Quantum Key Distribution Literature Review

Zican Wang

December 2021

Abstract

In this paper, we look at how each stage of quantum cryptography utilizes mathematical principles to achieve security. Concretely, let us investigate a particular protocol of quantum key distribution (QKD) called BB84 followed by error correction, privacy amplification alongside some other helpful quantum properties that are helpful for quantum-safe cryptography. We will see, by using equations and tables, why the QKD protocol is more secure. We will also discuss some variants of the protocols and their implementation in the real world. In the end, the limitations and future developments will be considered.

1 Introduction

Our effort of guarding a message so that only entrusted parties can know the true meaning can be dated back to ancient Egypt, where we have seen messages carved on wood being covered by wax[1]. Now, with online communication expanding, we developed new and advanced encrypting and decrypting algorithms as the need for secrecy surged.

We begin with going through some new representations that are not used in classical computation, and some underlying rules in quantum cryptography that do not appear obvious and trivial.

For most encrypting schemes, another string of values is used accordingly to turn the plaintext(the original message) into the cyphertext(the encrypted message). These are called keys. People usually work with different ways to generate, distribute and use keys to achieve corresponding protocols. Section 3 talks about how a type of quantum key distribution can achieve better security than the traditional counterparts even when attacked by someone with unlimited computing power. The QKD process will leave both entrusted parties the same key in principle if there exist no eavesdroppers. Then they can use the keys generated as one time pads to encrypt their messages.

However, due to the noise and inaccuracy generated when implementing the QKD scheme in the real world, and some possible alterations caused by active eavesdropping, the relevant parties would end up with a certain key that is partially the same instead. To get the same key without giving away too much information to the public or the attacker, we can use an error correction algorithm discussed in section 4 following the key distribution process.

Now that the entrusting parties have the same keys, we should still not use the key in encryption straight away yet, since we have not eliminated the eavesdropper's knowledge of the key. An eavesdropper can pick up some information about the key during both the distribution and the error correction phase. Achieving better security means we need to dispose of the parts of the message where the eavesdropper has information.

For convenience reasons and by convention, we call the party which wants to send information Alice and call the party which wants to receive Bob. The eavesdropper or the attacker is called Eve.

2 Preliminaries

2.1 Representation

In classical computation, we use strings of 0's and 1's to represent information. The 0's and 1's (computational basis) can be one of many bases in quantum computation and information. 0 and 1 can be represented by vectors, and so can all other strings generated from the computational basis. The unit of computation in quantum is called a

'qubit' in contrast to a 'bit' in classical computation. The vector represents the state of a qubit, which corresponds to binary representing the state of a bit. All unit vectors can be qubits, which means that vectors representing qubits have more than one non-zero element, in contrast to the bits in the computational basis.

$$0 \to \begin{bmatrix}1\\0\end{bmatrix}, 1 \to \begin{bmatrix}0\\1\end{bmatrix}, x \to \begin{bmatrix}0\\0\\\vdots\\1\\\vdots\end{bmatrix} \begin{matrix}\text{'...00'}\\\text{'...01'}\\\vdots\\\text{'x in binary'}\\\vdots\end{matrix}$$

Thus the computational basis can be seen as a vector span with 0 being its horizontal vector and 1 being its vertical vector. Other bases form spans that are to some angle of the computational basis (Figure 1). All unit vectors (qubits) trace out a circle indicated in orange.

Figure 1: Different bases

The vector notation of bits can be abbreviated by Dirac notation, this is simply adding '|' and '>' around the bit value. Adding '<' and '|' means the conjugate transpose of the column vector the bit value represents.

$$|\psi\rangle = \begin{bmatrix}\alpha\\\beta\end{bmatrix}, \langle\psi| = \begin{bmatrix}\alpha^*, \beta^*\end{bmatrix}$$

This way, superpositions can be easily expressed using additions and subtractions of vectors. The concatenations of bits are represented by '$\otimes$' named 'tensor products' and sometimes it can be omitted, for example, $|10\rangle = |1\rangle \otimes |0\rangle = |1\rangle\,|0\rangle$.

