Quantum Key Distribution: Leveraging the Laws of Physics for Perfectly Secure One-Time Pad Encryption

Alexis Goodfellow

April 13, 2019

Abstract

A fundamental problem that encryption algorithm designers must face is that the information they wish to secure must be sent through a classical network connection, which is inherently insecure. On the classical computers that dominate the modern computing landscape, encryption is achieved through numerous methods that all rely on computationally intractable problems in mathematics. Quantum Key Distribution - or QKD - is interesting because it does not rely on computational complexity to ensure that data is protected; its security is instead guaranteed by the laws of physics. Furthermore, QKD is advantageous because the key exchange process itself is sensitive to eavesdropping, and transmissions can be automatically canceled should an eavesdropper be detected. This paper is intended to bring these exciting implications of QKD to the attention of the wider information security community while being presented in a manner that is accessible without a substantial background in the inner workings of particle physics.

1 Introduction

To the general public, the principles behind quantum mechanics and quantum computing are generally either misunderstood or completely unknown. This is partially due to the difficulty in understanding the concepts, but it is also due to a lack of attention from popular to the core principles that govern the physical world at the quantum level. When the media does cover these topics, it usually does so in a manner that is sensationalist. Journalists tend to use the phrase "the quantum world"; as if to divorce the effects of quantum mechanics from the classical physics that govern .

Presenting "the quantum world" as one entirely separate from the macroscopic world is a fundamental misstep. Quantum phenomena are very much real and tangible. Quantum effects may be hard to comprehend - and even harder to encapsulate in an intuitive mental model of reality - but their effects and implications cannot and should not be divorced from the macroscopic world. The original formation of the thought experiment called "Schrödinger's Cat" - named after its originator, Erwin Schrödinger - is a wonderful example of the inseparable nature of behavior at the quantum level and behavior at the macroscopic level.

Any discussion relating the microscopic world of quantum particles to the macroscopic world of computing machinery must first dispel this notion that there exists a dichotomy between the quantum world and the macroscopic world. Indeed, the essential premise of quantum computing is that it is possible to take advantage of microscopic quantum phenomena to produce meaningful information which can be utilized by macroscopic systems. This quantum information is processed and consumed in a manner completely unlike the classical information contained in the transistors that form the memory of classical computers. Quantum Key Distribution is an interesting and active area of research which utilizes quantum effects to produce encryption keys, which in turn allow for the computers generating the keys to securely share secrets through a public network without fear of being spied on.

2 To the community

There is both a short-term advantage and a long-term advantage to QKD over the classical key generation protocols with which the working security professional is already familiar. In the short

term, QKD offers the ability to detect eavesdroppers before sending sensitive information across a network. In the long term, QKD provides protection against quantum computers, which will break the majority of popular asymmetric encryption algorithms extremely efficiently.

Classical networks have no mechanism for detecting eavesdropping. Because eavesdroppers are impossible to detect on these classical networks, classical computers must send sensitive information regardless of whether there is an eavesdropper present or not. This has the potential to be quite dangerous, but at present it is a risk that simply must be accepted. The immediate benefit of QKD over other modern asymmetric encryption algorithms is that with QKD it possible to detect the presence of an eavesdropper during the key distribution process. Since this eavesdropper can be detected before any sensitive information is sent across a network, both of the communicating parties can act with the knowledge that there is an eavesdropper, which can include aborting the transmission entirely. In the short term, QKD protocols with this property alone are worth studying, researching, and implementing.

In the long term, QKD provides protection against an important potential threat to most major asymmetric encryption algorithms: Shor's Algorithm. Shor's Algorithm has been proven to be capable of breaking many popular encryption algorithms, but it requires a quantum computer in order to be efficient enough to be a viable encryption cracking technology. While the large-scale quantum computers needed for Shor's Algorithm are many years from being fully developed, the fact remains that small-scale proof-of-concept experiments for Shor's Algorithm have already been successful. There is no reason to believe that quantum computers of a significant size cannot be built, and it is the imperative of the security community - particularly government agencies - to mitigate these potential threats to information security before they become exploitable. QKD protocols provably cannot be broken by Shor's Algorithm, which makes QKD even more important for the security community to understand and research.

3 Defining Security - The CIA Triad

In order to explain how QKD ensures information security, it's first important to define what exactly it means for information to be secure. The standards for security used here are referenced directly from the widely used standard known as the CIA Triad. CIA is an acronym, which in this context stands for "Confidentiality", "Integrity", and "Availability". Confidentiality is the rule system that assures the secret information is held only by those who should be in possession of it. Integrity is the assurance that the authenticity and content of the information has not been compromised in any way. Availability means that access to the secret information should be consistently allowed for those who can supply the proper credentials to view said information.

