computers within its cloud platform. These resources allow users to refine their algorithms in a controlled and optimized virtual environment. Most significantly, however, IBM Cloud provides access to physical quantum machines where users can run their quantum code on real quantum hardware. This access democratizes quantum computing, making it accessible to a broader range of users, from individuals to corporations, who want to experiment with this groundbreaking technology.

6.1.2 Fourier Transform and Quantum Fourier Transform [QFT]

The Fourier Transform is a mathematical concept that converts a function into a form that describes the frequencies in the original function. This transformation is commonly used in signal processing, for instance, extracting individual tones from a musical chord. Qubits in a superposition state can be considered complex signals like tones in a chord; a QFT converts the superposition to a new state that reveals the frequency of the original state.

Just as a Fourier Transform helps extract the frequencies in a chord, the QFT extracts the quantum frequencies, known as periodicities. These periodicities allow quantum algorithms to

perform calculations efficiently. For example, in Shor's algorithm, the QTF is used to determine the periodicity of a modular exponentiation function. This function plays a crucial role in the factorization of large numbers. [48]

6.1.3 Hadamard Gate

The Hadamard gate is a core operation in quantum computing necessary for transitioning qubits from their base state to a superposition state. By applying it to a qubit in either binary state, the Hadamard gate produces an equal superposition of both states, which sets up equal probabilities upon measurement.

The Hadamard gate's importance extends into Grover's and Shor's algorithms. In Grover's algorithm, it is used to establish uniform superposition such that probabilities of outcomes can be amplified through quantum interference. In Shor's algorithm, Hadamard gates are used to initialize qubits before the QFT stage. [49]

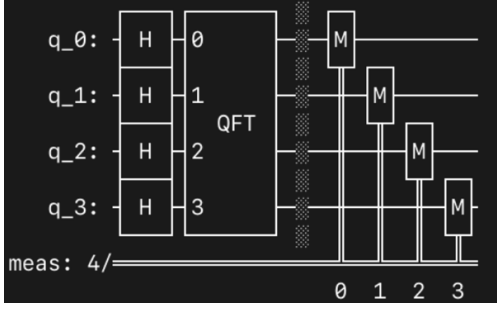
6.2 Creating a QTF Program with Python and Qiskit

The IBM Qiskit framework provides a comprehensive library of tools for developing quantum programs. Qiskit allows for creating quantum circuits that can be executed on quantum simulators or hardware. In the following experiment, multiple variations of a similar quantum circuit will be explored, as well as how specific gates affect the output of the circuit.

Each circuit variation will be run on three quantum interfaces – a local simulator, a remote IBM Cloud simulator, and a remote IBM Quantum Computer. Testing the various circuits on each quantum interface will allow for a deeper understanding of how the various gates affect output on circuits in simulated and real-world performance.

Table 1: Variations of Quantum Circuits

Qubits	Hadamard	QFT	Measurement	Diagram
4	NO	NO	YES	
4	YES	NO	YES	
4	NO	YES	YES	

4	YES	YES	YES	
---	-----	-----	-----	--

6.3 Executing the QTF Program

Each quantum circuit will be tested on three interfaces: a local simulator, a remote simulator in IBM Cloud, and a remote quantum computer in IBM Cloud. The local simulator is the Qiskit ‘AER’ backend, the remote simulator is a cloud resource named `ibmq_qasm_simulator`, and the real quantum computer is a 127 qubit cloud resource named `ibmq_brisbane`. The local and remote simulators should be expected to perform the same with the simple quantum circuits that this experiment runs. However, the hardware on which the remote simulator is hosted is likely much more powerful than a desktop computer; the addition of the remote simulator will add an additional dimension to the experiment.