2.2 Projection

Here, we can understand projection operators P onto a basis vector x used on a vector v as the length of v in the x direction. Figure 2 shows an example of a projection.

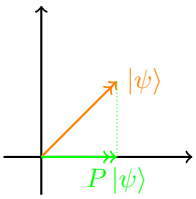

Figure 2: Projection onto the computational basis

We can easily conclude that for a vector u, applying a projection onto the vector orthogonal to u (or $u^{\perp}$) would yield 0 and applying a projection onto u itself would yield 1.

2.3 Entropy

Two types of entropy definitions are used in this paper, the Shannon entropy and the Rényi entropy.

The Shannon entropy is used here to determine the uncertainty, measured in bits, one has regarding a set of strings X[2]. It is defined as:

$$H(X) = -\sum_{x \in X} p(x) \log_2(p(x)) \tag{1}$$

where p(x) is the probability distribution of each string x in the set X. Here we always use capital letters for sets and lowercase letters for elements in a set.

The Rényi entropy of a set of strings X is and can be represented by:

$$R(X) = -\log_2 \sum_{x \in X} (p(x))^2 \tag{2}$$

The Shannon entropy can be treated as the expected value of $-\log(X)$ whereas the Rényi entropy represents $-\log$ of the expected value of X. Thus, the Rényi entropy is always positive and upper bounded by the Shannon entropy of the same object. This is because $\mathbb{E}(-\log(X)) \geq -\log(\mathbb{E}(X))$.

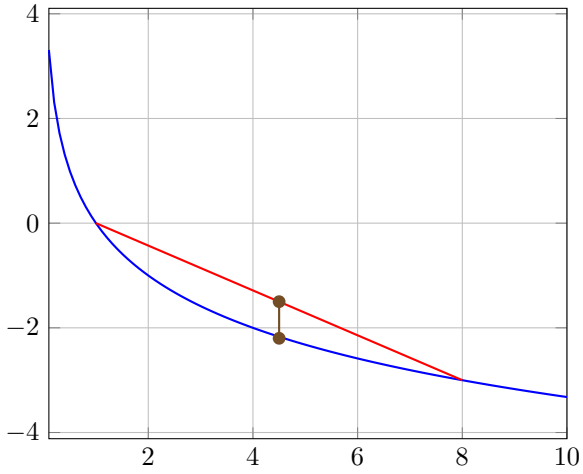

Figure 3: Inequality of $\mathbb{E}(-\log(X)) \geq -\log(\mathbb{E}(X))$

Figure 3 shows $-\log(X)$ in blue. The expected value of a range of numbers can be shown as the middle point of that range. In this example, we have a set of numbers ranged from [1,8], $-\log(\mathbb{E}(X))$ is the point on the blue curve and $\mathbb{E}(-\log(X))$ is the point on the red line which is higher than $-\log(\mathbb{E}(X))$. In fact, since the curve is convex, we can conclude that $\mathbb{E}(-\log(X)) \geq -\log(\mathbb{E}(X))$ is always true.

The conditional Rényi entropy, is the expected value of the Rényi entropy of a distribution of the given condition set Y.

$$\begin{aligned} R(X|Y) &= \sum_y p(y) R(X|Y=y) \\ &= -\sum_y p(y) \log_2 \sum_{x \in X} (p(x|y))^2 \end{aligned} \tag{3}$$

The Rényi entropy definition is mainly used in this paper to describe Eve's ignorance of the shared key between Alice and Bob. For example, if Eve knows the third bit of the string is a '0', all strings with a '1' on the third bit would have a probability of 0 for Eve. Both definitions should be viewed as representing expectations, not absolute measures[2].

3 BB84

The idea of using quantum properties in safe key distribution protocols was introduced by Charles H. Bennett and Gilles Brassard in 1984[3], therefore the name BB84. The purpose of the BB84 protocol is to have a channel

explicitly for distributing keys where eavesdropping of more than a certain amount can be detected. Discarding the messages that might be eavesdropped on, we can then theoretically generate secret keys that are absolute-random to the attackers, making their probability of getting the correct plain-text randomly distributed. The main idea of the BB84 protocol is to use two sets of non-orthogonal bases to send the encryption keys from Alice to Bob so that Eve will know little about it. We look at the details in the following subsections.

