Quantum Communications and Cryptography

Henry McNeil
hcmcneil@unomaha.edu
University of Nebraska, Omaha
Omaha, Nebraska

Zexi Xing
zxing@unomaha.edu
University of Nebraska, Omaha
Omaha, Nebraska

Casey Schmitz
caseyschmitz@unomaha.edu
University of Nebraska, Omaha
Omaha, Nebraska

Bryan Tomey
btomey@unomaha.edu
University of Nebraska, Omaha
Omaha, Nebraska

ABSTRACT

Quantum computers provide many new opportunities for cryptography, such as key distribution, confidentiality, integrity, and non-repudiation. Because of the destructive and probabilistic nature of quantum measurements and the no-cloning theorem which prevents the duplication of quantum states, quantum cryptographic protocols are dramatically different from classical cryptography. A large number of these protocols have been proposed, but little comparative work has been done. We perform a survey of existing protocols, with a focus on practical applications, in order to provide an overview of the current state of the field. We also present an implementation of a practical quantum cryptographic algorithm capable of providing confidentiality, integrity, and non-repudiation. In addition, we generalize our implementation process in order to provide a template for the translation of algorithms from a research paper into a quantum programming framework.

ACM Reference Format:
Henry McNeil, Zexi Xing, Casey Schmitz, and Bryan Tomey. 2021. Quantum Communications and Cryptography. In *CYBR-8950 Cybersecurity Capstone.* ACM, New York, NY, USA, 14 pages.

1 INTRODUCTION

Quantum computers have the world on the brink of a second computing revolution. By using the unique properties of subatomic particles, quantum computers will be able to solve complex problems in minutes which would take a classical computer thousands of years. Foremost among these problems are many which are foundational to modern cryptography, such as the prime factorization problem at the core of internet security. While current quantum computers do not have the level of sophistication necessary to break today's encryption systems, it is only a matter of time before they gain that capability. This project aims to explore the implications of quantum computing on cryptography in order to stay ahead of this anticipated threat and preserve the security of sensitive communications as used for everything from internet browsing to military instructions.

In highly critical communications systems, such as those used to initiate a military action, messages must have guaranteed delivery, must not be tampered with, and must be authenticated and correct. These systems must not allow for false messages, which could lead to widespread loss of life in the worst case, so non-repudiation is a primary concern. The goal of this project is to identify the best ways for a critical system to send confidential messages with a guarantee of integrity and non-repudiation in a world where quantum computers are prevalent and capable. These cryptographic mechanisms may include quantum key distribution, quantum digital signatures, and quantum-resistant encryption algorithms performed on a classical computer.

1.1 Background and Problem Context

In the era of 5G high-speed internet surfing, data can be transmitted almost instantly from one side of the world to another. The downside of this technological convenience is that the secret information between both sides of a communication can be easy to expose to a malicious third party. Fortunately, current cryptographic algorithms are sufficient to protect users' privacy because most of them are impossible to break with the computational power of traditional computers. However, when quantum computers become available, almost every currently used encryption algorithm will be vulnerable, and the massive data leakage will be inevitable. Hence, it is necessary to understand what quantum computers are and why they can destroy current existing cryptography.

A quantum computer is a type of physical device that is capable of bypassing many limitations of classical computers. Quantum computers take advantage of the principles of quantum mechanics to perform high-speed mathematical and logical operations to store and process qubit information.[41]. Just as traditional computers distinguish between zeros and ones, or bits, by switching on and off logical gates in an integrated circuit, quantum computers have their own basic unit, the quantum bit. In these quantum bits, or qubits, the classical bit states 0 and 1 are replaced by two quantum states, $|0\rangle$ and $|1\rangle$. In a quantum computer, these states may be represented by the two orthogonal polarization directions of a photon, the spin directions of an electron in a magnetic field, the two directions of a nuclear spin, or the different energy levels of an electron in an atom.

A more significant difference between bits and qubits is that qubits can exist in a superposition of two logical states. "Unlike classical bits, a quantum bit can be put in a superposition state

CYBR-8950 Cybersecurity Capstone, Spring 2021, Omaha, NE

that encodes both 0 and 1. There is no good classical explanation of superpositions: a quantum bit representing 0 and 1 can neither be viewed as 'between' 0 and 1 nor can it be viewed as a hidden unknown state that represents either 0 or 1 with a certain probability."[39] Qubits can also become entangled with one another, which causes a measurement of one qubit to affect others. Hence, based on these attributes of qubits, quantum computers are able to process information differently and efficiently solve a different class of problems than classical computers.

1.2 Motivation

Encryption is omnipresent in modern life, with uses ranging from internet browsing to medical devices, wireless car keys to nuclear control systems. Quantum computing threatens almost all current encryption protocols in one way or another, whether by effectively weakening key strength, or breaking the algorithm entirely. As such, an understanding of the significance of quantum computing and new cryptographic protocols is critical, with impacts in areas ranging from personal privacy to national security.

Unlike in the last century, the struggle between countries is no longer a simple matter of using advanced weapons or geographical advantage to suppress or defeat the enemy. Instead, the main force of modern warfare is information transmission and manipulation, or information warfare. In this case, it is extremely important to properly and securely transmit information between the agent and commander. For example, in highly critical communications systems, such as those used to initiate a military action, messages must have guaranteed delivery, must not be tampered with, and must be authenticated and correct. These systems must not allow for false messages, as these could lead to massive leakage of national confidentialities or even widespread loss of life in the worst case, so non-repudiation is a primary concern.

1.3 Goals and Research Questions

As stated previously, our goal in this project is to identify the best ways for a critical system to send confidential messages with the guarantee of confidentiality, integrity, and non-repudiation in a world where quantum computers are prevalent and capable. These cryptographic mechanisms may include quantum key distribution, quantum digital signatures, and quantum-resistant encryption algorithms performed on a classical computer. As a result, we have proposed six questions below which we attempt to answer in this paper:

- Research Question 1: What are the impacts of quantum computing on today's commonly used cryptographic protocols for hashing, symmetric, and asymmetric encryption? Which of these protocols or classes of protocols need to be replaced by quantum or post-quantum solutions?
- RQ2: What are the most effective quantum protocols for the creation and distribution of cryptographic secret keys?
- RQ3: What are the most promising techniques, either quantum or a hybrid of classical and quantum techniques, for maintaining the confidentiality of data at rest and in motion?
- RQ4: What are some reasonable transmission protocols that can be used for verifying the integrity of quantum data?
- RQ5: What are the best quantum cryptographic algorithms for ensuring the authenticity and non-repudiation of messages?
- RQ6: Can a standard process be followed to translate a theoretical quantum algorithm into an implementation suitable for hands-on testing?

2 RELATED WORK

2.1 Contemporary Quantum Communication and Cryptography

After reviewing some academic journals and authorized publications, we found that current analysis of quantum technology covers a wide range and has significant achievement. But not all those resources are critical, some of those might run in the opposite direction against our goal and are hard to accomplish based on contemporary technology. As a result, our team categorized several feasible resources, whose theories might be possible to be utilized in the real world, into 5 groups with respect to the project goal.

2.1.1 Quantum Key Distribution. Shor's algorithm poses a threat to current conventional cryptography and that QKD protocols have been proven to provide unconditional communication security. In some researcher analysis, they have compared results while some QKD protocols like BB84, B92, and BBM92 have been eavesdropped by a third party to check how many keys can be received and how many errors can occur during the transmission. Finally, they find if we can implement QKD in a proper way on a quantum computer, the unconditional quantum communications security can be proved[35].

The current rate of key generating QKDN is a low speed. As a result, some other studies have explored another conception, which is called Quantum Key Pool (QKP), to mitigate the inefficiency of key production by storing generated keys. But the security of QKD will decrease because the QKP needs to store keys for a while, and the basic performance of QKDN will be harmed[29].

There still exists a gap between the current QKD system and ideal one, some scientists have conquered this problem by using Decoy-State QKD scheme (DSQKD) to improve the security and performance of QKD transmission. However, in the DSQKD scheme, the pre-request is the data exchanges between two nodes are infinite. With the limited data exchanges rate of the real world QKD system, it is hard to achieve[44].

In ideal TF-QKD, the secure key rate scale can be enlarged to almost twice, so it can be used for a long-distance transmission which is much longer than the traditional QKD. In this case, TF-QKD not only maintains the confidentiality of data, but also can be used on a relevantly longer distance communication. Nevertheless, constructing the mode matching systems to finish the first interference of two types of lasers is very difficult[37].

2.1.2 Quantum Non-Repudiation. Digital signatures are an important aspect for verifying the integrity and authenticity of a message, so some researchers develop a scheme which uses a dynamic map based on quantum dots, a permutation and substitution scheme like AES, and DNA coding to create a quantum digital signature with a high degree of security as long as the signature is of sufficient length. In other words, by sending a dynamic quantum system's control parameter and critical points as well as some initial point in phase space, two parties could implement these digital signatures using a quantum computer, but without requiring a quantum channel for transmission[23].

Quantum signatures can be used for both classical and quantum messages. The ability to sign a message using only a single qubit and a trusted third party is valuable. With Arbitrated Quantum Signatures (AQS), the key forgery is impossible, but perfect non-repudiation is not[27].