Table 2: Simulators and Quantum Hardware

Type	Cloud Resource Name	Qubits	Cost per/sec
Aer Local Simulator			\$0.00
IBM Remote Simulator	<code>ibmq_qasm_simulator</code>		\$0.00
IBM Quantum Hardware	<code>ibmq_brisbane</code>	127	\$1.60

6.4 Results

6.4.1

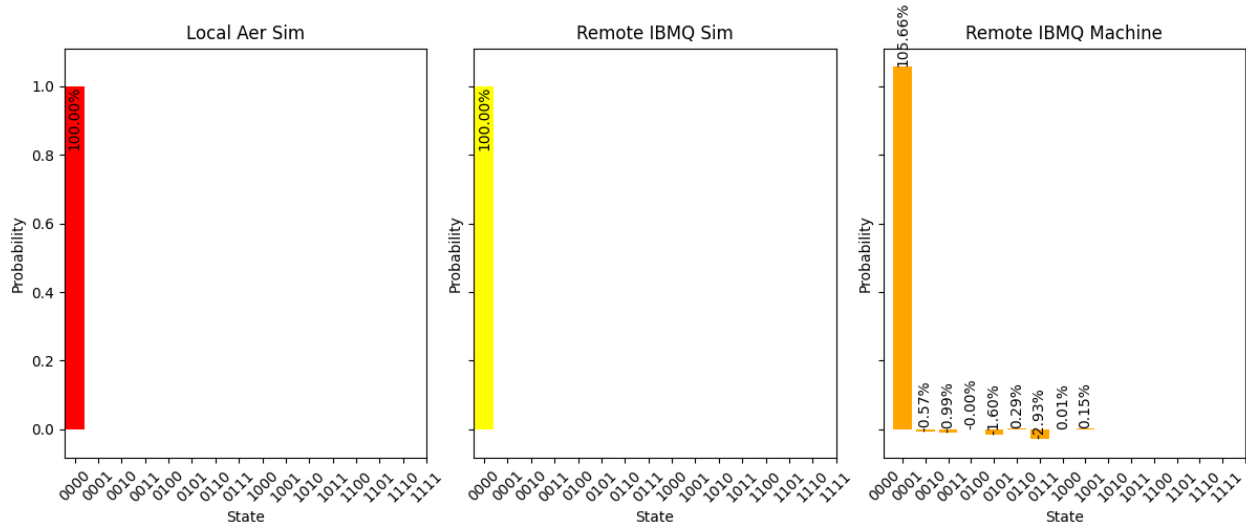
In the following experiment, a series of tests were executed using an interactive custom Python script. The script allowed for configuring the quantum circuit in each test. In every test, the qubits were initialized to the ‘0000’ state, providing a controlled starting condition. Additionally, each test was run with 1024 shots, or the number of times the quantum circuit will run in each test to measure the probabilities more accurately.

The Python script allows the user to configure various aspects of the experiment, including setting the number of qubits, turning on or off Hadamard gates and QFT, selecting the endpoint for the quantum job, and setting a name for a JSON file containing the results from the test. This customization streamlined the process of modifying the circuit and endpoint configurations for every test.

6.4.2 No Gates Applied

In the first test, a quantum circuit was created with four qubits, which were measured with no gates applied to them. Under these conditions, each of the four qubits is expected to remain in their original state as no operations are performed.