3.1 Quantum measurements

In contrast to measurements in the classical physic world, where it does not have a noticeable effect on the object being measured, quantum measurements can change the state of the quantum particle or qubit[4].

In brief, the original state of some qubits can be super-positions – each string has a probability of being the measurement outcome, shown by the square of the corresponding element in the state vector. Thus, the probability of getting a string x (computational basis) when measuring ψ is calculated by $\langle x| \times |\psi\rangle$ or $\langle x|\psi\rangle$ in short. The state of those qubits after measuring collapses to the outcome we get with fix probability 1. This is often referred to as the measurement postulate or Born's rule.

For example, the state $|\psi\rangle = \begin{pmatrix} 1/\sqrt{2} \\ 1/\sqrt{2} \end{pmatrix}$, when being measured in the computational basis, has probability $(1/\sqrt{2})^2 =$ 50% of getting the outcome 0 or 1. After the measurement, the state of ψ becomes 1 if the outcome is 1, and becomes 0 otherwise (Figure 4). Also note that $|\psi\rangle$ can be a linear combination of the two basis vectors, which infers that a state vector can be represented by a linear combination of the vectors in another non-orthogonal basis. Thus, linear combinations can be comprehended as superpositions and in this example, $|\psi\rangle$ is a superposition of 0(1,0) and 1(0,1). $|\psi\rangle = (|0\rangle + |1\rangle)/\sqrt{2}$.

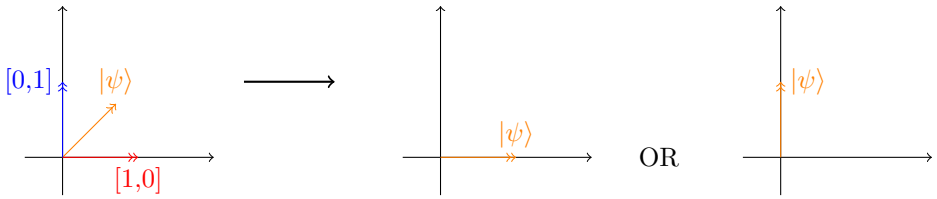

Figure 4: measuring $|\psi\rangle$

From this, we can see that measuring a state using a basis non-orthogonal to the state's basis will give more than one possible outcome and result in a change of the state.

3.2 The protocol

In general, the BB84 protocol involves the sender Alice sending each qubit in a random choice of two non-orthogonal bases to Bob, the receiver, so that the eavesdropper Eve has no knowledge of the basis in use for each particular qubits. Eve then has to 'guess' the basis randomly thus making the outcome of Eve's measurement on the qubits uniformly distributed when Eve guesses the wrong basis, according to the measurement postulate. This not only means Eve has a negligible possibility of gaining all information but also infers that Bob will know if Eve has eavesdropped or not by comparing the states he receives to Alice's since Eve's measuring will alter the qubits' states.

Alice and bob first publicly decide on the two non-orthogonal state they are going to use. Table 1 illustrates the different possible outcomes when measuring a qubit that Alice send, where A and B are two non-orthogonal bases.

Receiving basis	Sending basis and state			
	A,0	A,1	B,0	B,1
A	0	1	0 50% 1 50%	0 50% 1 50%
B	0 50% 1 50%	0 50% 1 50%	0	1

Table 1: Sending and receiving for a single qubit

The following table 2 shows examples of the steps followed by Alice and Bob for the BB84 protocol.