4 Essential Properties of Cryptographic Systems

Cryptographic systems are critical to information security precisely because they implement all of the security concerns of confidentiality, integrity, and availability simultaneously. In order to accomplish this, every algorithm designed to keep information secure has some necessary conditions which must be satisfied in order to qualify as an encryption algorithm.

Firstly, unlike a hashing algorithm which can be used only to check the integrity of such pieces of information as passwords, all encrypted messages must be capable of being decrypted back to the original message. Formally, for any operation f that is applied to a message x, there must exist an inverse operation f' such that $f'(f(x)) = x$. A proof of this reversibility property also provides a proof of integrity; it demonstrates that unless tampered with in transit, the message being received has the same contents as the message that was sent.

In order for encryption algorithms to ensure confidentiality, it must be the case that if Alice sends a message to Bob, an eavesdropper Eve cannot reverse the encrypted message into its original form. Formally, this means that if Eve receives an encrypted message $f(x)$, it is computationally intractable to generate any f' such that $f'(f(x)) = x$. These functions f and f' are commonly conceptualized with the metaphor of "keys", which when used on the contents of a "locked" message will "unlock" its contents. If an algorithm can generate secure "keys" f and f' in an automated fashion, then this also satisfies the idea of availability.

In short, any encryption algorithm must be reversible, it must be impossible to reconstruct the "keys" in order to decrypt the information, and there must be a process by which generating

"keys" can be made automatic.

5 Cryptography in Modern Practice

There are a great many different encryption schemes used in modern practice. Just some of the notable symmetric encryption algorithms are AES, Blowfish, DES, and IDEA. Among the notable asymmetric encryption algorithms are RSA, Diffie-Hellman, DSA, Elliptic Curve DSA (ECDSA), and XTR. All of these algorithms vary in difficulty for classical computers to break, but the asymptotic runtime complexity for breaking any of these schemes is exponential.

However, all of the asymmetric encryption schemes rely on certain classes of mathematical problems being computationally intractable. RSA, for example, relies on the fact that classical computers cannot factor semi-prime numbers efficiently. Diffie-Hellman relies on the difficulty of computing discrete logarithms, another problem that cannot be solved in less than exponential time. Elliptic Curve DSA - which was once an encryption algorithm used by the NSA - relies on the computational difficulty of the elliptic-curve discrete logarithm problem.

If it is possible to implement Shor's Algorithm on a large enough quantum computer, all three of these computationally intractable problems will have efficient solutions of linear time complexity. RSA, Diffie-Hellman, ECDSA, and many more common asymmetric encryption algorithms will be completely broken, and broken beyond repair. Simply increasing the key sizes to larger numbers of bits will not be enough; these encryption schemes would need to be overhauled completely.

The fundamental flaw with every one of these encryption algorithms is that they rely on the assumption that certain problems in mathematics will be intractable forever. When quantum computers become sizable enough and Shor's Algorithm is implemented well enough, that assumption will no longer be sound.

QKD takes a different approach entirely. Instead of relying on the difficulty of solving some kind of mathematical problem, a more secure encryption scheme would be to use "keys" that are not merely difficult to brute force, but provably *impossible* to brute force. Fortunately, there is already a known algorithm for this - it is known as the one-time pad.

Though this algorithm has been known and proven to be secure for more than a century, it has been impossible to implement in the context of computer networks. The fundamental difficulty of implementing one-time pad is that in order for the algorithm to meet the criteria of being a reversible process, both parties need to agree on the pad to use before the message itself is transmitted. Thus, the two actors Alice and Bob would have to agree on a random sequence of bits to use for the pad, but they must do so without transmitting the pad itself over the network for fear of it being intercepted by Eve. This key exchange cannot occur securely over a classical network, because classical network traffic can be easily and invisibly intercepted while it is in transit.

This is not to say that one-time pad is not used. Government agents occasionally hand-deliver one-time pads to foreign dignitaries, when the information being sent is absolutely top secret. However, this process is expensive, time-consuming, and doesn't satisfy the requirement of availability. In an ideal scenario, the pad could be agreed upon using only information gleaned from network traffic, while simultaneously not transmitting any data that an eavesdropper could use to construct the pad. As of the writing of this paper, there is no known algorithm that can be implemented on a classical computer that can achieve this ideal key distribution scenario.