It may be possible that integrity, data origin authentication, and non-repudiation can be better achieved with quantum cryptographic methods. For example, some think Quantum Message Authentication Codes (QMACs) offer an advantage over classical methods for message authentication. However, after certain studies, researchers figured the information-theoretically secure message authentication are performed better in classical cryptography, and some known QMAC schemes are inferior to their classical counterparts[34].

Few researchers identify the binary classical messages can be authenticated by a set of QMAC protocols that, using a single qubit as the authentication key, allow for the successful authentication of messages with probability of forgery less than one. This QMAC protocol also provides the possibility of key reuse, though not with guaranteed security[18].

2.1.3 Integrity and Post-Quantum Security. By using an algorithm of amplitude amplification technique, quantum collision and multi-target preimage search, it can improve attacks against hash functions, key recovery in multi-user settings, and collision attacks on block cipher operation modes. This algorithm may also be used as building blocks for more complex cryptanalysis. In addition, the presented algorithm improves the on the time complexity of existing algorithms while requiring less quantum memory. Comparisons between new and existing algorithms are made under several conditions concerning the availability of quantum memory, ultimately suggesting that this new algorithm is superior unless quantum memory becomes as cheap as classical memory and parallelization is difficult to achieve[14].

2.2 Impacts of quantum computing on classical cryptography

Quantum computing offers an exponential growth to computational power. Quantum computing's computational power is directly proportional to the size of the system. This computational growth is called computational parallelism and this growth is what makes quantum computing the potential next step in computing evolution. Quantum computing by itself does not threaten encryption or communication, but when quantum computing is combined with quantum algorithms then there is potential for threat. To help mitigate the threat that quantum computing poses to asymmetric encryption quantum key distribution was created to protect asymmetrical key distribution. Quantum computing could threaten current hashing standards and digital signatures, quantum non-repudiation was developed to help protect digital signatures. As quantum computing grows it becomes a greater threat to standard encryption, then quantum algorithms must be used to create mitigations to these threats.

Current computers figure out how to do something by trying every possible combination and picking the correct one. Quantum computers can try every combination at once, because of superposition a quantum computer can be the right path and all the wrong paths at the same time. Quantum computing uses uncertainty in its state, it both is and isn't the right path. Using entanglement you can measure the answer without collapsing the quantum state. Entanglement is two particles that are linked but physically separate. Using one set of the particles you can measure the state without collapsing the wave function.

2.2.1 Shor's Algorithm Impact on Asymmetric Cryptography. Peter Shor created a quantum algorithm, called Shor's Algorithm. Shor's Algorithm is a quantum algorithm that uses polynomial time for factoring integers. Shor's Algorithm is too resource intensive to be run by a common computer, to get the full effect of Shor's Algorithm it needs to be run on a quantum computer. Shor's Algorithm needs a high amount of quantum bits, quantum bits or qubits are a unit of quantum information. Using Shor's Algorithm which is a polynomial-time factorization problem that, with sufficient qubits, can compromise the security of RSA, elliptic curve Diffie-Hellman and most other current asymmetrical encryption. For Shor's Algorithm to threaten current asymmetrical encryption the number of qubits needed are higher than can be currently created. That does not mean that asymmetric encryption is safe[42].

To secure classical computing against the future potential of quantum computing, Shor's Algorithm researchers have created lattice and ring based encryption. Lattice and ring based encryption are based on mathematical algorithms that have been studied since the 1980 and no known attack has worked on them. Quantum computing can use quantum key distribution to secure its data in motion. quantum key distribution using BB84, Bennett and Brassard proposed this in 1984. BB84 can use multiplexing and is good out to 200 km, without multiplexing it can go 240 km. Tools are not fully developed yet. Protocol E91 can use multiplexing and can transmit out to 200 km, 240 km without multiplexing. Protocol E91 Developed in 1991 by Arthur Ekert, using Protocol E91 the attackers can't guess results. Protocol E91 is too resource intensive. MDI-QKD has the best range of 404 km, but it needs specialized configuration on the transmission channel.

Asymmetric encryption is used to secure current communication systems and currently is a safe way to send data over the Internet. This can all change with Quantum Computing so to secure future communication Quantum Key Distribution (QKD) is being developed to help secure communication. Quantum Key Distribution is a One-Time-Pad encryption "As we all known, One-Time-Pad is the most secure way to build communication between two network nodes, so the Quantum Key Distribution (QKD) is taking advantage from it to build a much safer network environment called QKDN"[29].

2.2.2 Grover's Algorithm Impact on Symmetric Cryptography. The quantum search algorithm proposed by Grover is a powerful algorithm of quantum computing, which is suitable for solving the following problem: to find a specific object from N unclassified objects. More specifically, the classical algorithm can only search one after another until it finds the object it wants. This algorithm has

$O(N)$ complexity on average, whereas Grover's quantum algorithm only has $O(\sqrt{N})$ complexity on average[24].

For example, Grover's algorithm can reduce the time required for brute force attacks. For public key encryption algorithms such as AES and 3DES, a quantum computer can reduce the security of keys by a factor $O(\sqrt{n})$, rendering the problem of brute forcing a 256-bit key equivalent to brute forcing a 128-bit key with a traditional computer. The algorithm proposed by Grover reduces the time of collision attack and reduces the security strength of hash function. With the quantum computer, the security strength of SHA256 was also reduced from 128-bits to 80-bits or less, and the security strength of SHA384 was reduced from 192-bits to 128-bits[22].

In 2019, Google used a 53-qubit quantum computer to prove that quantum computing systems have some special capabilities that can beat traditional computers (solving a problem that would take supercomputers 1,000 years to solve at 2.30 minutes), despite IBM's dissenting opinion that it would take only two days instead of 1,000 years. But it has essentially shown that quantum computers do outperform traditional supercomputers on specific problems that will take humanity to new horizons never explored before[9].

2.3 Attacks against quantum cryptography techniques

2.3.1 Photon Splitting Attack. In the ideal BB84 protocol, an important assumption is that Alice uses a single photon source. However, it is difficult to prepare a single photon source in the actual system, and a weakly coherent light source is usually used, which can be obtained by attenuating the laser light source. The photon number distribution of weakly coherent light source obeys the Poisson distribution, and there is a non-negligible multi-photon component in it. For multi-photon components, Eve can eavesdrop using photon-number splitting (PNS) attacks.

The basic principle of PNS attack is as follows: Eve intercepts the weak coherent pulse sent by Alice to Bob and obtains the photon number information through quantum non-destructive measurement. For the part of the single-photon state, all interceptions are no longer sent to Bob; for the multi-photon part, Eve extracts one photon from it and stores it in its own quantum memory and sends the remaining photons to Bob through a low loss or even no loss channel (ideally). After Bob publishes his measurement basis vector, Eve measures the photons stored in his quantum memory under the same basis vector. Then, combining with the basis vector information published by Alice, Eve carries out the same data post-processing process as Bob, so that Eve can obtain the exact same key as Bob[13].

2.3.2 Denial of Service. According to the NSA's report in 2020, they declared "Quantum key distribution increases the risk of denial of service. The sensitivity to an eavesdropper as the theoretical basis for QKD security claims also shows that denial of service is a significant risk for QKD"[33].

Exactly, as the classical communication network, the quantum network is still vulnerable to denial of service attack. Briefly speaking, a DoS attack, or denial of service attack, refers to hackers trying to compromise the target machine or server to make it stop functioning, which is one of the common attack methods of hackers. To accomplish DoS attack, hackers generally send numerous malicious requests through the network to overuse the target resources until it has crashed, so other lawful users cannot correctly access these resources at the moment. In common, DoS can usually result in a huge financial loss regarding different areas like governments and enterprises[2].

2.3.3 Man in the Middle Attack. QKD generates the necessary keys for the encryption algorithm to ensure the privacy of the communication, but QKD itself cannot provide an authentication mechanism to the source of the transmission. The security of QKD is weakened because it cannot prevent a man in the middle (MitM) attack from its own technical perspective. The NSA report raises serious questions about this, which cannot be avoided and cannot be left undiscussed[33].

We could seek out this scenario from the quantum transmission layer. The sender randomly selects a polarization of 0° and 45°, denoted as + and × respectively, for each qubit sent, and the receiver randomly selects one of + and × for each qubit received. The receiver then tells the sender its sending bases over an insecure channel, such as the Internet, and the sender indicates which parts of it are correct. In this case, the sender and receiver will ignore those qubits whose listeners are set incorrectly. The sender and receiver then compare half of the remaining qubits, and if there is an error indicating that there is an additional listener in addition to the receiver, whose presence interferes with the photon's vibrational direction. If there is no error, the bits are discarded and the remaining bits are used as the key. If an eavesdropper is present, he will cause the last check operation to fail, because he will change the state of the original photon and cause the qubits to turn out to be the wrong answer half the time.

Now, we could consider the MitM attacks scenario. The initiator of a MitM attack is a more powerful actor than the listener, who not only has access to the entire Internet communication packets between the two parties but can also modify those packets as whatever he wants. Thus, he can present himself as the receiver to the sender and present himself as the sender with respect to the receiver side.

Once the attack is started, the middleman randomly selects one of the + and × bases at the beginning of the communication, tampers with the message so that the sender receives the receiver's bases that is exactly same with middleman's bases, and tampers the message so that the receiver receives the correct pattern that is the qubits for which the middleman and the receiver have the same bases. In the final verification process, it is obvious that the sender and the middleman retain the same bits, and the middleman also knows the bits that the receiver retains[20].