Alice's string	0	1	1	0	1	1	0	0	...	0	1	1	0	1
Alice's randomly chosen sending basis	B	A	B	B	A	A	B	A	...	B	A	A	B	A
Bob's randomly chosen receiving basis	A	A	B	A	A	B	A	A	...	B	B	A	B	A
Bob gets his version of Alice's string	1*	1	1	0*	1	0*	0*	0	...	0	0*	1	0	1
Bob checks Alice's basis	✗	✓	✓	✗	✓	✗	✗	✓	...	✓	✗	✓	✓	✓
Bob discards the ones measured in the wrong basis		1	1		1			0	...	0		1	0	1

Table 2: A specific example of the BB84 protocol

Using techniques developed from the fact that measuring the state $|\psi\rangle = \begin{pmatrix} 1/\sqrt{2} \\ 1/\sqrt{2} \end{pmatrix}$ results in a uniformly distributed outcome, Alice generates a set of non-pseudo-randomly selected bases[5]. She then uses the set of bases to send the string to Bob. Bob, according to the protocol, comes up with his set of choices of the two bases and measures the incoming qubit string. After that, Alice and Bob check their bases publicly and discard the ones with different sending and receiving bases. The string of bits they have at the end of this process is called the sifted key. They would then check some bits from the sifted key publicly. If the majority of the bits are matched, there would likely be no eavesdroppers since an eavesdropping activity would alter the bits, and Alice and Bob would discard the checked bits in the end. If most of the bits are different, they would start a new session because more different bits indicates a higher chance of an adversary[3].

Assuming Eve wants to listen in on Alice's and Bob's conversation, she can at best have 1/2 probability of guessing the correct basis that Alice used. Notice that for each bit, Bob also have a 1/2 chance of getting the correct basis. This means that each bit has a probability of 1/2 of being discarded. In total, Eve would have a 1/4 probability of getting one bit that Alice and Bob would eventually use. To have a key that is perfectly random to Eve, we need to privacy amplification on the keys Alice and Bob have in section 5. To ensure that Alice and Bob have the exact same key to do privacy amplification on, they can use the error correction algorithm covered in section 4.

However, Eve may want to get a copy of the qubits sent by Alice and give the original ones to Bob, thinking this might cover up her eavesdropping behaviours. She would then measure the qubits after Alice publishes her bases. Fortunately, another quantum principle prevents this from happening.

3.3 The no cloning theorem

Cloning at the quantum level, similar to measuring, is different from the classical perspective, where it is easy to achieve (how many times did the word 'Alice' appear by now). In fact, it is proved that we cannot clone a single arbitrary quantum perfectly[6]. We can assume that if there exists a perfect cloning device, it would outcome the following states:

$$|M_s\rangle\,|\psi\rangle \rightarrow |M_f\rangle\,|\psi\psi\rangle \tag{4}$$

Where M_s is the state of the device at the start, and M_f is the state when it finishes. As we have seen from Figure 4, any state A can be a superposition of two vectors X and Y, where X and Y are in a basis non-orthogonal to A's. Therefore for our device, we have:

$$|M_s\rangle\,|\updownarrow\rangle \rightarrow |M_v\rangle\,|\updownarrow\updownarrow\rangle \tag{5}$$

$$|M_s\rangle\,|\leftrightarrow\rangle \rightarrow |M_h\rangle\,|\leftrightarrow\leftrightarrow\rangle \tag{6}$$

Where $|\updownarrow\rangle$ and $|\leftrightarrow\rangle$ are the basis vectors that are not orthogonal to $|\psi\rangle$'s so that $|\psi\rangle$ can be a superposition of $|\updownarrow\rangle$ and $|\leftrightarrow\rangle$: $|\psi\rangle = \alpha\,|\updownarrow\rangle + \beta\,|\leftrightarrow\rangle$. M_v and M_h are states of the machine for both basis vectors when the machine finishes, we can assume they are equal to M_f for the best case. Since the state and the basis of the input ψ are to be cloned without measuring, the device would have to use a fixed basis, in this case $|\updownarrow\rangle$ and $|\leftrightarrow\rangle$. Then according to equation 4, we would have:

$$|M_s\rangle\,(\alpha\,|\updownarrow\rangle + \beta\,|\leftrightarrow\rangle) \quad\rightarrow\quad \alpha\,|M_f\rangle\,|\updownarrow\updownarrow\rangle + \beta\,|M_f\rangle\,|\leftrightarrow\leftrightarrow\rangle \tag{7}$$