Though a classical computer on a classical network may not be able to implement one-time pad, it is possible to leverage quantum phenomena and probabilistic analysis such that Alice and Bob can agree upon the contents of the pad without ever sending the pad itself across the network. Furthermore, any eavesdropping by Eve can be detected and responded to before sensitive information crosses the network.

In order to understand how quantum phenomena could make such an ideal scenario become actualized in reality, it is first necessary to have some understanding of the underlying mechanisms that enable quantum computers to work so differently from classical computer architectures.

6 Essential Quantum Concepts in Brief

Though the subject of this paper is not to relate the technical mathematics of the underlying quantum interactions, it is imperative to understand these mechanisms at a conceptual level in order

to understand why the mechanisms behind quantum key distribution are simply not achievable on a classical computer. Though a quantum physicist might claim the following explanations to be incomplete, they should be serviceable for the purposes of comprehending the main principles of QKD.

6.1 Qubits and Bits

A single bit is the smallest possible unit of differentiable information, as it can only encode two possible values: 0 and 1. Classical computers use transistors to model bits, which model 0 and 1 through the presence or absence of stored electrical energy. Every quantum particle has a property called "spin", which only has two possible states; it can be either "spin-up" or "spin-down". Generating a mapping function from the set of spins to the set $\{0, 1\}$ is trivial. For this reason, spin states can be referred to as "qubits", short for "quantum bits".

In truth, any property of a particle that can be trivially mapped to the set of values expressible by a single bit can be used as a qubit. The set of these properties are not limited to spin states; they also include polarization states of photons - which can only be positive or negative. In fact, polarization states are the most commonly used type of qubit in modern research [1]. Nonetheless, it is these properties of particles that encode information inside a quantum computer.

Qubits can be generated by two physical processes. In order to remove the unnecessarily mathematical quantum details, it is best to consider these processes in an abstract sense. Call them p_A and p_B. Because of the underlying quantum mechanics, it is provably impossible to determine whether a given qubit was produced by p_A or p_B without observing the qubit the directly. In addition, it is impossible to make a copy of a qubit [2]; qubits can only be generated, transferred through a quantum communication channel, and observed. Furthermore, the observation of a qubit results in its destruction.

6.2 Measuring Quantum States

In physics, Heisenberg's Uncertainty Principle states that there are certain properties about quantum particles that cannot be measured accurately at the same time. A classic example of a pair of properties that cannot be studied simultaneously is the location of a particle in space and its momentum. To study the position of a particle precisely, it is necessary to sacrifice precision in the measurement of said particle's momentum, and vice versa.

A direct consequence of Heisenberg's Uncertainty Principle is that observing a quantum system changes the evolution of said system. In classical physics, objects are thought of as having concrete positions in physical space. For instance, this paper - assuming it is actually printed on paper - is a physical object, and its position in space can be concretely described through a set of coordinates, such as GPS coordinates. If you were to set this paper down and leave the room, its position in the room would not change.

Once at the quantum level, though, these ideas about objects having a concrete position in space no longer hold true. Quantum objects such as photons do not have a concrete position until they are directly observed. This idea that a particle like a photon has no absolute state until being observed is the fundamental notion underlying superposition.

Though it is impossible to say *exactly* where any individual photon is before observing it, there may be other portions of a quantum system which are known. These pieces of knowledge about the system mean that it is possible to determine the *probability* of finding a quantum particle in a given state. All knowledge about the possible observable features of a particle in a system is encapsulated in a single mathematical expression, known as its wavefunction. When the particle is observed, the wavefunction representing the field of possible states of that particle collapses down to one single state. This is why observing a qubit destroys the qubit; its state will be changed irreversibly through the mere act of observation.

6.3 Randomness

It is impossible to generate random numbers on classical computers. At best, classical computers can generate pseudo-random numbers. The "pseudo" qualifier in "pseudo-random" is important: even if the output of the function generates values evenly across the result space with no apparent connection between its input and output values, those output values are still acquired through a

deterministic process that could theoretically be reverse-engineered. No matter how complex the algorithm, a deterministic process is incapable of generating true randomness.

Though classical computers are deterministic by nature, the observation of quantum particles in superposition can be utilized to generate truly random numbers. These Quantum Random Number Generators can be used for a wide variety of applications, but they are useful in the context of QKD because they allow for truly random raw key generation. In short, the randomness that originates in the act of observing particles in the physical world propagates through into the computational system.

6.4 Entanglement

Though the state of a particle cannot be known until it is observed, it is possible to manufacture a system whereby the state of two particles are strongly coupled. This coupling can be thought of as an "entanglement". Call these two discrete particles α and β. If it is possible to generate this entangled pair of particles α and β, then collapsing the wavefunction for one of the particles α means information is instantaneously known about β.