Graph 1: No Gates Applied to Four Qubits

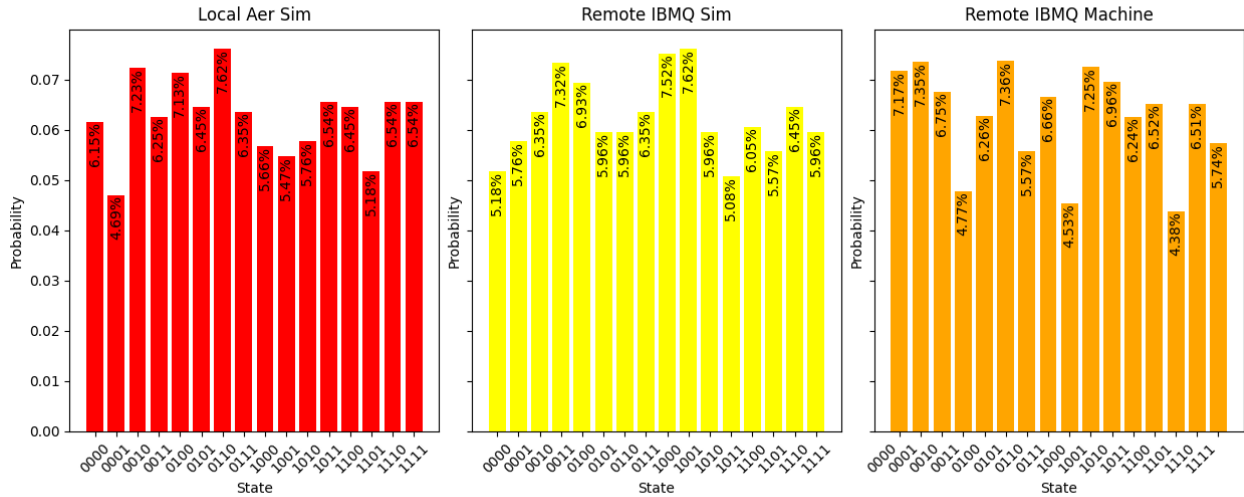


In the experiment, the two simulated machines returned the expected results. However, when testing on the quantum hardware, unusual results emerged. The probability listed for the ‘0000’ state exceeded 100%. The sum of the probabilities still adds to 100%, factoring in the negative probabilities for some of the other states. These anomalies can likely be attributed to various factors such as environmental noise, errors in quantum gates, and decoherence. This result highlights some of the current limitations in quantum hardware, especially the lack of precision for practical applications such as running Shor’s algorithm.

6.4.3 Hadamard Gate Applied

The second test involves applying a Hadamard gate to each of the four qubits and measuring them. Applying the Hadamard gate puts the qubits in a state of superposition where there is a uniform superposition of the qubits being in either state. The expected result of this test would be for an even distribution of probabilities across the sixteen possible states.

Graph 2: Hadamard Gate Applied to Four Qubits

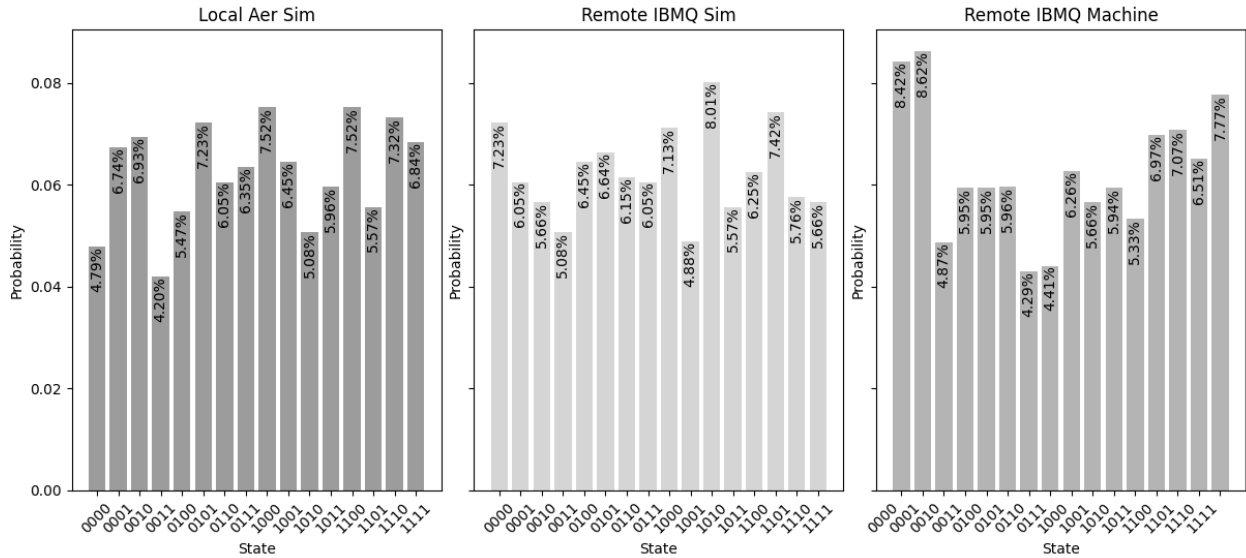


This test shows similar results across each of the quantum interfaces. While the data appears random, there are different variabilities among each quantum interface. Based on how this experiment was conducted, it would be challenging to accurately attribute the differences in variability to specific quantum interfaces without re-running the test with more qubits. Future experiments with more qubits could provide more comprehensive insight into how the distribution changes when a circuit has more or fewer qubits.