For the "cloned" qubit remove M_s and M_f, we have:

$$|\psi\rangle = \alpha\,|\updownarrow\rangle + \beta\,|\leftrightarrow\rangle \quad \rightarrow \quad \alpha\,|\updownarrow\updownarrow\rangle + \beta\,|\leftrightarrow\leftrightarrow\rangle \tag{8}$$

However, what we want for the clone is $|\psi\psi\rangle$, which is:

$$\begin{aligned} |\psi\psi\rangle &= (\alpha\,|\updownarrow\rangle + \beta\,|\leftrightarrow\rangle)(\alpha\,|\updownarrow\rangle + \beta\,|\leftrightarrow\rangle) \\ &= \alpha^2\,|\updownarrow\updownarrow\rangle + \alpha\beta\,|\updownarrow\leftrightarrow\rangle + \beta^2\,|\leftrightarrow\leftrightarrow\rangle \end{aligned} \tag{9}$$

Therefore, equation 8 is not an actual clone of the state $|\psi\rangle$, which means that no device would be able to clone an arbitrary qubit, thus rendering Eves resolve non-practical.

3.4 Variants

After publishing the BB84 protocol, Charles H.Bennett came up with more QKD protocols with the same principle but in mutated form, for example, a QDK system using only two non-orthogonal states[7] in contrast to table 1 (4 different combinations of bases and states).

Receiving basis	Sending state	
	$\|u_0\rangle$	$\|u_1\rangle$
P_0	$\|u_0\rangle$ $p1\%$ $\|u_1\rangle$ $p2\%$	$\|u_0\rangle$ $p3\%$ $\|u_1\rangle$ 0%
P_1	$\|u_0\rangle$ 0% $\|u_1\rangle$ $p4\%$	$\|u_0\rangle$ $p5\%$ $\|u_1\rangle$ $p6\%$

Table 3: Sending and receiving for using two non-orthogonal states

In table 3, $|u_0\rangle$ and $|u_1\rangle$ are two non-orthogonal states to be sent by Alice. They can represent 0 and 1 according to Alice's choice, P_0 and P_1 represent the projections onto the vector $|u_1^{\perp}\rangle$ and $|u_0^{\perp}\rangle$ respectively. According to section 2.2, P_0 will eliminate $|u_1\rangle$ and P_1 will eliminate $|u_0\rangle$. As a result, Bob will be sure that Alice sends a $|u_0\rangle$ when he measures and gets $|u_1\rangle$ using P_0, and vice versa for P_1.

There are other variants such as SARG[5] and so on. Different protocols may have a variance in efficiency when implemented.

3.5 Limitations

Quantum key distribution protocols can prevent eavesdropping effectively, to our current understanding, from adversaries with any amount of computational power. Indeed, the powerful protocols generate keys that can be safely used, and yet it still has limitations and needs follow-up procedures to ensure safer communication.

Firstly, for BB84 itself as proposed in the original paper[3], it does not promise a 0% rate of Eve getting any information from the channels, as explained in section 3.2. In addition, other aspects like radiation are also factors that can alter the bit values Alice sends, making the keys Alice and Bob hold at the end of the protocol slightly different. Thus, post-processing algorithms like error correction and privacy amplification are needed to make the protocol sufficient for practical use.

Secondly, other aspects of security are also not taken into account in QKD. For example, man-in-the-middle attacks and denial of service attacks(DOS). For preventing man-in-the-middle attacks, we can use a short amount of the secret key for each session as authentication keys[8]. However, few existing solutions were published for proofing denial of service attacks besides QKD networks(Section 6). This is because, as we have seen before, DOS attacks can be applied by simply eavesdropping actively or blocking the communication channels in implementation (mostly optic fibres). In addition, local machines are not protected by QKD protocols, so attacks on them can still be a threat. These are problems that we suffer from with our current major encryption schemes, so QKD is by no means a step back.