2.4 Cryptography Survey Methodology

Two other groups have conducted similar surveys of numerous cryptographic protocols. In their 1997 paper, Jorstad and Smith[25] attempt to answer the question "Can a standard objective framework for the measurement and specification of cryptographic algorithm strength be created?" Their methodology relied heavily on known characteristics of existing algorithms and the ways in which they might be compared. They focused almost entirely on civilian encryption algorithms, both symmetric and asymmetric, meant for use in commercial products and which were operating in Electronic

Code Book (ECB) mode. However, their work is somewhat dated due to the selection of algorithms, only including those that existed in 1997, but their proposed classification scheme is useful. Their proposed classification scheme covers seven categories, including type (symmetric or asymmetric), functions (secrecy, integrity, etc.), key size, rounds, complexity, attack, and strength.

A similar study was conducted by Khan et al.[28] in 2020 which focused entirely on QKD protocols. Most contemporary QKD protocols have not been compared in depth with regard to security; however, especially with regards to simulation and implementation, it is necessary to verify the deviation between the theoretical aspects and real-world usage. The authors of the paper provide a simple quantitative comparison of 11 different QKD schemes across six different factors, as well as a simulated analysis of the BB84 and 2-dimensional KBM09 protocols. Based on their result, even though their experimental comparison is limited to only evaluating the Quantum Bit Error Rate (QBER) reliability of two protocols, it still shows useful guidance on the ways protocols can be compared both theoretically and experimentally and serves as a basic approach for the classification of the quantum cryptography schemes.

2.5 Post-quantum cryptography

The National Institute of Standards and Technology (NIST) is leading a project called Post Quantum Standardization (PQS), which aims to define new algorithms that can address quantum computer threats. The PQS project is now in its final stage and is expected to be completed within two years[32].

In order to realize the transition to quantum secure computing, SSH, VPN, IPSec, SSL/TLS and other security protocols also need to be upgraded. These protocols need to be combined with existing protocols, but also need to introduce an additional layer to establish secure communication to protect against quantum attacks. This change will have an impact on asymmetric encryption and key generation algorithms, and it is necessary to increase the key size of symmetric cryptography algorithms. There is also an impact on performance and bandwidth. Hardware vendors will also need to upgrade their hardware to align and transition with these new algorithms.

Additionally, we need to promise those new algorithms must not fall into bounded-error quantum probabilistic polynomial (BQP) complexity class. The BQP class can be traced back to 1993, when computer scientists Ethan Bernstein and Umesh Vazirani defined a new class of complexity they called BQP for "bounded error quantum polynomial time"[12]. They define this category as all the decision problems that a quantum computer can effectively solve – problems where the answer is yes or no. In other words, for a BQP problem, there is a quantum algorithm that takes polynomial time to run and has a high probability of getting the right answer. For any given situation, the chance of getting the wrong answer should be less than 1/3. BQP can also be regarded as the quantum computer version of BPP, which stands for bounded-error, probabilistic, and polynomial time[43].

2.6 Types of Quantum Computing

Adiabatic quantum algorithms are used to optimize the Hamiltonians, a function that represents all energy in a system. The Hamiltonian uses an operator to correspond to the energy of a system. The Hamiltonian corresponded to the total energy in both kinetic and potential energy. Quantum adiabatic algorithm (QAA) was designed to solve the optimization issues in quantum computing. QAA in a dedicated device will optimize the combinatorial optimization problem. The combinatorial optimization problem is solved by evolving adiabatically, only energy is transferred, when in the ground state. Adiabatic quantum computing can be as powerful as a non-stoquastic Hamiltonians in the circuit model. This means that the eigenvalue gap is at its minimum complicated many-body Hamiltonian. QAAs allow for quantum speedup and can overcome some of the issues in qubit quantum computing, such as needing less qubits to crack asymmetric encryption. The issue with adiabatic quantum computing is that it is inherently unstable as the qubits get higher, this creates a greater limitation to its future use. QAAs are great for solving satisfiability problems and combinatorial search[4].

The circuit model of quantum computing is the most common form of Quantum Computing currently. Circuit model quantum computing uses Hilbert space, a vector space that allows defining lengths and angles, and has a series of unitary quantum logic gates to manipulate its qubits. Multiple quantum logic gates can be connected together to generate more complex quantum operations. A big difference between classical logic gates and quantum logic gates are that quantum logic gates are able to go backward and forwards, where classical logic gates can only move forwards. Despite the unique nature of quantum gates, it is still possible to implement a full boolean algebra and even a full quantum Turing Machine using quantum gates[39].

3 METHODOLOGY

Our research methodology closely resembles that used by Khan, Meraj, and Khan in their 2020 comparison of QKD protocols[28]. This research primarily involved performing a comparative analysis of quantum cryptography protocols. A number of algorithms representative of different aspects of cryptography, including key distribution, confidentiality, integrity, and non-repudiation, were selected from the literature. We examined these protocols and identified important characteristics which we felt could best be used to differentiate them from one another.

Algorithms were classified using unique criteria based on their intended function, as categorizing an encryption algorithm based on its verifiability would not make sense. Some algorithms which provided multiple functions, such as confidentiality and non-repudiation in the case of [5], appear in multiple categories. This examination was conducted with the goal of answering our research questions and selecting algorithm candidates for implementation in a quantum simulator.

3.1 Quantum Key Distribution

The QKD stands for the Quantum Key Distribution, which is defined as transferring a secure key between two users by using quantum related technology[28]. Our original goal for this project is to seek out a proper and efficient Quantum Key Distribution protocol to satisfy both performance and security during the key transmission between two network nodes or even more users. Since the real implementation of QKD is different from the theoretical way, we would like to do some research with respect to make QKD practical. In the below list, we have provided a general view of those different QKD protocols that either are under analyzing by other researchers or have been experimented using real quantum related devices.

The first QKD protocol is called BB84, and it was proposed in 1984 by Bennett and Brassard – that's where the name comes from. The idea is to encode every bit of the secret key into the polarization state of a single photon. Because the polarization state of a single photon cannot be measured without destroying this photon, this information will not be available to the eavesdropper without revealing himself or resending the photon. Here, the pros of the BB84 are obvious. It can provide a reasonable security level because Eve's measurement will change the status of original qubits.

Regarding the cons of BB84, we realize that the transferring message will be affected while there is an eavesdropper, but it is still an implementable protocol by using current infrastructure. Moreover, the basic procedures during the key transmission can be separated to six steps:

(1) Alice transmits a random sequence of 0s and 1s qubit, alternating the bases × and + randomly.
(2) Bob receives the qubit sequence from Alice and randomly alternates the measures between bases × and +.
(3) Alice broadcasts the succession of bases used in a public channel.
(4) Bob reports to Alice in what cases he was able to guess the origin bases.
(5) They both select a part of the result to compare to see if the error rate is above or under the requirement.
(6) By using the bits of two match identical bases, they both have defined a random succession of bits that will do as OTP for transmission.

Besides BB84, there are two similar protocols that have the same key transmission length. The first one is E91, and the other is B92. More specifically, the B92 is a simplified version of BB84 because it just uses 2 polarization status (0 degrees and 45 degrees) while BB84 utilizes 4 (0, 45, 90, 135 degrees); but the security level of B92 is less than BB84. On the other hand, the E91 is the first protocol to use opposite measures responses between two parties, and its design is based on quantum entanglement. Even though its security level is little bit higher than BB84, it is too many resources consuming which includes system, hardware, and supplement etc.

Moreover, we have analyzed other two QKD protocols that are considered as the most secure protocols, and have long-distance qubits transmission rate:

3.1.1 *Measurement Device Independent QKD.*

In Measurement Device Independent QKD (MDI-QKD), transmission distance is 404 km, and it is known as a simple solution which can remove all (existing and yet to be discovered) detector side channels, especially during the implementation phase. Theoretically, it shows that it has both excellent security and performance. To be honest, it is a best choice to implement compared to the other QKD protocols, and its transmitting rate is reasonable, but it is hard to configure by using current network transmitting channels.

3.1.2 *Twin-Field QKD.*

Twin-Field QKD (TF-QKD) has the longest photon transmission rate: 509 km. Dual-field quantum key distribution is a measurement device independent quantum key distribution that uses the detection after single photon interference as an effective detection event. It requires only a single detector response and does not require the traditional measurement equipment independent quantum key distribution of two photons to meet the required simultaneous response of two detectors. In other words, the information in TF-QKD is encrypted in the process of the optical pulses initiated by the two users who want to establish the secure communication. Then, the secret key is generated by a single photon interference measurement from a user in the middle. It sounds like a feasible choice, but it requires too much controlling systems like phase-lock maintaining system, timing control system, and polarization system etc.

By the way, there still exists many QKD protocols we have not explored yet like Differential Phase-Shift (DPS) Protocol which can prevent PNS attack in some ideal photon source condition; SARG04 Protocol is increasing the secure level against PNS attack because its reconciliation phase; Coherent One Way (COW) Protocol is designed for countering the challenge of single photon sources; KMB09 Protocol is a more advanced protocol which can improve the detection rate while an eavesdropper appears. Finally, S13 and LZWW16 protocols are the improved or optimized version of BB84. Thus, we know the BB84 is a very important QKD protocol to analyze and implement because once we achieve it, it can provide a shortcut for us to explore the other protocols that are designed based on it[28].