6.4.4 Quantum Fourier Transform Applied

The next test involves running a QFT with no superposition state applied. The objective of the test is to observe the effect of a QFT while the qubits are in their base states. The expected result of this test is that the qubits will remain in their base states since the QFT relies on superposition to function. If the expected outcome is achieved, the results should appear similar to the first test, where no gates were applied to the qubits.

Graph 3: Quantum Fourier Transform Applied to Four Qubits



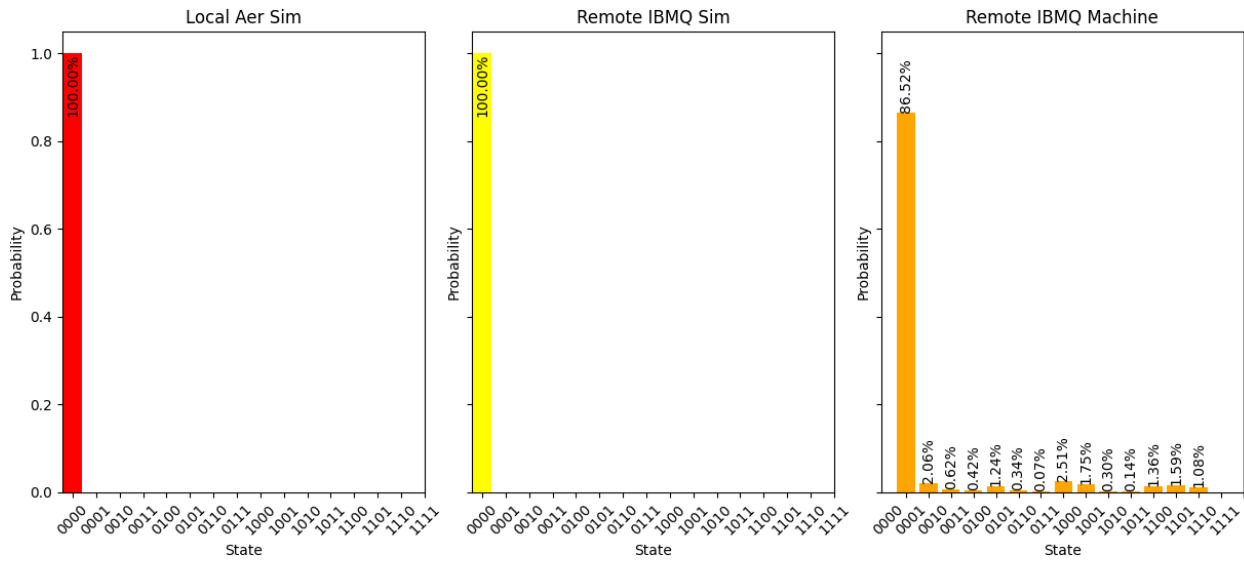
Contrary to the initial expectations, executing a QFT on qubits in their base states resulted in a seemingly random distribution of probabilities similar to the results in the test where just the Hadamard gate was applied, and the qubits were measured. After further investigation, no explanation for these results could be identified. Environmental noise, gate errors, and decoherence aren't causes since the anomaly was observed on all three quantum interfaces, and the simulated interfaces wouldn't be susceptible to those issues. Due to the inability to identify a cause, these results will not be included in the conclusions of these tests to preserve the integrity of the other tests and overall findings.

6.4.5 Hadamard Gate and QFT Applied

The final test involves a quantum circuit with both a Hadamard gate and a Quantum Fourier Transform. In this configuration, the Hadamard gate is first applied to the qubits to put them in a superposition state. Then, they go through the QFT. This should effectively reverse the Hadamard gate and result in the base states of the qubits being observed.

This expectation is based on the principle that the QTF reverses the superposition created by the Hadamard gate. The test will help analyze the effect quantum gates have on each other and the dynamics of the quantum computing process.