Last but not least, as we will soon cover in later sections, the whole QKD scheme, including the post-processing part, involves Alice and Bob dropping out a significant proportion of bits to get the final secret key. Although this can be counteracted by the speed of transmitting each qubit we can achieve, it drops the efficiency of key distributions.

4 Error Correction

The error correction after the key distribution is called reconciliation. It was first proposed by G. Brassard and L. Salvail in 1994[9]. It detects and corrects, with high probability, the difference between Alice's and Bob's strings which are created by various causes[10].

First, using functions such as hash functions, Alice sends a hashed string to Bob. Bob then compares his hashed string using the same functions with the one sent by Alice. They would say there is an error when two hashed strings differ. Typically, the strings differ, so Alice and Bob would need a reconciliation scheme.

The scheme, in simple words, depends on Bob having a string that is very similar to Alice's as a result of the QKD protocol. Alice, choosing a function g that is $universla_2$[10], encode her string x with the chosen function. With g(x) and Bob's string y that's similar to x, he has a high probability of retrieving the correct x that Alice has, from a set of possible strings X. The attacker would have a negligible chance of getting x.

Bob, after Alice publishes the encoded string, now knows g(x) in addition to y. Initially, he would already have a set of strings from X compatible with his y. Now Bob also knows the strings compatible with g(x). He then utilizes the intersection of the strings compatible with both y and g(x) to deduce Alice's string x. Doing so, he has a much better advantage than Eve with only g(x).

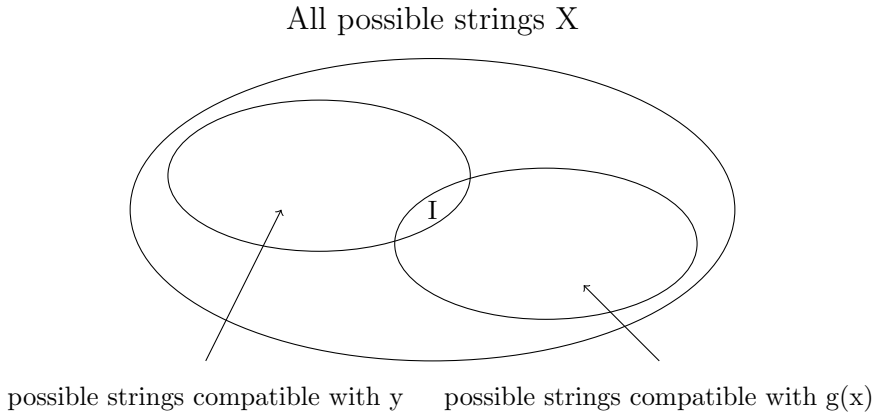

Figure 5: Bob's and Eve's set of possible strings from g(X)

In figure 5, the intersection I is what Bob would know given y and g(x). g(x) is published in the classical channel, so there is a chance that the attacker would gain information about g(x). Alice needs to choose a random g with the size big enough to be random for Eve, and small enough to have a small intersection for Bob to retrieve Alice's string. Bob and Alice should check their string using the detection function and repeat until they have the same string.

5 Privacy Amplification

After Alice and Bob have enough confidence that they have the same string, they would want some algorithm to eliminate or reduce most of Eve's information about the shared string. This is where privacy amplification comes in. Published in 1988[11], this algorithm extracts a string with better secrecy from a partially secret longer string.

Alice and Bob now have the same string W, with n bits, Eve has a string V that contains t bits of information from W and Eve has a ignorance of R(W) for the string W. The algorithm g takes in the n bits W and outputs an r bit string Q with no correlated information with V. g is chosen from a set of functions G, and we need to consider the set G to be known by the public (especially by the attacker). We use Rényi's entropy for calculating Eve's information for privacy amplification because it tends to its limit, Shannon's entropy, on independently and identically distributed states. Therefore, Eve's ignorance of Q given she knows the set G is $R(Q|G)$ which is the expected value of Eve knowing each g in the set G, therefore, according to equation 3:

$$
\begin{aligned}
R(Q|G) &= \sum_g p(g) \left(-\log_2 \sum_q (p(q|g))^2 \right) \\
&\geq -log_2 \left(\sum_g p(g) \sum_q (p(q|g))^2 \right)
\end{aligned}
\tag{10}
$$

where the inequality is derived from the inequation from section 2.3. Considering the part within the $-\log_2$, we have:

$$
\begin{aligned}
\sum_g p(g) \sum_q (p(q|g))^2 &= \sum_g p(g) \sum_q \left(\sum_{x:g(x)=q} p(x) \right)^2 \\
&= \sum_g p(g) \left(\sum_{x:g(x)=q} p(x) \right)^2
\end{aligned}
\tag{11}
$$

The set of functions G needs to satisfy one property, which is that the probability of $G(x_1) = G(x_2)$ is smaller than $1/2^r$ where r is the length of the output. This is the $universal_2$ mentioned in section 4. Thus, from equation 11, we get:

$$
\begin{aligned}
\sum_g p(g) \left(\sum_{x:g(x)=q} p(x) \right)^2 &= \sum_g p(g) \left(\sum_{x_1:g(x_1)=q} p(x_1) \right) \left(\sum_{x_2:g(x_2)=q} p(x_2) \right) \\
&= \sum_g p(g) \left(\sum_{x_1,x_2:g(x_1)=g(x_2)} p(x_1)p(x_2) \right) \\
&= \sum_g p(g) \sum_{x_1} p(x_1) \sum_{x_2} p(x_2) \delta_{g(x_2)}^{g(x_1)} \\
&= \sum_{x_1} p(x_1) \sum_{x_2} p(x_2) \text{prob}\{G(x_1) = G(x_2)\} \\
&\leq \sum_{x_1=x_2} p(x_1)^2 + \sum_{x_1 \neq x_2} p(x_1)p(x_2) \times 2^{-r}
\end{aligned}
\tag{12}
$$

From equation 2 we get $\sum_x p(x)^2 = 2^{-R(x)}$. This can help formulate the first half of the summation. We also know that $p(x_1 \neq x_2) = 1 - p(x_1 = x_2)$ which can be used for the second half to derive:

$$
\begin{aligned}
\sum_g p(g) \sum_{q|G=g} (p(q|g))^2 &\leq 2^{-R(x)} + 2^{-r} \left(1 - \sum_{x_1} p(x_1)^2 \right) \\
&= 2^{-R(x)} + 2^{-r}(1 - 2^{-R(x)}) \\
&= 2^{-R(x)} + 2^{-r} - 2^{-r-R(x)}
\end{aligned}
\tag{13}
$$

Looking back at $R(Q|G)$, using the inequalities we derived, we can get:

$$
\begin{aligned}
R(Q|G) &\geq -\log_2(2^{-R(x)} + 2^{-r} - 2^{-r-R(x)}) \\
&= -\log_2(2^{-r} \times (2^{r-R(x)} + 1 - 2^{-R(x)})) \\
&= r - \log_2(1 + 2^{r-R(x)})
\end{aligned}
\tag{14}
$$

Here, we dispose the last term $-2^{-R(x)}$ since it is negligible with a long string $x = W$ for Alice and Bob. Remember r is the length of the output Q and in the privacy amplification algorithm, we set $r = R(W) - d$, where d is a small

constant. Thus, Eve's knowledge of the final secret key Q given that she already knew the set of functions G would be:

$$R(Q|G) \geq R(W) - d - \log_2(1 + 2^{-d}) \tag{15}$$

The last term is ignored with order $O(2^{-d})$. This shows that Eves' ignorance of the secret key Q counted in bits would be at least $R(W) - d$, therefore it loses at most d bits of randomness compared to the original $R(W)$.

To recap, when doing privacy amplification, Alice and Bob would first choose one function from a set of universal functions. They then decide on the length r of the shrunken key Q, according to the bits of randomness they have to lose d where the lower bound of Eve's ignorance is $R(W) - d$.