It is important to qualify that *deterministic* information cannot be communicated via entanglement. After all, the observation of either part of the entangled pair will generate a random result. All that entanglement allows us to claim is that if α and β are entangled and α is observed, then β will have the complementary result to α.

This is a very theoretical notion that is best explained by a concrete example. Suppose a scientist manufactures two entangled particles, such that α is contained in their lab, and β is in some other lab in a completely different country. The scientist could only possibly observe α. The scientist makes this observation, and determines that α is spin-up. Even though the scientist cannot directly observe β - on account of β being in a different country - the scientist knows that β is entangled with α. Since entangled particles must be in complementary states, the scientist knows that β must be spin-down. Thus, the scientist knows information about the state of β *without ever directly observing* β.

7 Quantum Key Distribution Protocols

In order to fully explain how any QKD protocol works, it is necessary to introduce the idea of qubit observing machines. Though these machines have analogues in the real world and their behavior can be proven through the laws of physics, it is best to think of them as abstract machines to avoid some of the uglier mathematics that occur at the quantum level.

As discussed previously in section 6.1, qubits can be generated by one of two abstract processes, p_A or p_B. There is no way to determine whether a given qubit was spawned through either p_A or p_B. Likewise, there exist abstract machines Observer A and Observer B, both of which are designed to observe an input qubit and produce a set of classical bits as output. For the purposes of brevity, call these abstract machines o_A, o_B. For any given classical bit x, the following algebraic rules apply[1]:

- $o_A(p_A(x)) = x$
- $o_A(p_B(x)) = random\ result$
- $o_B(p_A(x)) = random\ result$
- $o_B(p_B(x)) = x$

7.1 The E91 Protocol: QKD via Quantum Entanglement

The E91 protocol for quantum key distribution utilizes the idea of entanglement in order to generate secure encryption keys. Originally proposed by Artur Ekert, E91 utilizes a source that Alice and Bob both trust - denoted here by Sue - to generate a series of entangled particle pairs [3]. This protocol exploits a unique property of entangled particle pairs; if Alice and Bob use the same observer machine on particles which are a part of the same entangled pair, then the underlying

[1]The E91 protocol actually uses three observation machines, but their algebraic properties are not included as they are irrelevant to the E91 protocol.

quantum physics shows that the bit values generated by recording each member of the entangled pair will be exactly opposite. The following is a sequence of steps used in the E91 protocol, from the point where Bob requests some secret information from Alice:

1. Alice alerts Sue that she is about to send a message. In response, Sue generates a series of entangled particle pairs, and for each pair sends one of the entangled particles to Alice, and the other to Bob.

2. Alice and Bob record both the results of observing the incoming qubits, and which machines they used for their observations of the incoming qubits. They encode the result of their observation of the qubit with one bit in b_{results}, and the machines used to observe the qubit on two bits in $b_{\text{observers}}$. Of the possible four permutations of these two bits, three are used to represent which one of the three observer machines was used on the incoming qubit. The fourth represents an error state where a qubit was expected but not detected.

3. After making their observations, Alice and Bob send their local $b_{\text{observers}}$ bit vector to each other over a public classical channel. Since Alice and Bob both now know which observation machines the other used, they can sort the results into three categories: those where they agreed on which observation machine to use, those where they did not agree on the same observation machine, and those where some error occurred.

    - If either Alice or Bob recorded an error state for the evaluation of a component of an entangled pair, the result recorded from that evaluation in b_{results} is thrown out and unused.
    - If Alice and Bob used the same observer on both entangled particles, then Alice knows that Bob recorded the exact opposite result as she did. Both parties take all the results from b_{results} for which their choice of observation machine was correct, and add that to another bit vector called b_{correct}.
    - If Alice and Bob did not use the same observer on both entangled particles, they get added to another bit vector, b_{wrong}.

4. Alice and Bob exchange their versions of b_{wrong} on a classical channel. Using this and their knowledge of which observation machines were used throughout the process, they perform a probabilistic analysis to determine whether the quantum channel was truly private. This is the stage at which an eavesdropper would be detected mathematically, but the math behind the process is in-depth and unnecessary for a top-level understanding of the protocol.

5. If the channel is determined to be compromised, Alice can terminate the transmission before sending any sensitive information. If the channel is determined to be secure, Alice proceeds with the one-time pad algorithm. All Alice needs to do is flip the values of all the bits in her version of b_{correct}, and her key will match Bob's version of b_{correct} exactly. She will utilize this bit-flipped version of her b_{correct} bit vector as the pad, so that Bob can decrypt it trivially.