Graph 4: Hadamard Gate and Quantum Fourier Transform Applied to Four Qubits



The final test yielded expected results with some variances in the physical quantum hardware. The two simulated interfaces show a 100% probability of the base state, which aligns with the theoretical prediction that the QFT would reverse the effect of the Hadamard gate. However, when running the circuit on the remote quantum computer, the probability of measuring the '0000' state was only 86.52%, with the remaining probabilities spread across the other states. This result suggests an influence from outside factors similar to the initial 'no-gate' test. The slight differences between simulated and real hardware underscore the challenges in the quantum realm, specifically the precision required to achieve accurate results.

6.5 Conclusion of Hadamard and QFT Testing

In this set of experiments, four quantum circuit designs were executed across three quantum interfaces. These tests focused on the impacts of the Quantum Fourier Transform and

Hadamard gates on qubits. The results from the experiments from both the local and remote simulators generally matched expectations, especially on the final test with both Hadamard gates and QFT; both simulators showed qubits returning to their initial states.

However, the same test run on the quantum hardware showed notable deviations from the expected results. It is likely that factors such as environmental noise, gate errors, and decoherence contributed to the anomalies in the quantum hardware and demonstrated the challenges and opportunities of quantum in the NISQ era. Understanding and developing ways to address these issues, specifically quantum error correction will be a critical task for the individuals and companies working to advance quantum technologies.

Overall, the Hadamard and QFT testing showed practical examples of how these different quantum gates affect qubits in different circuits, as well as the imperfections of running circuits on quantum hardware when compared to a quantum simulator.

7.0 Summary

As this investigation into the convergence of quantum computing and cybersecurity concludes, the immense potential of the quantum field becomes increasingly evident.

Cybersecurity strategies must be modified to safeguard information in the quantum era.

While cybersecurity has evolved significantly since the dawn of the internet, the encryption algorithms used today to protect all confidential information throughout the internet have remained essentially unchanged over the last two decades. Quantum algorithms such as Grover's and Shor's demonstrate applications beyond the strictly theoretical that threaten the current cryptographic practices.

Public key algorithms such as RSA and ECC are the backbone of digital security. RSA relies on the difficulty of factorizing large numbers, and ECC relies on the discrete logarithm problem; both mathematical problems are vulnerable to Shor's algorithm on a suitably equipped quantum computer.

The National Institute of Standards and Technology [NIST] is currently standardizing the cryptographic systems that will replace RSA and ECC for public key cryptography applications. These algorithms, such as CRYSTALS-Kyber, CRYSTALS-Dilithium, FALCON, and SPHINCS+, use lattice-based and hash-based encryption, which are resistant to quantum computers and classical computers alike.

Quantum-enabled technologies such as Quantum Key Distribution [QDK] and Quantum Random Number Generation [QRNG] have also emerged as critical components for modern cybersecurity. QDK allows for transmitting encryption keys without public key cryptography, and QRNG leverages quantum phenomena to generate truly random numbers, a task impossible for classical computers.

In the experiments involving Qiskit and quantum circuits, fundamental aspects of quantum computing were explored, particularly focusing on the behavior of quantum operations across simulators and real quantum hardware. The observed discrepancies between the results from simulators and quantum hardware highlight the unpredictability inherent in NISQ [Noisy Intermediate-Scale Quantum] hardware. Despite these challenges, industry leaders such as IBM and Google are persistently advancing their quantum technologies, indicating that the development of fault-tolerant quantum systems is a foreseeable achievement.

In summary, the quantum computing era presents both challenges and opportunities. Staying informed and adaptable is key to navigating the changing landscape, harnessing the power of quantum computing, and protecting the digital world against its potential threats.