3.2 Confidentiality

Criteria for the classification and evaluation of quantum cryptosystems have not been formally defined by research institutions like NIST's CRSC. We determined the following criteria to be appropriate for our classification and evaluation of selected quantum cryptosystems with respect to their ability to ensure confidentiality:

- Target data
- Key characteristics and reusability
- Resource requirements/limitations

The data that a cryptosystem works to secure is an important consideration in evaluating its candidacy for real-world implementation. As with classical cryptosystems, the characteristics and reusability of keys used in quantum cryptosystems can serve as a primary differentiator of comparable systems and as an indicator of potential system weaknesses and vulnerabilities. Notable resource requirements or limitations for a cryptosystem's implementation are also important in considering a system's viability in real-world applications.

To date, few quantum algorithms have been designed that are focused on ensuring confidentiality during the transmission of quantum or classical data. However, despite the lack of work regarding such quantum cryptosystems, a few proposed systems stand out as candidates for proof-of-concept implementation or to serve as a base for further research. Kak's Three-Stage Protocol

[26] is one of few entirely quantum solutions for data encryption through a public channel that can also serve as a simple method for key exchange. A scheme proposed by Amerimehr and Dehkordi [5] which - in addition to providing integrity and non-repudiation - ensures confidentiality during the transmission of classical data without the use of a public channel. Another scheme, proposed by Pleşa [38], achieves confidentiality by way of a multi-channeled, hybrid system that uses a quantum teleportation circuit for key exchange and a classical channel for data transmission. This hybrid system demonstrates the feasibility of near-term implementation of quantum cryptosystems that integrate with existing classical infrastructure. A summary of these algorithms and their important characteristics can be seen in Table 1.

Technical challenges regarding the ability to store qubits have limited the research and development of algorithms for encrypting quantum data at rest. Advancements in this field appear to be few and far between; even in perfect conditions, the longest period of time that qubits can be stored before decoherence is between three and six hours [40, 45].

3.3 Integrity

Integrity in classical computing deals with making sure that the data is real, accurate and secure. Making sure that the data is real and accurate can be an issue with quantum computing due to the inherent nature of quantum. For integrity of Classical computing against the future threat of quantum computing, AES for data at rest. When data is in motion lattice based encryption, NIST is still doing testing, has shown the best resistance to quantum computing. NIST is still testing different hashing algorithms. For integrity of quantum computing Zero knowledge will tell you if data has been tampered with. The nature of quantum can increase integrity, when in superposition data can't be tampered with or the wave function collapse. For data in motion QKD paired with quantum authentication encryption using a pre-shared secret key will maintain quantum computing integrity. These choices have been vetted by multiple outside research.

It is hard to ensure integrity when the answer is neither right nor wrong until measured and analyzed. "This process is known as quantum parallelism. However, measuring the output states will randomly yield only one of the values in the superposition, and at the same time destroy all of the other results of the computation."[39] The nature of quantum can also increase data integrity through the wave function, if the data is tampered with it will change the superposition and collapse the wave function. The data will be safe from outside interference until its quantum entanglement is measured. "Einstein, Podolsky, and Rosen proposed that each particle has some internal state that completely determines what the result of any given measurement will be."[39] Integrity of data already retrieved from the quantum computer can be ensured. Data at rest can be tested with Zero Knowledge and data in motion can be secured with QKD paired with quantum authenticated encryption.

NTRUEncrypt is a lattice-based public-key Asymmetric encryption. NTRUEncrypt was first introduced in 1996 and is one of the most researched lattice-based encryptions designed to protect against quantum computing. "We choose implementations offering 128-bit security (except RSA-1024 offering 80-bit security) for comparison. Our AvrNTRU outperforms the RSA implementation, achieving 82.8 times faster decryption, even though RSA-1024 cannot match the same security level."[16]. NTRUEncrypt is the precursor to NTRUPrime. NTRUPrime is a faster updated version of NTRUEncrypt, and is currently in trials with the U.S. Government for possible U.S. Government endorsement for hardening classical computing system against quantum computing. While quantum computing cannot produce enough qubits to threaten today's encryptionschemes, van Oorschot-Wiener's classical computing parallel collision searching algorithms is more of a threat to integrity than quantum computing currently[31].

3.4 Non-Repudiation

Amiri and Andersson[6] conducted an in-depth review of unconditionally secure quantum signatures in 2015, on which we heavily relied in setting the direction of our research into quantum non-repudiation. In their work, Amiri and Andersson describe signature schemes, whether for classical or quantum data, as having three goals: unforgeability, non-repudiation, and transferability. It should be impossible for an adversary to send a signed message impersonating a legitimate party, and impossible for a legitimate sender to deny a signed message. Furthermore, when one party verifies or rejects a signature they should be confident that any other party would verify or reject in the same way.

We chose to categorize the quantum signature algorithms we examined first according to whether they target classical or quantum data, as this is indicative of a major difference in their usage. Next, we examined whether the signature produced was reusable, i.e. could be verified by more than one party without destroying the data. We also categorized the algorithms on the underlying security principle behind the quantum signature, as well as the role of a third party arbitrator in the protocol, if one was present at all.

Our selection of algorithms ranges from the original quantum signature scheme as proposed by Gottesman and Chuang in 2001[21] to recent research conducted by Hematpour, Ahadpour, and Behnia involving the dynamics of quantum dots[23], and covers a variety approaches to quantum non-repudiation. A summary of our reviewed protocols can be seen in Table 2.

Common to most early schemes for quantum signatures[11, 18, 21] is the lack of a trusted third party, without which it is impossible to produce unconditionally secure signatures of quantum messages[27]. In more recently developed quantum signature schemes, a trusted third party either generates private keys[15] or provides non-repudiation[23, 27]. The only recent protocol we found which provides non-repudiation without the use of a trusted third party was proposed by Amerimehr and Dehkordi, and also provides confidentiality and integrity through the use of classical encryption and keyed hashing in combination with transmission over a quantum channel[5], though this protocol is not capable of signing quantum data.

4 ANALYSIS & RESULTS

4.1 Implementation of a Quantum Cryptosystem

As a primary goal of this research was to identify quantum cryptographic algorithms with practical applications, we felt it necessary

Table 1: Classification of Quantum Confidentiality Algorithms

Algorithm	Target Data	Key Characteristics	Resource Requirements/Limitations
Kak's Three-Stage 2006	quantum	single-qubit	quantum channel, single-qubit data
Amerimehr and Dehkordi 2018	classical	length equal to message length, reusable	limited message length, algebraic ECC
Pleşa 2017	classical	single-qubit	multi-channel; shared, entangled qubits

Table 2: Classification of Quantum Non-Repudiation Algorithms

Algorithm	Target Data	Reusable?	Security Principle	Third Party
Gottesman and Chuang 2001	Classical	Maybe	Quantum trapdoor function	None
Curty and Santos 2001	Classical	No	Entanglement	None
Barnum et al. 2002	Quantum	No	Purity testing codes	None
Kang et al. 2015	Quantum	No	Quantum trapdoor function	Provides non-repudiation
Amiri et al. 2016	Classical	Yes	Quantum key distribution	Provides non-repudiation
Chen et al. 2017	Quantum	No	One-Time pad	Arbitrator generates private keys
Amerimehr and Dehkordi 2018	Classical	Yes	Classical encryption and HMAC	None
Hematpour et al. 2020	Classical	No	Unique system state and critical points	Provides non-repudiation

to implement a theoretical algorithm described in a research paper using a quantum programming framework.Qiskit was selected as our implementation tool of choice due to team member familiarity, its overall ease of use, quality of documentation, and simplicity of running against IBM's cloud-connected quantum hardware. The algorithm we identified as the most promising candidate for implementation was proposed by Amerimehr and Dehkordi in their 2018 paper "Quantum Symmetric Cryptosystem Based on Algebraic Codes"[5], as the proposed system is simple and provides confidentiality, integrity, and non-repudiation. For simplicity we shall follow the naming convention used by BB84, B92, and others, and refer to this cryptosystem as AD18.

The algorithm itself resembles BB84 in that the transmitting party, Alice, selects random bases for the transmission of her message. While the recipient, Bob, measures the received qubits in a random basis in BB84, in AD18 Bob measures all of the received message values at a 22.5° angle and encounters an approximately 15% error rate in his measurements. The actual message sent by Alice includes an algebraic error correcting code of sufficient quality to correct the errors Bob encounters, thus allowing for a successful transmission without the public announcement of bases. Interspersed with the message qubits is a keyed hash value, and a pre-shared key is used in conjunction with some functions $f(k)$ and $g(k)$ to determine the locations and bases used in the transmission of the hash qubits.

The AD18 cryptosystem can be thought of in terms of quantum and classical components. The quantum piece simply involves preparing, transmitting, and measuring qubits. The classical component of the algorithm involves encryption, keyed hashing, and error correction. For the quantum component of the algorithm, we were able to leverage the BB84 code example provided in the Qiskit documentation[1] to demonstrate the preparation and measurement of qubits in different bases. This was accomplished in BB84 by applying a combination of negation (X) and Hadamard (H) gates during preparation, and H gates during measurement. In AD18, as Bob is measuring in a 22.5° basis, we applied an R_Z gate with $\theta = -\pi/8$.