7.2 The BB84 Protocol: QKD via Heisenberg's Uncertainty Principle

The BB84 protocol - named after its inventors Bennet and Brassard - communicates quantum information like the E91 protocol, but does not rely on entanglement or a mutually trusted source to do so [4]. Instead, it only relies on probability and the existence of a quantum channel. With the algebraic rules at the beginning of this section in mind, here is a rough algorithmic outline of the BB84 protocol, from the point where Bob has requested a secret piece of information from Alice:

1. Alice uses a quantum random number generator to generate a bit vector much longer than the message she wants to send. For brevity, call this bit vector b_{raw}. Alice uses this quantum random number generator to generate another bit vector called $b_{\text{processes}}$, such that $b_{\text{processes}}$ has the same length as b_{raw}. The bits in $b_{\text{processes}}$ will represent whether she will use p_A or p_B on a given bit x in the bit vector b_{raw}.

2. For every bit in b_{raw}, she sends a qubit to Bob across a quantum channel generated by the corresponding process encoded in $b_{\text{processes}}$. For an example case, if the first bit in b_{raw} is x and the first bit in $b_{\text{processes}}$ is 0, then Alice will send the qubit $p_A(x)$. If the first bit in $b_{\text{processes}}$ is 1, she will send the qubit $p_B(x)$ instead.

3. As Bob receives these qubits across the quantum channel, he has no information about which process spawned them, so he has no idea what machine to use to reverse them. He decides which machine to use for each incoming qubit by a quantum random number generated bit sequence of his own, called b_{guesses}. In the average case, Bob will choose the correct machine to use on half of these incoming qubits, and the incorrect machine on the other half. Choosing the correct machine guarantees a correct result, and - on average - choosing an incorrect machine will return a correct result value in half the cases, simply by random chance. He records these results in memory in a bit vector called b_{results}.

4. Bob now has stored in memory two bit vectors. The first is b_{guesses}, representing *which machines he used* on the incoming qubits. The second is b_{results}, representing the results of *applying those machines* to the incoming qubits. Of these two bit vectors, Bob only sends the bit vector b_{guesses} to Alice over a classical channel. He keeps the bit vector b_{results} private.

5. Alice now has access to a bit vector representing which machines Bob chose to use to observe the qubits - called b_{guesses}. She also has b_{raw} and $b_{\text{processes}}$ stored in memory. Since Alice knows b_{guesses} and $b_{\text{processes}}$, she can generate yet another bit vector representing whether or not Bob guessed the correct machine for each qubit Alice sent across the quantum channel. Alice sends Bob this new bit vector, $b_{\text{correctness}}$.

6. Bob now knows both b_{results} and $b_{\text{correctness}}$. Using these two bit vectors, he can sift out all the bits in b_{results} which would be random by the algebraic laws at the beginning of this section. By these same algebraic laws, the nonrandom bits in b_{results} will match Alice's original key, b_{raw}. The result of sifting b_{results} can be labeled b_{prelim}. At the same time, Alice does the same sifting process, but instead using $b_{\text{correctness}}$ to sift through b_{raw}. Due to the fact that both Alice and Bob have thrown out any bits for which Bob would have generated a random result, both Alice and Bob are aware of an identical bit vector b_{prelim}, *without ever having communicated it directly.*

7. If the length of b_{prelim} is less than or equal to the length of the message, the whole process starts over with a reset packet being sent by Alice to Bob. Otherwise, Alice and Bob need to assure each other that their bit vectors truly are equal. Since the resultant bit vector b_{prelim} is longer than necessary to encrypt Alice's message, Alice and Bob can exchange a portion of their respective b_{prelim} bit vectors to check for accuracy, while preserving enough random bits to use one-time pad on the entire length of the message.

8. This accuracy check is necessary to detect the presence of Eve. If the bits used to check for accuracy are all bitwise equal, then Alice and Bob know that they are both aware of a random bit sequence that is exactly long enough to use as a one-time pad on the message being sent. With this knowledge, Alice continues with the one-time pad algorithm, using this shared bit vector b_{pad}. If the accuracy check fails, Alice knows that an eavesdropper is present on the quantum channel, and she can terminate the transmission before sending any sensitive information that could be intercepted by an eavesdropper.