Sources

- [1] Caltech Faculty, “What Is Quantum Physics?,” Caltech Science Exchange. Accessed: Nov. 06, 2023. [Online]. Available: <http://scienceexchange.caltech.edu/topics/quantum-science-explained/quantum-physics>
- [2] I. L. Chuang, N. Gershenfeld, and M. Kubinec, “Experimental Implementation of Fast Quantum Searching,” *Phys. Rev. Lett.*, vol. 80, no. 15, pp. 3408–3411, Apr. 1998, doi: 10.1103/PhysRevLett.80.3408.
- [3] P. W. Shor, “Algorithms for quantum computation: discrete logarithms and factoring,” in *Proceedings 35th Annual Symposium on Foundations of Computer Science*, 1994, pp. 124–134. doi: 10.1109/SFCS.1994.365700.
- [4] J. P. Buhler, H. W. Lenstra, and C. Pomerance, “Factoring integers with the number field sieve,” in *The development of the number field sieve*, A. K. Lenstra and H. W. Lenstra, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 1993, pp. 50–94.
- [5] P. Baran, “RELIABLE DIGITAL COMMUNICATIONS SYSTEMS USING UNRELIABLE NETWORK REPEATER NODES,” The RAND Corporation, P-1995, May 1960. Accessed: Apr. 10, 2022. [Online]. Available: <https://www.rand.org/content/dam/rand/pubs/papers/2008/P1995.pdf>
- [6] International Business Machines Corporation (IBM). Data Processing Division, “IBM 1440 Brochure.” 1962. Accessed: Jun. 01, 2022. [Online]. Available: <https://www.computerhistory.org/brochures/doc-4372956faadb0/>
- [7] S. Ruthfield, “The Internet’s history and development: from wartime tool to fish-cam,” *XRDS Crossroads ACM Mag. Stud.*, vol. 2, no. 1, pp. 2–4, Sep. 1995, doi: 10.1145/332198.332202.
- [8] J.-M. Robert and T. Chen, “The Evolution of Viruses and Worms,” in *Statistical Methods in Computer Security*, vol. 20041441, W. Chen, Ed., in Statistics: A Series of Textbooks and Monographs, vol. 20041441., CRC Press, 2004, pp. 265–285. doi: 10.1201/9781420030884.ch16.
- [9] Bolt Beranek and Newman Inc., “ARPA NETWORK, LOGICAL MAP, SEPTEMBER 1973.” Sep. 1973. [Online]. Available: https://upload.wikimedia.org/wikipedia/commons/3/3a/ARPA_Network%2C_Logical_Map%2C_September_1973.jpg
- [10] Ray Tomlison, “The First Network Email,” The First Network Email. Accessed: May 25, 2022. [Online]. Available: <https://web.archive.org/web/20060506003539/https://openmap.bbn.com/~tomlinso/ray/firstemailframe.html>
- [11] “NCP/TCP transition plan,” Internet Engineering Task Force, Request for Comments RFC 801, Nov. 1981. doi: 10.17487/RFC0801.
- [12] “Bulletin-board system | Online Forum, Message Board, Networking | Britannica.” Accessed: Nov. 20, 2023. [Online]. Available: <https://www.britannica.com/technology/bulletin-board-system>