The classical portions of the algorithm involving encryption and hashing did not appear to be as important to the overall concept as qubit measurement and error correction. For ease of implementation, we selected a Salsa20 stream cipher with a 128-bit key for the encryption and decryption of the message, and an HMAC-MD5 keyed hash due to its short length. These protocols could easily be replaced with alternatives without materially impacting the functionality of the algorithm, though changes would affect the total number of bits sent and the overall security of the protocol. For our $f(k)$ function, we used a trivial implementation in which the message and hash are simply concatenated together, though this would need to be modified for a usable implementation in order to maintain the security of the system. The $g(k)$ function used to determine the transmission bases of the keyed hash was also kept simple, with Alice transmitting hash bit H_i in B_Z if $k_{(i \mod len(k))} = 0$ and in B_X otherwise.

The primary difficulty in successfully implementing this algorithm arose in the identification of an algebraic error correcting code (ECC) that was capable of handling the observed error rates for non-trivial messages. In the paper describing AD18, the authors transmit a two bit message which is expanded to five bits by their $[5, 2, 3]$ linear error correcting code. As we wished to transmit the significantly longer message "hello world", we selected the Python library `commpy` and used its convolutional coding functions. A transmission success rate of $> 90\%$ on a three character message was achieved by using a memory size of 3 and a G-Matrix initialized with the value `array([[1, 3, 5, 7, 9, 11, 1, 3, 5, 7, 9, 11]])`. Simpler convolutional code trellises resulted in dramatically worse performance of the cryptosystem, with transmission success rates $< 25\%$ per byte for a value `array([[5,7]])`. Additionally, the G-Matrix used in our implementation would likely not be practical in a real implementation as it adds 100 bits of error correction per byte transmitted, which may not be feasible for use given the bitrates of current quantum channels.

A larger concern that is not specific to our implementation is that the algorithm does not provide a high level of consistency for the transmission of longer messages. An analysis of the algorithm's performance over 1000 executions for string lengths ranging from 1 to 8 using the above parameters shows a success rate of approximately 0.96 per character transmitted, so for an 8 character message we expect and observe a success rate of $0.96^8 = 0.72$. While it may be possible to tune the error correction algorithms further to increase performance, the probabilistic nature of the quantum measurements combined with a hash check against the whole message means that the failure to correctly decode even a single bit of data using ECC will result in a significantly different hash value and thus a failure to authenticate the message.

While the authors are not experts in the field of error correcting codes and may have used a suboptimal mechanism in our implementation, we still find the proposed system to be impractical for non-trivial examples. When Bob makes his measurements in this cryptosystem, every bit measured has a 15% chance to be measured incorrectly. As a result, there is always a non-zero probability of a decoding error regardless of the ECC used, and the overall error rate rapidly approaches a prohibitive level. For a trivial example with a two bit message and three bits of error correction as presented in the paper, the chance of Bob correctly measuring 4 or more bits as required for error correction is $0.85^5 + 5 * (0.15 \cdot 0.85^4) = 84\%$. If we use the same error correction mechanism and expand to 8 message bits with 12 bits of ECC, then the chance of correctly measuring a sufficient number of bits to decode the message falls to 50%, and is cut in half for each subsequent 20-bit block of error corrected data we append.

4.2 General Process For Quantum Cryptographic Algorithm Implementation

As the act of implementing a proposed quantum algorithm can prove invaluable in uncovering its weaknesses, we believe that sharing and generalizing the methodology used in our implementation can benefit other researchers seeking to answer similar questions regarding other quantum cryptographic protocols. Our implementation was done in Qiskit, but this approach is language agnostic within the circuit-based quantum computing paradigm and is expected to apply to Cirq, Q#, AWS Braket, and other quantum programming languages. The approach presented here has the additional limitation of not representing the transmission of quantum data, though this is largely a limitation of the current state of quantum programming.

The quantum cryptographic algorithms we reviewed are generally made up of the following four stages and their corresponding activities:

1. Preparation: Pre-shared secrets established, channels selected, and values initialized as qubits.
2. Sender Processing: Cryptographic operations performed on message qubits.
3. Transmission: Data transmitted over selected channel, public values announced.
4. Recipient Processing: Cryptographic operations performed on message qubits, data validated, and eavesdropper detection performed.

Steps may be repeated or reordered, and roles may change, as may be observed in multi-party signature schemes where data is passed back and forth between sender, recipient, and a trusted third party, with various operations being applied along the way.

4.2.1 Preparation. In this stage of a cryptographic protocol, the sender and receiver establish a message channel, establish which pre-shared secret values to use, and prepare any necessary data. Should any classical cryptography, such as encryption or hashing, be required in the protocol, we recommend applying it at this stage whenever possible. Data preparation involving classical data can be done by converting into binary, then applying an X gate to qubits which are meant to represent 1s in the binary data. As classical data can be large, it can quickly push simulators to their limits when trying to perform even simple operations on a 32-bit or larger classical value. To avoid this pitfall, consider that it is typically not necessary to construct a complicated circuit in which all values are processed in parallel, unless all of the qubits interact in some manner. Instead, consider representing the interaction as an array of simpler circuits. For quantum data, in this stage the sender would apply appropriate operations in order to set the qubits into the appropriate state, such as executing Grover's algorithm up to the final measurement stage prior to hypothetically signing and transmitting the results of this algorithm. We also recommend looking for places where initial preparation can be simplified without materially impacting the functionality of the protocol under study.

When implemented, the input to this stage should be classical data in the form of a string, byte array, bit array, or similar, and the output should be a quantum circuit or array of quantum circuits.

4.2.2 Sender Processing. This stage is the least specific as it will vary the most widely from protocol to protocol. The most common operations we observed in this stage were simple rotations, which are typically applied either as the Hadamard (H) gate or rotation ($R_{\{x,y,z\}}$) gates of arbitrary value. As the names of gates in quantum frameworks are generally well documented and only differ mildly amongst the various languages, referring to API documentation at this stage will resolve many challenges. Protocols which require custom unitary gate operations are also supported by many frameworks, such as Qiskit's `UnitaryGate` class.

When implemented, this stage will take as its input the quantum circuit that was produced in the preparation stage, and the output should be a quantum circuit or array of quantum circuits with additional operations applied.

4.2.3 Transmission. The quantum computing frameworks we have used do not provide the capability to easily represent the transmission of quantum data from one party to another. If noise or an eavesdropper are to be present, they must be represented with additional gates and measurements at this stage. In a typical eavesdropper scenario, Eve makes measurements using any available public information, then would retransmit the data to Bob. In some cases Eve may apply her own set of gates before retransmitting the data.

This stage can be omitted in most cases and simply represented as the passing of the quantum circuit from the previous stage to the recipient processing stage. If an eavesdropper, noise, or other events which impact transmission are to occur in the simulation, then the stage should accept and return a quantum circuit.

4.2.4 *Recipient Processing.*

This stage can be quite protocol dependent as well, though it will frequently involve the application of one or more gates prior to taking a final measurement of the received qubits. After measurement, additional public sharing or comparison of values may take place. This public sharing of values is limited by the same lack of capability described in the transmission stage, but is usually easily represented by passing parameters into a function. For signature or hashing schemes, this is where a final validation of the signature or hash will occur, either by computing a classical keyed hash or by performing additional quantum operations involving a third-party arbitrator.

The implementation of this stage should take a quantum circuit to represent the data received by the recipient, and return either a classical value as in the case of encryption and decryption, or a boolean value indicating the success of the signature or hash validation.

4.2.5 *Workflow.*

Our proposed workflow follows an iterative model of development, similar to that seen in test-driven development and agile methodologies. After identifying the desired algorithm for implementation, and validating that any quantum operations are supported by the chosen framework, the first step in this approach is to categorize the steps of the protocol into the previously described stages. In our implementations, we found it helpful to create a primary method to represent the full protocol, then create empty methods named for each stage, such as `alice_prepare_message` or `bob_decrypt_message`. For more complicated protocols, there will likely be multiple methods created for some of the stages. Before adding code to each method, it may be useful to describe the specific actions of the protocol in comments, and to identify which require classical data manipulation and which require quantum operations.

The second step of our workflow is to identify and initialize any values external to the algorithm under study, such as pre-shared secret keys or configuration of the quantum simulator. These should typically be defined as shared values outside the scope of any methods, as this makes them easier to locate and change and lowers the complexity of method signatures. We also highly recommend either using a standard logging library, or adding a boolean value `debug = true` so that helpful messages throughout the code can be conditionally enabled or disabled.

The next step in the workflow is to begin implementing the simplest possible version of the protocol by only coding the inputs and outputs of each stage. Use a small input of all zeros or a single letter string, create a quantum circuit with no gates or an identity gate for the transmission, only apply a measurement operation at the recipient, and perform any validation by simply returning `true` or `false`. This allows for easy verification that data is able to flow from end to end through the algorithm without introducing extra complexity. Judicious use of debugging statements at this stage can make later troubleshooting significantly simpler.

The fourth step should be to implement any classical operations used by the protocol, such as classical encryption, using appropriate libraries where possible. Validate that these operations function as expected outside of the quantum protocol under study, and add any relevant debugging statements.