7.3 Detecting Eavesdropping with Probability

Both the E91 protocol and the BB84 protocol rely on probability to ensure their security. The probabilistic mechanism that the E91 protocol relies on to prove the existence of an eavesdropper is called Bell's Theorem, but the mathematics regarding the security proof of E91 are quite complex [3]. Because the security proof for the BB84 protocol is much easier to understand on a conceptual level, it will be the main focus of this section, but it should be noted that the security of the E91 protocol is also guaranteed through a different probabilistic analysis technique.

In the BB84 protocol, the initial raw key Alice generates is much longer then the message she wants to send. The protocol is designed this way for multiple reasons. The more obvious reason for this is that - in the average case - half the values derived from qubits in the initial raw key will be unusable in the BB84 protocol. This is simply due to randomness, and it is unavoidable. The more subtle reason why the key is longer than it needs to be are the final comparison bits sent in step 6 and used in step 8. This comparison step is what makes the BB84 protocol interesting; it can detect eavesdroppers purely due to the randomness of the underlying quantum physics. To understand why, it is necessary to again imagine the existence of an eavesdropper, Eve.

Firstly, even if Eve was listening to the entire conversation between Alice and Bob on the classical channel, Eve could not reconstruct the one-time pad from the data on that channel alone, because b_{raw} and b_{results} are never sent over the classical channel. In order to actually acquire the data necessary to reconstruct the pad, she would have to intercept communication on the quantum channel as well as the classical one.

Since Eve must listen on the quantum channel, that means she must intercept each qubit as it is sent from Alice to Bob. However, because of the physics governing the behavior of qubits, intercepting a qubit means observing it. As discussed in section 6.1, observing a qubit means destroying it. Thus, for every qubit Eve intercepts, she must send one of her own across the quantum channel to Bob. This is called an Intercept/Resend Attack, or I/R Attack.

Unfortunately for Eve, this introduces a large problem. Due to the fact that the qubits generated by p_A and p_B are indistinguishable, she is incapable of determining which machine to use to observe them to guarantee a correct result. She is in the same position as Bob in step 3, and has to make random guesses. Just like Bob, she will choose the correct machine in half[2] the guesses, and generate the correct data bit every three out of four qubits she measures.

For every qubit Eve measures, she has to send a qubit to Bob, or else Bob will know something is wrong with the transmission and abort it. Since Eve cannot communicate any information to Alice without revealing herself, the best she can do is to send Bob a series of qubits that is as close to correct as possible. In order to accomplish this, Eve records the result of observing the qubit sent by Alice, called x. Eve also notes which machine she used to observe Alice's qubit, called o. For each qubit Eve intercepts, she sends Bob the qubit $p(x)$, where p is the process for which $o(p(x)) = x$. Based on this behavior of Eve, the following scenario exemplifies the problem with the basic I/R attack.

Eve recieves Alice's qubit and observes it with the incorrect machine, yielding a random result. Eve encodes the random result using the opposite process as Alice, and sends it to Bob. Bob chooses to use the observer that matches Alice's original qubit. Since Eve's resent qubit has the opposite process as Alice's original qubit, the result of Bob's evaluation will be random, *even though he used the correct machine according to Alice.*

When Alice and Bob exchange subsets of the preliminary key for accuracy, these errors due to randomness from incorrect re-sends by Eve will be detected. If Eve is present and performing an I/R attack, only 75% of the bits in the accuracy check will be equal in the average case, as opposed to 100% of bits being equal if Eve is absent. Thus, Alice and Bob have a reasonably large likelihood of detecting the presence of Eve during the key generation process, before any sensitive information is sent.

8 Protocols in Practice: Attacking Implementations

Let this be completely clear: at the abstract machine level, the QKD protocols described in this paper are perfectly secure. Unfortunately, every abstract machine must eventually be implemented in some sort of physical hardware. This translation from the abstract machine to the physical machine unfortunately has the potential to introduce hardware vulnerabilities to an otherwise perfect system. These physical vulnerabilities are what attacks on QKD systems seek to exploit.

8.1 Photon Number Splitting (PNS) Attack

The ideal QKD system makes a simplifying assumption that is simply not true in any real system. The ideal scenario assumes that each qubit is sent individually, and further that each individual qubit is sent only once. Unfortunately, there is no known method for sending single qubits individually [3].

This opens any realistic QKD system up to a style of attack called a photon number splitting (PNS) attack [5]. The principle behind this attack is that rather than sending discrete photons, real implementations of QKD utilize photon pulses. The average number of photons per pulse

[2] Due to the inherent randomness present in the system, this section is written an average case analysis. In a worst case analysis, Eve would still be capable of both remaining undetected and cracking the encrypted transmission. However, the likelihood of a situation resembling this worst case scenario occurring when using large enough key sizes and transmissions is so statistically remote that it is not worth worrying about in practice.