- [13] T. Berners-Lee, "Information Management: A Proposal," CERN, Mar. 1989.
- [14] "What is the ILOVEYOU worm, what does it do, and how do I detect and remove it?" Accessed: Nov. 20, 2023. [Online]. Available: <https://kb.iu.edu/d/aioe>
- [15] D. Winder, "This 20-Year-Old Virus Infected 50 Million Windows Computers In 10 Days: Why The ILOVEYOU Pandemic Matters In 2020," *Forbes*. Accessed: Nov. 20, 2023. [Online]. Available: <https://www.forbes.com/sites/daveywinder/2020/05/04/this-20-year-old-virus-infected-50-million-windows-computers-in-10-days-why-the-iloveyou-pandemic-matters-in-2020/>
- [16] A. Press, "Microsoft Attacked by Worm, Too," *Wired*. Accessed: Nov. 20, 2023. [Online]. Available: <https://www.wired.com/2003/01/microsoft-attacked-by-worm-too/>
- [17] BetaFred, "Microsoft Security Bulletin MS02-039 - Critical." Accessed: Nov. 20, 2023. [Online]. Available: <https://learn.microsoft.com/en-us/security-updates/securitybulletins/2002/ms02-039>
- [18] "Stuxnet explained: What it is, who created it and how it works," www.kaspersky.com. Accessed: Nov. 10, 2023. [Online]. Available: <https://www.kaspersky.com/resource-center/definitions/what-is-stuxnet>
- [19] "Building a Cyber Secure Plant." Accessed: Nov. 10, 2023. [Online]. Available: <https://web.archive.org/web/20210421092726/https://www.totallyintegratedautomation.com/2010/09/building-a-cyber-secure-plant/>
- [20] E. MacAskill, G. Dance, F. Cage, G. Chen, and N. Popovich, "NSA files decoded: Edward Snowden's surveillance revelations explained," *the Guardian*. Accessed: Nov. 20, 2023. [Online]. Available: <http://www.theguardian.com/world/interactive/2013/nov/01/snowden-nsa-files-surveillance-revelations-decoded>
- [21] "Caesar Cipher in Cryptography," *GeeksforGeeks*. Accessed: Feb. 16, 2022. [Online]. Available: <https://www.geeksforgeeks.org/caesar-cipher-in-cryptography/>
- [22] Walther, *Chiffriermaschine "Enigma."* 1943. Accessed: Nov. 14, 2023. [Online]. Available: https://commons.wikimedia.org/wiki/File:Bundesarchiv_Bild_183-2007-0705-502,_Chiffriermaschine_%22Enigma%22.jpg
- [23] *158,962,555,217,826,360,000 (Enigma Machine) - Numberphile*, (Jan. 10, 2013). Accessed: Nov. 14, 2023. [Online Video]. Available: https://www.youtube.com/watch?v=G2_Q9FoD-oQ
- [24] W. Diffie and M. Hellman, "New directions in cryptography," *IEEE Trans. Inf. Theory*, vol. 22, no. 6, pp. 644–654, 1976, doi: 10.1109/TIT.1976.1055638.
- [25] R. L. Rivest, A. Shamir, and L. Adleman, "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems," *Commun ACM*, vol. 21, no. 2, pp. 120–126, Feb. 1978, doi: 10.1145/359340.359342.
- [26] "The Legacy of DES - Schneier on Security." Accessed: Nov. 15, 2023. [Online]. Available: https://www.schneier.com/blog/archives/2004/10/the_legacy_of_d.html