In the final step, we implement the quantum portion of the algorithm within the structure we have created. Apply appropriate gate operations to the empty circuits that were constructed in the third step of the workflow, then test and verify the circuit on a local simulator. Once the protocol is functioning as expected, perform any desired refactoring activities. Creating a test harness which performs a repeated execution of the protocol with a variety of inputs, and validates the outputs, is highly recommended for any subsequent analysis.

4.3 Example Using Kak's Three-Stage Protocol

Using the workflow described above, we present a simple implementation of Kak's Three-Stage Protocol[26] for encryption of a single bit message. First, we categorize the steps of the protocol as follows:

- Preparation: Alice selects a value x to transmit. Alice and Bob each establish their own secret key.
- Sender processing 1: Alice applies $U_A = R(\theta)$ to her qubit.
- Transmission 1: Alice transmits to Bob
- Recipient processing 1: Bob applies $U_B = R(\phi)$ to the received qubit.
- Transmission 2: Bob transmits back to Alice
- Sender processing 2: Alice applies $U_A^\dagger$ to the qubit.
- Transmission 3: Alice transmits back to Bob
- Recipient processing 2: Bob applies $U_B^\dagger$ and has now decrypted the message.

As we are not planning to simulate an eavesdropper or noise in this protocol, all three transmission steps will not require any code and can simply be represented by comments and values being passed between methods. The stub of our Kak's protocol implementation after the completion of the first step of our implementation methodology can be seen in Listing 1.

Listing 1: Implementation of Kak's Three-Stage Protocol, Step 1

```
def kak_3_stage(message):
    prepared = alice_prepare_message(message)
    tr_1 = alice_apply_rotation(prepared)
    tr_2 = bob_apply_rotation(tr_1)
    tr_3 = alice_remove_rotation(tr_2)
    decrypted = bob_remove_rotation(tr_3)
    return decrypted
```

In the second step of our methodology, we can see that Kak's protocol uses two secret keys, which we can initialize as desired. As the secret keys are used to generate the θ and ϕ values used by Alice and Bob, we choose to generate a random value for each between 0 and 2π in order to keep the implementation simple. As seen in Listing 2, we also create a `debug` flag and define `backend = qasm_simulator` for use later.

Listing 2: Implementation, Step 2

```
debug = True
backend = 'qasm_simulator'
alice_key = np.random.uniform(0, 2*pi)
```

```
bob_key = np.random.uniform(0, 2*pi)
if debug:
    print("Alice's␣key:␣%s" % alice_key)
    print("Bob's␣key:␣%s" % bob_key)
```

Next, we can implement the simplest version of our methods in which Alice and Bob simply apply an *I* gate at each step, and Bob performs a measurement in the final step. While most of these methods are trivial, the final one in which Bob performs his measurement will contain the code necessary to execute the quantum circuit on a simulator. The method for Bob's final rotation removal and measurement appears in Listing 3.

Listing 3: Implementation, Step 3 (Partial)

```
def bob_remove_rotation(qc):
    qc.i(0)
    qc.measure(0,0)
    if debug: print(qc)
    # Execute in simulator
    qasm_sim = Aer.get_backend(backend)
    qobj = assemble(qc, shots=1, memory=True)
    result = qasm_sim.run(qobj).result()
    return int(result.get_memory()[0])
```

As the protocol contains no classical functions, we move to the final step where we properly prepare Alice's message and implement rotation operators to replace the identity gates we used in the previous step. As Qiskit provides an R_Z gate capable of performing an arbitrary rotation, we can use Alice and Bob's secret key values directly in this gate to apply their rotations. In the code, this simply involves replacing `qc.i(0)` with `qc.rz(alice_key, 0)`, as seen in Listing 4. After completing the implementation, we can verify that the operation works correctly by checking `kak_3_stage(0) == 0` and `kak_3_stage(1) == 1`. From this starting point, it would be possible to refactor the protocol to take a longer input, use a more complicated key generation method, or include an eavesdropper.

Listing 4: Implementation, Step 5 (Partial)

```
def alice_remove_rotation(qc):
    qc.rz(-alice_key, 0)
    if debug: print(qc)
    return qc
```

4.4 Discussion

4.4.1 Confidentiality. While quantum computing's contributions to confidential communications are most immediately going to come by way of Quantum Key Distribution paired with classical cryptography, it is clear that it could play a role in communication itself as the capabilities of quantum computers and networks improve.

Kak's Three-Stage Protocol is a foundational demonstration of entirely quantum encrypted communications. The protocol is capable of perfect security when accompanied by a classical protocol to ensure identity of communicating parties and a mechanism for error-checking/-correction[26]. Kak's Three-Stage Protocol has been extended by further research that address its shortcomings, including implementations that enable multi-photon transmissions[30] and the correction of errors that occur over a quantum channel[36]. While these advances bring the realization of an entirely quantum cryptosystem closer to near-term implementation in real-world applications, such solutions are limited by the capacity of circuit-based quantum processors which, at the time of writing, operate with less than 100 qubits[3, 9].

The scheme proposed by Amerimehr and Dehkordi ensures confidentiality by using a single pre-shared encryption key, which is an advantage over similar schemes that rely on multiple secret keys as a single key reduces pre-processing overhead[5]. In addition to providing a performance benefit, this pre-shared key can be reused securely, which is a desirable characteristic of a quantum cryptosystem that enhances its real-world viability. While this system's theoretical capabilities are promising for real-world implementation, its reliability quickly degrades as message length increases, as demonstrated in 4.1. This is, again, largely due to the limited capacity of quantum computers and suggests that the cryptosystem should not be considered for real-world applications until the capacity of quantum processors increases or a more efficient algebraic ECC is implemented.

Despite their additional computational overhead, hybrid quantum cryptosystems like the one proposed by Pleşa are somewhat less reliant on the advancement of quantum computers because they transmit encrypted messages over classical channels. As a result, these systems have a greater likelihood of real-world implementation in the near future. Pleşa's system achieves confidentiality by use of a quantum circuit that guarantees perfect randomization[38]. While its ability to ensure confidentiality can be proven through experimentation, its real-world application is hindered by the requirement for each communicating party to share a pair of entangled qubits. The challenges introduced in maintaining and transporting entangled qubits are not unique to this scheme but are notable obstacles that are likely to delay the real-world implementation of this system.

4.4.2 Non-Repudiation. Quantum non-repudiation is challenging due to the the impossibility of unconditionally secure signatures for quantum data without a trusted third party, as well as the destructive nature of measurements for both signatures and data. Quantum signatures are frequently not reusable, which makes their transferability questionable. Quantum signatures of classical data are unlikely to be very important in the near future as classical signatures are still viable, and NIST is actively researching post-quantum signature protocols for classical data.

Quantum signatures of quantum data will become more important in the future when quantum computers become more prevalent. Just like current certificates use trusted root authorities, quantum signatures will need trusted quantum arbitrators, and this concept has not been well explored to date. Several promising quantum signature schemes have been experimentally realized in recent years, including an implementation by An et al.[8] in 2019, and another by Ding et al.[19] in 2020. Both of these practical implementations are based off of a proposal by Amiri et al.[7] in 2016 for a secure quantum signature scheme which functions over insecure channels.

In the signature protocol proposed by Amiri et al.[7], the sender, Alice, communicates with two receiving parties, Bob and Charlie.

Alice generates correlated bit strings separately with Bob and Charlie using a key generation protocol such as BB84. Bob and Charlie exchange half of their generated bit strings with one another over a secure classical channel. Alice signs and transmits a message over a classical channel to her desired recipient, say Bob, who then checks the signature against his own key and the key received from Charlie. If the number of mismatches between the signature and secret key is below some limit, then Bob can accept the message. Bob is also able to transmit the message and signature to Charlie for verification in the same manner. As quantum channels are only required during the key generation stage of this protocol, it is well suited for implementation as BB84 or other QKD protocols can be used.

5 CONCLUSION

In this paper we have explored the impacts of quantum computing on classical cryptography, examined the current state of the art of quantum cryptography, implemented and analyzed Amerimehr and Dehkordi's 2018 symmetric quantum encryption algorithm[5], and presented a generalized process for the implementation of quantum cryptographic algorithms. The impacts of quantum computing on today's cryptographic systems are well understood, and efforts are being made by NIST to select suitable replacements for vulnerable public-key encryption, key-establishment, and signature algorithms[17, 32]. We identified promising quantum algorithms for important aspects of cryptography, including twin-field QKD for key distribution, zero knowledge proofs for integrity, AD18 for encryption, and Amiri et al.[7] for non-repudiation. While some of these algorithms, such as AD18, appear to need additional research and refinement to be practical for many applications, they can serve as a basis from which to build for future research.

To aid in future evaluations of quantum cryptographic protocols, we examined implementations of BB84, AD18, and Kak's Three-Stage Protocol, and proposed a standard process by which other researchers can approach the challenge of implementing quantum cryptographic algorithms. As the field of quantum cryptography is still in its infancy, our hope is that these final two contributions will be of particular value. Finally, we will provide recommended actions organizations can take today and in the near future to prepare themselves for advances in quantum computing and cryptography.