[3] For the sake of this discussion, qubits are considered to be encoded on photons - hence the name "Photon Number Splitting".

is still extremely low, but unfortunately the hardware is not yet capable of sending individual photons. Thus, Eve only needs to intercept a fraction of the photons sent in the pulse, and the rest can remain untouched. Since Bob still receives the pulse from Alice, he will not be suspicious of the integrity of the quantum channel, and he will continue the protocols without detecting Eve.

Eve, on the other hand, is now in possession of a set of photons sent through the quantum channel. She captures these photons, but does not observe them. Instead, she waits until Bob communicates to Alice what observers he used on his incoming qubits. Now that Eve knows what observers Bob used on his qubits, she can evaluate the fraction she intercepted from the pulse with the same observers that Bob used. From the information communicated on the classical channel, Eve can then sift her key in the exact same manner as Alice and Bob, without being detected.

In practice, this attack relies on extremely specialized hardware, and requires complex algorithms in order to work properly. This makes the attack generally cost-prohibative. However, for the kinds of state actors who would be interested in learning secrets regardless of cost, the PNS attack is a viable method of undetectable key interception.

8.2 Light Injection Attack

A Light Injection Attack - also known as a trojan horse attack - targets the distribution machines themselves, instead of the quantum channel [5]. In order to understand how this attack works, it is necessary to partially remove the layer of abstraction provided by considering QKD in terms of abstract processes and observers.

In reality, these observation machines are photon detectors, and the processes are photon generators such as lasers. In a light injection attack, Eve sends her own light pulses at one of the devices used in the private conversation. This light pulse is reflected back in Eve's direction by the recipient's hardware, and registered by Eve's machine.

Because of the QKD hardware's design, this reflected pulse will indicate which process will be used by Alice to generate the next qubit. Since Eve now knows which process Alice is about to use to encode her qubit, Eve can perform an I/R attack with 100% certainty that she will use the correct observer for every incoming qubit. Because Eve will use the correct observer every time, she will be able to both capture the entire key as Alice sends it, and send qubits to Bob with assurance that neither Alice or Bob will be able to detect her presence. Using the information communicated over the classical channel, Eve will be able to synthesize the same key as Alice and Bob.

8.3 Faked-States Attack

The Faked-States Attack is also based on the I/R attack. The critical observation that makes the Faked-States attack more devastating than a simple I/R attack is that so long as Eve can be certain that Bob will record the qubits Eve resends such that Bob will reproduce Eve's results, than Eve can use the verification information communicated between Alice and Bob over the classical channel to determine 100% of their shared key *while remaining completely undetected.*

The attack works by sending enough background photon radiation at Bob's detectors so that they are effectively "blinded" by receiving the same small amount of background photon noise [6]. Superimposed on top of this background radiation, Eve sends a bright polarized light pulse. Due to the hardware used to detect photons, this extra spike of light intensity will be detected by only one of Bob's machines, in such a way that Eve can force Bob to produce the same results that Eve generates from intercepting the qubit stream.

Since Eve can manufacture the light pulses such that Bob always detects the photon pulse in the same way Eve does, Eve can force Bob to record the same results she possesses. Since Eve and Bob's results are identical, when Alice and Bob communicate on the classical channel, Eve can use that communication to synthesize the same key that Alice and Bob believe to be private. When the encrypted message is sent across the classical channel from Alice to Bob, Eve will be able to decrypt the transmission trivially.

9 Action Items

It is an unfortunate truth that at present, the technology required to implement unconditionally secure QKD systems in hardware is not yet widely available. There is still much research to be

done in this field, and there are three significant outstanding problems preventing physical QKD systems from being as secure as their abstract counterparts. Besides monetary expense, these are the current unsolved problems preventing QKD from becoming as widespread as the public-private key algorithms commonly used by contemporary classical computers.

9.1 Detection Mechanisms

In order to secure against both the trojan horse attack and the faked-states attack, the photon detection hardware needs to be improved. Such detectors ideally need to be designed to be sensitive enough to increases in background photon radiation so that they can be detect an incoming faked-states attack. Furthermore, these detectors need to be designed to not reflect light pulses in such a way that Eve can perform a light injection attack on either communication apparatus. In short, the physical devices used to detect photons in QKD systems need further research and investment.

9.2 Single Photon Generation

Alongside the need for more sensitive photon detectors, the security of the quantum channel itself needs to be improved. The PNS attack is only possible because light is sent across the quantum communication channel in pulses of multiple photons, rather than in emissions of a single photon at a time. Research into improving the ability to control photon emissions to the single photon is incredibly important, as it will guarantee the security of the quantum communication channel.