- [27] “The Electronic Frontier Foundation.” Accessed: Nov. 15, 2023. [Online]. Available: https://web.archive.org/web/20170507231657/https://w2.eff.org/Privacy/Crypto/Crypto_misc/DESCracker/HTML/19980716_eff_des_faq.html
- [28] I. T. L. Computer Security Division, “AES Development - Cryptographic Standards and Guidelines | CSRC | CSRC,” CSRC | NIST. Accessed: Nov. 20, 2023. [Online]. Available: <https://csrc.nist.gov/projects/cryptographic-standards-and-guidelines/archived-crypto-projects/aes-development>
- [29] C. Taylor, “3DES Encryption - CyberHoot Cyber Library,” CyberHoot. Accessed: Nov. 20, 2023. [Online]. Available: <https://cyberhoot.com/cybrary/3des-encryption/>
- [30] admin, “Elliptic Curve Cryptography,” GlobalSign. Accessed: Nov. 20, 2023. [Online]. Available: <https://www.globalsign.com/en/blog/elliptic-curve-cryptography>
- [31] Acosta, Rocky, “File:Turing Machine Model Davey 2012.jpg.” Oct. 21, 2012. Accessed: Apr. 11, 2022. [Online]. Available: https://commons.wikimedia.org/wiki/File:Turing_Machine_Model_Davey_2012.jpg
- [32] R. P. Feynman, “Simulating physics with computers,” *Int. J. Theor. Phys.*, vol. 21, no. 6–7, pp. 467–488, Jun. 1982, doi: 10.1007/BF02650179.
- [33] R. P. Feynman, “Quantum mechanical computers,” *Found. Phys.*, vol. 16, no. 6, pp. 507–531, Jun. 1986, doi: 10.1007/BF01886518.
- [34] F. Arute *et al.*, “Quantum supremacy using a programmable superconducting processor,” *Nature*, vol. 574, no. 7779, pp. 505–510, Oct. 2019, doi: 10.1038/s41586-019-1666-5.
- [35] F. Pan, K. Chen, and P. Zhang, “Solving the sampling problem of the Sycamore quantum circuits,” *Phys. Rev. Lett.*, vol. 129, no. 9, p. 090502, Aug. 2022, doi: 10.1103/PhysRevLett.129.090502.
- [36] “IBM Quantum Computing | Roadmap.” Accessed: Nov. 12, 2023. [Online]. Available: <https://www.ibm.com/quantum/www.ibm.com/quantum/roadmap>
- [37] N. Releases, “Cleveland Clinic and IBM Unveil First Quantum Computer Dedicated to Healthcare Research,” Cleveland Clinic Newsroom. Accessed: Nov. 16, 2023. [Online]. Available: <https://newsroom.clevelandclinic.org/2023/03/20/cleveland-clinic-and-ibm-unveil-first-quantum-computer-dedicated-to-healthcare-research/>
- [38] J. Preskill, “Quantum Computing in the NISQ era and beyond,” *Quantum*, vol. 2, p. 79, Aug. 2018, doi: 10.22331/q-2018-08-06-79.
- [39] M. Swan, R. P. dos Santos, and F. Witte, *Quantum Computing: Physics, Blockchains, and Deep Learning Smart Networks*, vol. 02. in Between Science and Economics, vol. 02. WORLD SCIENTIFIC (EUROPE), 2020. doi: 10.1142/q0243.
- [40] C. Mathas, “The basics of quantum computing—A tutorial,” EDN. Accessed: May 17, 2022. [Online]. Available: <https://www.edn.com/the-basics-of-quantum-computing-a-tutorial/>
- [41] J. Roffe, “Quantum error correction: an introductory guide,” *Contemp. Phys.*, vol. 60, no. 3, pp. 226–245, Jul. 2019, doi: 10.1080/00107514.2019.1667078.

- [42] P. W. Shor, “Scheme for reducing decoherence in quantum computer memory,” *Phys Rev A*, vol. 52, no. 4, pp. R2493–R2496, Oct. 1995, doi: 10.1103/PhysRevA.52.R2493.
- [43] L. K. Grover, “A fast quantum mechanical algorithm for database search.” 1996.
- [44] “NIST Announces First Four Quantum-Resistant Cryptographic Algorithms,” *NIST*, Jul. 2022, Accessed: Nov. 20, 2023. [Online]. Available: <https://www.nist.gov/news-events/news/2022/07/nist-announces-first-four-quantum-resistant-cryptographic-algorithms>
- [45] “A (somewhat) gentle introduction to lattice-based post-quantum cryptography – Cybersecurity-Blog.” Accessed: Nov. 21, 2023. [Online]. Available: <https://www.cybersecurity.blog.aisec.fraunhofer.de/en/a-somewhat-gentle-introduction-to-lattice-based-post-quantum-cryptography/>
- [46] A. Gillis, “What is Quantum Key Distribution (QKD) and How Does it Work?,” TechTarget. Accessed: Nov. 21, 2023. [Online]. Available: <https://www.techtarget.com/searchsecurity/definition/quantum-key-distribution-QKD>
- [47] “National Security Agency/Central Security Service > Cybersecurity > Quantum Key Distribution (QKD) and Quantum Cryptography QC.” Accessed: Nov. 20, 2023. [Online]. Available: <https://www.nsa.gov/Cybersecurity/Quantum-Key-Distribution-QKD-and-Quantum-Cryptography-QC/>
- [48] “Quantum Fourier Transform - an overview | ScienceDirect Topics.” Accessed: Nov. 24, 2023. [Online]. Available: <https://www.sciencedirect.com/topics/mathematics/quantum-fourier-transform>
- [49] M. Publications, “All about Hadamard Gates,” Manning. Accessed: Nov. 24, 2023. [Online]. Available: <https://freecontent.manning.com/all-about-hadamard-gates/>