5.1 Recommendations for Today

Our first recommendation for organizations today is to gain an awareness of quantum computing, what problems it can and cannot be used to solve, and how it threatens current cryptographic systems. Outside of cutting edge security or research concerns, we do not believe it is necessary to immediately hire quantum specialists, but having an awareness of the state of the art in quantum computing will likely provide a competitive advantage to businesses over the next few years. Software engineers and security experts would benefit from learning the basic concepts of quantum computing as well, just as they have been encouraged to do with concepts such as cloud computing and machine learning in the recent past.

As today's most commonly used encryption systems are either weakened or broken by quantum computing, we highly recommend that companies be prepared to implement post-quantum encryption once final candidate algorithms are approved by NIST. Though quantum computers will not be capable of breaking RSA2048 for many years, encrypted data could still be captured by an adversary today and stored until decryption becomes feasible in the future. For data at rest, we recommend using AES with a 256-bit key length, and discontinuing the usage of 3DES and AES with 128- or 192-bit key lengths, as the impact of Grover's algorithm lowers the effective key lengths below those recommended by NIST in Section 3.4 of Special Publication 800-175B[10].

5.2 Recommendations for the Future

It is extraordinarily difficult for experts to make accurate predictions of future developments in any field. The authors of this work make no claims to be more than novices in the field of quantum computing, but we felt we would be remiss to not provide a couple of broad predictions and accompanying recommendations for the next ten to twenty years.

In the next two decades, we predict that quantum computing will become widespread. Quantum computers will become more capable and easier to access, and an increasing number of companies will hire quantum computing experts. Quantum algorithms will be commonly used for applications such as entropy generation, key distribution, and optimization problems. If dramatic hardware advancements are made, we may even see specialized quantum processing units with a small number of qubits appear in personal computers. To stay ahead of these predictions, we reiterate our previous recommendation that organizations begin building their quantum computing capabilities. We also recommend that organizations be wary of the inevitable wave of charlatan companies that will promise expensive quantum computing offerings which are capable of solving all the world's problems.

As quantum computers become ubiquitous and capable, we predict that large numbers of new algorithms will be discovered. These algorithms will have broad impacts, ranging from threatening the security of previously safe cryptographic protocols, to producing rapid advances in materials science, medicine, and artificial intelligence. The technological and sociopolitical impacts of these new discoveries may alter society as fundamentally as industrialization, automobiles, aviation, and the internet did. Our recommendation is that organizations be prepared to adapt to rapid paradigm shifts in the security and technology landscape.

5.3 Future Research

In future work, we would like to further research and explore the encryption of quantum data, as this part of the field appears to be underdeveloped when compared to other aspects of quantum cryptography. In particular, we would like to research whether encryption protocols could diverge to handle quantum data at rest and in motion, as we have seen symmetric and asymmetric encryption develop in the classical realm. We believe an opportunity exists to further generalize our process for implementing quantum cryptographic algorithms, so that it can apply to non-circuit paradigms of quantum computing. We would also like to further explore the space of algebraic error correcting codes in order to revisit AD18 and improve the efficiency and overall capability of the protocol.

6 FINAL PAPER CONTRIBUTIONS

Sections of this paper were contributed as follows:

- Abstract and initial outline - Henry
- Introduction - Zexi
- Related Work - Zexi, Bryan (Moved a large portion of Bryan's work from methodology to here)
- Methodology - Full team. QKD was written by Zexi, Confidentiality by Casey, Integrity by Bryan, Non-Repudiation by Henry.
- Analysis - Henry (implementation and analysis), Casey (Discussion)
- Conclusion - Henry
- LaTeXtypesetting - Henry
- Proofreading and editing - Full team

REFERENCES

[1] [n.d.]. Quantum Key Distribution. https://community.qiskit.org/textbook/ch-algorithms/quantum-key-distribution.html

[2] [n.d.]. What Is a DoS Attack? https://academy.binance.com/en/articles/what-is-a-dos-attack

[3] 2020. IBM's Roadmap For Scaling Quantum Technology. https://www.ibm.com/blogs/research/2020/09/ibm-quantum-roadmap/

[4] Tameem Albash and Daniel A. Lidar. 2016. Adiabatic Quantum Computing. (Nov. 2016). https://doi.org/10.1103/RevModPhys.90.015002

[5] A. Amerimehr and M. H. Dehkordi. 2018. Quantum Symmetric Cryptosystem Based on Algebraic Codes. *IEEE Communications Letters* 22, 9 (Sept. 2018), 1746–1749. https://doi.org/10.1109/LCOMM.2018.2844245

[6] Ryan Amiri and Erika Andersson. 2015. Unconditionally Secure Quantum Signatures. *Entropy* 17, 8 (Aug. 2015), 5635–5659. https://doi.org/10.3390/e17085635

[7] Ryan Amiri, Petros Wallden, Adrian Kent, and Erika Andersson. 2016. Secure quantum signatures using insecure quantum channels. *Physical Review A* 93, 3 (March 2016), 032325. https://doi.org/10.1103/PhysRevA.93.032325

[8] Xue-Bi An, Hao Zhang, Chun-Mei Zhang, Wei Chen, Shuang Wang, Zhen-Qiang Yin, Qin Wang, De-Yong He, Peng-Lei Hao, Shu-Feng Liu, Xing-Yu Zhou, Guang-Can Guo, and Zheng-Fu Han. 2019. Practical quantum digital signature with a gigahertz BB84 quantum key distribution system. *Optics Letters* 44, 1 (Jan. 2019), 139–142. https://doi.org/10.1364/OL.44.000139

[9] Frank Arute, Kunal Arya, Ryan Babbush, Dave Bacon, Joseph C. Bardin, Rami Barends, Rupak Biswas, Sergio Boixo, Fernando G. S. L. Brandao, David A. Buell, Brian Burkett, Yu Chen, Zijun Chen, Ben Chiaro, Roberto Collins, William Courtney, Andrew Dunsworth, Edward Farhi, Brooks Foxen, Austin Fowler, Craig Gidney, Marissa Giustina, Rob Graff, Keith Guerin, Steve Habegger, Matthew P. Harrigan, Michael J. Hartmann, Alan Ho, Markus Hoffmann, Trent Huang, Travis S. Humble, Sergei V. Isakov, Evan Jeffrey, Zhang Jiang, Dvir Kafri, Kostyantyn Kechedzhi, Julian Kelly, Paul V. Klimov, Sergey Knysh, Alexander Korotkov, Fedor Kostritsa, David Landhuis, Mike Lindmark, Erik Lucero, Dmitry Lyakh, Salvatore Mandrà, Jarrod R. McClean, Matthew McEwen, Anthony Megrant, Xiao Mi, Kristel Michielsen, Masoud Mohseni, Josh Mutus, Ofer Naaman, Matthew Neeley, Charles Neill, Murphy Yuezhen Niu, Eric Ostby, Andre Petukhov, John C. Platt, Chris Quintana, Eleanor G. Rieffel, Pedram Roushan, Nicholas C. Rubin, Daniel Sank, Kevin J. Satzinger, Vadim Smelyanskiy, Kevin J. Sung, Matthew D. Trevithick, Amit Vainsencher, Benjamin Villalonga, Theodore White, Z. Jamie Yao, Ping Yeh, Adam Zalcman, Hartmut Neven, and John M. Martinis. 2019. Quantum supremacy using a programmable superconducting processor. *Nature* 574, 7779 (Oct. 2019), 505–510. https://doi.org/10.1038/s41586-019-1666-5

[10] Elaine Barker. 2020. *Guideline for using cryptographic standards in the federal government:: cryptographic mechanisms.* Technical Report NIST SP 800-175Br1. National Institute of Standards and Technology, Gaithersburg, MD. NIST SP 800–175Br1 pages. https://doi.org/10.6028/NIST.SP.800-175Br1

[11] Howard Barnum, Claude Crepeau, Daniel Gottesman, Adam Smith, and Alain Tapp. 2002. Authentication of Quantum Messages. *The 43rd Annual IEEE Symposium on Foundations of Computer Science, 2002. Proceedings.* (2002), 449–458. https://doi.org/10.1109/SFCS.2002.1181969 arXiv: quant-ph/0205128.