9.3 The Scalability Problem

Even supposing that the detection mechanisms used in the hardware were perfect and that single photons could be sent instead of photon pulses, there is still a problem of scalability. In order for a quantum channel to communicate quantum information successfully, it must be the case that no part of the transportation medium observes or interacts with the qubits being transported. In practice, the broadcast medium for quantum information is usually fiber-optic cable, but these fiber-optic cables still dissipate information over long enough distances.

Dissipation of signal strength is a well known issue in the field of networks. In classical computer networks, they resolve this loss of signal strength through the use of signal amplifiers. These signal amplifiers cannot be used in a QKD system, as they necessarily involve observing the quantum information being transported. This mere act of observation destroys the integrity of the quantum channel. Thus, dissipation is still an unsolved problem on quantum channels.

Long-distance quantum channels will not help to secure the information sent through the protocols by the machines any further, but they will be necessary to assure the consistency of information access. This is a requirement for information to be available, and QKD technology will not become widespread if it cannot be utilized over long distances. There is much work being done on expanding the distance range of quantum channels, but this research should be promoted far more than it already is.

10 Risks Which Cannot be Mitigated

No matter how secure an encryption system, encryption only provides a mathematical guarantee for the security of secret messages. Information security as a whole requires more than the security of messages that encryption can provide. Full information security is multifaceted, and requires securing both the physical locations that the messages are transported through and the social systems surrounding who should be able to read each message. The following physical and social factors are critical to information security, but simply cannot be solved with encryption.

10.1 Denial of Service

Because there is no known way to transfer qubits wirelessly without observing them, it follows that these qubits must travel through physical wires such as fiber-optic cables. Issuing a Denial of Service attack, then, would be as simple as severing the connection of the physical cable. Though this risk

can lessened somewhat by establishing networks of fiber-optic cables with built-in redundancies, the fact remains that a DoS attack will remain as simple as cutting enough cords.[4]

10.2 Man In the Middle

Though these protocols ensure detection of an eavesdropper on a private conversation, they make no guarantees about whether the recipient of the information is who they claim to be. Thus, it is still possible for Eve to pose as the intended recipient of Alice's message. Eve can then forward this message to Bob, who was the actual intended recipient. Eve now possesses the information that was supposed to be secret between Alice and Bob. Unfortunately, even with all the new mechanisms that QKD offers, there is not yet any means to prevent such an attack. This is a social engineering problem, and out of the scope of an encryption system.

11 Conclusion

QKD may seem like an impossible promise of perfectly secure secrets, but the proof of security on an abstract machine is airtight. Unfortunately, due to current difficulties with implementing QKD systems, it is very easy to dismiss them as being too far from market, too expensive to build, or too infeasible for wide adoption. With their current capabilities and designs, the physical implementations of QKD systems are still insecure due to vulnerabilities on the hardware level.

Though these difficulties certainly exist, they do not imply that there is something fundamentally wrong about the algorithms and protocols. They only show that our current physical capabilities at implementing those protocols on hardware are limited. These difficulties should not be a premise for discounting the possibility of secure QKD systems. Rather, they should impel further research and funding into uncovering a resolution for the practical issues plaguing physical QKD networks. This research will take massive amounts of time and money, and it will require a multi-disciplinary effort from the security community, the quantum physics community, and the computer engineering community. Nonetheless, the promise of perfectly secure encryption - and, by extension, perfectly secure communication - is absolutely worthy of such an enormous investment of resources.

[4] This should not be viewed as a failing of QKD specifically, as the same criticism also applies to classical networks.

References

[1] A. Zeilinger, "Long-distance quantum cryptography with entangled photons," *Quantum Communications Realized*, Sep 2007.

[2] P. Pajic, "Quantum cryptography," 2013.

[3] A. K. Ekert, "Quantum cryptography based on bell's theorem," *Physical Review Letters*, vol. 67, p. 661–663, May 1991.

[4] C. H. Bennett and G. Brassard, *Proceedings of IEEE International Conference on Computers, Systems, and Signal Processing, Bangalore, India*, p. 175–179.

[5] R. Aggarwal, H. Sharma, and D. Gupta, "Analysis of various attacks over bb84 quantum key distribution protocol," *International Journal of Computer Applications*, vol. 20, no. 8, p. 28–31, 2011.

[6] I. Gerhardt and V. Makarov, "How we eavesdropped 100% of a quantum cryptographic key," Dec 2016.