[12] Ethan Bernstein and Umesh Vazirani. 1997. Quantum Complexity Theory. *SIAM J. Comput.* 26, 5 (Oct. 1997), 1411–1473. https://doi.org/10.1137/S0097539796300921

[13] Gilles Brassard, Norbert Lütkenhaus, Tal Mor, and Barry C. Sanders. 2000. Limitations on Practical Quantum Cryptography. *Physical Review Letters* 85, 6 (Aug. 2000), 1330–1333. https://doi.org/10.1103/PhysRevLett.85.1330

[14] André Chailloux, María Naya-Plasencia, and André Schrottenloher. 2017. An Efficient Quantum Collision Search Algorithm and Implications on Symmetric Cryptography. In *Advances in Cryptology – ASIACRYPT 2017 (Lecture Notes in Computer Science)*, Tsuyoshi Takagi and Thomas Peyrin (Eds.). Springer International Publishing, Cham, 211–240. https://doi.org/10.1007/978-3-319-70697-9_8

[15] Feng-Lin Chen, Wan-Fang Liu, Su-Gen Chen, and Zhi-Hua Wang. 2018. Public-key quantum digital signature scheme with one-time pad private-key. *Quantum Information Processing* 17, 1 (Jan. 2018), 10. https://doi.org/10.1007/s11128-017-1778-5

[16] Cheng, H., Großschädl, J., Rønne, P., and Ryan, P. 2021. A Lightweight Implementation of NTRUEncrypt for 8-bit AVR Microcontrollers. (Feb. 2021). https://doi.org/10.5281/ZENODO.4431753

[17] Information Technology Laboratory Computer Security Division. 2017. Post-Quantum Cryptography | CSRC | CSRC. https://csrc.nist.gov/projects/post-quantum-cryptography

[18] Marcos Curty and David J. Santos. 2001. Quantum authentication of classical messages. (March 2001). https://doi.org/10.1103/PhysRevA.64.062309

[19] Hua-Jian Ding, Jing-Jing Chen, Liang Ji, Xing-Yu Zhou, Chun-Hui Zhang, Chun-Mei Zhang, and Qin Wang. 2020. 280-km experimental demonstration of a quantum digital signature with one decoy state. *Optics Letters* 45, 7 (April 2020), 1711–1714. https://doi.org/10.1364/OL.389848

[20] Yang-Yang Fei, Xiang-Dong Meng, Ming Gao, Hong Wang, and Zhi Ma. 2018. Quantum man-in-the-middle attack on the calibration process of quantum key distribution. *Scientific Reports* 8, 1 (Dec. 2018), 4283. https://doi.org/10.1038/s41598-018-22700-3

[21] Daniel Gottesman and Isaac Chuang. 2001. Quantum Digital Signatures. *arXiv:quant-ph/0105032* (Nov. 2001). http://arxiv.org/abs/quant-ph/0105032 arXiv: quant-ph/0105032.

[22] Markus Grassl, Brandon Langenberg, Martin Roetteler, and Rainer Steinwandt. 2016. Applying Grover's Algorithm to AES: Quantum Resource Estimates. In *Post-Quantum Cryptography*, Tsuyoshi Takagi (Ed.). Vol. 9606. Springer International Publishing, Cham, 29–43. https://doi.org/10.1007/978-3-319-29360-8_3

[23] Nafiseh Hematpour, Sodeif Ahadpour, and Sohrab Behnia. 2020. Presence of dynamics of quantum dots in the digital signature using DNA alphabet and chaotic S-box. *Multimedia Tools and Applications* (Nov. 2020). https://doi.org/10.1007/s11042-020-10059-5

[24] Samuel Jaques, Michael Naehrig, Martin Roetteler, and Fernando Virdia. 2020. Implementing Grover Oracles for Quantum Key Search on AES and LowMC. In *Advances in Cryptology – EUROCRYPT 2020*, Anne Canteaut and Yuval Ishai (Eds.). Vol. 12106. Springer International Publishing, Cham, 280–310. https://doi.org/10.1007/978-3-030-45724-2_10

[25] Norman Jorstad and Landgrave T. Smith. 1997. Cryptographic Algorithm Metrics. In *Proceedings of the 20th National Information Systems Security Conference*. Baltimore, Maryland, United States. https://csrc.nist.gov/csrc/media/publications/conference-paper/1997/10/10/proceedings-of-the-20th-nissc-1997/documents/128.pdf

[26] Subhash Kak. 2006. A Three-Stage Quantum Cryptography Protocol. *Foundations of Physics Letters* 19, 3 (June 2006), 293–296. https://doi.org/10.1007/s10702-006-0520-9

[27] Min-Sung Kang, Chang-Ho Hong, Jino Heo, Jong-In Lim, and Hyung-Jin Yang. 2015. Quantum Signature Scheme Using a Single Qubit Rotation Operator. *International Journal of Theoretical Physics* 54, 2 (Feb. 2015), 614–629. https://doi.org/10.1007/s10773-014-2254-y

[28] E. Khan, S. Meraj, and M. M. Khan. 2020. Security Analysis of QKD Protocols: Simulation Comparison. In *2020 17th International Bhurban Conference on Applied Sciences and Technology (IBCAST)*. 383–388. https://doi.org/10.1109/IBCAST47879.2020.9044522 ISSN: 2151-1411.

[29] Xiang Liu, Xiaosong Yu, Yongli Zhao, Xiaotian Zhou, Shimulin Xie, Jincheng Li, and Jie Zhang. 2019. Multi-path based Quasi-real-time Quantum Key Distribution in Software Defined Quantum Key Distribution Networks (SD-QKDN). In *2019 18th International Conference on Optical Communications and Networks (ICOCN)*. 1–3. https://doi.org/10.1109/ICOCN.2019.8934684

[30] S. Mandal, G. Macdonald, Mayssaa El Rifai, N. Punekar, F. Zamani, Yuhua Chen, S. Kak, P. K. Verma, R. C. Huck, and J. Sluss. 2013. Multi-photon implementation of three-stage quantum cryptography protocol. In *The International Conference on Information Networking 2013 (ICOIN)*. IEEE, Bangkok, 6–11. https://doi.org/10.1109/ICOIN.2013.6496343

[31] Vasileios Mavroeidis, Kamer Vishi, Mateusz D., and Audun Jøsang. 2018. The Impact of Quantum Computing on Present Cryptography. *International Journal of Advanced Computer Science and Applications* 9, 3 (2018). https://doi.org/10.14569/IJACSA.2018.090354

[32] Dustin Moody. 2020. NIST PQC Standardization Update - Round 2 and Beyond. https://csrc.nist.gov/CSRC/media/Presentations/pqc-update-round-2-and-beyond/images-media/pqcrypto-sept2020-moody.pdf

[33] National Security Agency (NSA). [n.d.]. Quantum Key Distribution (QKD) and Quantum Cryptography (QC). https://www.nsa.gov/what-we-do/cybersecurity/quantum-key-distribution-qkd-and-quantum-cryptography-qc/

[34] Georgios M. Nikolopoulos and Marc Fischlin. 2020. Information-theoretically secure data origin authentication with quantum and classical resources. (Nov. 2020). https://doi.org/10.3390/cryptography4040031

[35] Ali Ibnun Nurhadi and Nana Rachmana Syambas. 2018. Quantum Key Distribution (QKD) Protocols: A Survey. In *2018 4th International Conference on Wireless and Telematics (ICWT)*. 1–5. https://doi.org/10.1109/ICWT.2018.8527822

[36] Abhishek Parakh and Joel van Brandwijk. 2016. Correcting rotational errors in three stage QKD. In *2016 23rd International Conference on Telecommunications (ICT)*. IEEE, Thessaloniki, Greece, 1–5. https://doi.org/10.1109/ICT.2016.7500409

[37] Chang Hoon Park, Min Ki Woo, Byung Kwon Park, Yong-Su Kim, Sangin Kim, and Sang-Wook Han. 2020. Research on Plug-and-Play Twin-Field Quantum Key Distribution. In *2020 International Conference on Information and Communication Technology Convergence (ICTC)*. 890–893. https://doi.org/10.1109/ICTC49870.2020.9289265 ISSN: 2162-1233.

[38] Mihail-Iulian Plesa. 2017. Hybrid scheme for secure communications using quantum and classical mechanisms. In *2017 9th International Conference on Electronics, Computers and Artificial Intelligence (ECAI)*. IEEE, Targoviste, 1–6. https://doi.org/10.1109/ECAI.2017.8166458

[39] Eleanor G. Rieffel and Wolfgang Polak. 1998. An Introduction to Quantum Computing for Non-Physicists. (Sept. 1998). https://arxiv.org/abs/quant-ph/9809016v2

[40] K. Saeedi, S. Simmons, J. Z. Salvail, P. Dluhy, H. Riemann, N. V. Abrosimov, P. Becker, H.-J. Pohl, J. J. L. Morton, and M. L. W. Thewalt. 2013. Room-Temperature Quantum Bit Storage Exceeding 39 Minutes Using Ionized Donors in Silicon-28. *Science* 342, 6160 (Nov. 2013), 830–833. https://doi.org/10.1126/science.1239584

[41] Donglu Shi, Zizheng Guo, and Nicholas Bedford. 2015. *Superconducting Nanomaterials*. William Andrew Publishing. https://doi.org/10.1016/B978-1-4557-7754-9.00008-1

[42] Peter W. Shor. 1997. Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer. *SIAM J. Comput.* 26, 5 (Oct. 1997), 1484–1509. https://doi.org/10.1137/S0097539795293172

[43] Ahmed Younes. 2015. A bounded-error quantum polynomial-time algorithm for two graph bisection problems. *Quantum Information Processing* 14, 9 (Sept. 2015), 3161–3177. https://doi.org/10.1007/s11128-015-1069-y

[44] Wei Yu, Yuanyuan Zhou, Xuejun Zhou, Lei Wang, and Shang Chen. 2020. Study on Statistical Analysis Method of Decoy-state Quantum Key Distribution with Finite-length Data. In *2020 IEEE 4th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*, Vol. 1. 2435–2440. https://doi.org/10.1109/ITNEC48623.2020.9084715

[45] Manjin Zhong, Morgan P. Hedges, Rose L. Ahlefeldt, John G. Bartholomew, Sarah E. Beavan, Sven M. Wittig, Jevon J. Longdell, and Matthew J. Sellars. 2015. Optically addressable nuclear spins in a solid with a six-hour coherence time. *Nature* 517, 7533 (Jan. 2015), 177–180. https://doi.org/10.1038/nature14025