We can see that this heavily depends on Eve's original ignorance $R(W)$, which means it would be ineffective when used after protocols other than QKD. The fact that QDK ensures, with extremely high probability, that Eve could only obtain a limited portion of the string without being detected by Alice and Bob is the core reason for using privacy amplification.

6 Implementation

Although the BB84 protocol is not the most efficient one on paper, it is easy to implement. The IDQ has adopted both BB84 and a protocol called SARG for their QDK scheme[5]. The privacy amplification algorithm we covered is what we are using for implementation. In 1989, Bennett and Brassard performed the first experimental QDk demonstration, where the key travelled an astonishing 30 cm in air. People then achieved farther distance using optical fibre and photons as qubits, yet for our current technology, the loss of photons increases significantly when the distance reached a certain threshold. Loss of photons would result in a loss of efficiency of key exchange. With our latest limit of 300km[12], quantum repeaters are used to build QKD networks[13]. Another approach for increasing the distance is to use satellites to convey messages as photons, since the absorption gets low in space[5]. The satellite receives the key when it travels above the first station, and transmits the key when it is above the other station. In 2016, China launched the first QKD satellite to perform this "pick-up" and "drop" technique along with other protocols[14]. These approaches are in an experimental stage where new improvements are being evaluated. We expect to see breakthroughs and commercial uses in the near future.

7 Conclusion

In section 3, we showed that the BB84 protocol achieves security at a higher level than our current major encryption scheme as BB84 detects eavesdropping. Our current scheme, the public key cryptography scheme is not as secure, since reversing a one-way function has not been proven impossible. The public key cryptography schemes depend on us having limited resources. New Mathematical discoveries, computers that are more powerful, and quantum computers will render public key cryptography obsolete while QKD schemes, which are quantum-safe, still stand firm.

We also looked at the post-processing algorithms for QKD schemes, which improve the outcome of QKD by matching the keys for both parties and reducing the attacker's information on the key.

Quantum systems, although mathematically rigid, currently lacks standardization[15] and implementation. For implementation, one big challenge, for now, is building quantum networks. These networks not only extend the QDK systems' distance but is also able to counteract denial-of-service attacks(3.5). And these networks are built with quantum repeaters. Quantum repeaters, as we mentioned in section 6, need to work as multiple Alices and Bobs, which means that they also need to be trusted and withstand attacks. A new concept of quantum repeaters, which transfer qubits without measuring and altering the photon, was proposed. These repeaters need to be quantum computers[5], which can decipher public key encryption with ease. Fortunately, and unfortunately, our technology has not achieved such a level at which quantum networks can be built.

Quantum information and quantum cryptography are existing field where we finally have the material and technology to build the hardware for. With more funding invested and more people participating, new discoveries and more practical uses will be developed in a few years.

References

[1] Simon Singh. *The code book*, volume 7. Doubleday New York, 1999.

[2] John Watrous. *The theory of quantum information*. Cambridge university press, 2018.

[3] Charles H Bennett and Gilles Brassard. Quantum cryptography: Public key distribution and coin tossing. *arXiv preprint arXiv:2003.06557*, 2020.

[4] Daniel E Platt. A modern analysis of the stern–gerlach experiment. *American Journal of Physics*, 60(4):306–308, 1992.

[5] ID QUANTIQUE SA. Understanding quantum cryptography, 2020. `https://www.quantumcommshub.net/wp-content/uploads/2020/09/Understanding-Quantum-Cryptography_White-Paper.pdf`, (accessed: 30.12.2021).

[6] William K Wootters and Wojciech H Zurek. A single quantum cannot be cloned. *Nature*, 299(5886):802–803, 1982.

[7] Charles H Bennett. Quantum cryptography using any two nonorthogonal states. *Physical review letters*, 68(21):3121, 1992.

[8] Jörgen Cederlöf. Authentication in quantum key growing, 2005.

[9] Tor Helleseth. *Advances in Cryptology–EUROCRYPT'93: Workshop on the Theory and Application of Cryptographic Techniques Lofthus, Norway, May 23–27, 1993 Proceedings*, volume 765. Springer, 2003.

[10] Charles H Bennett, Gilles Brassard, and Jean-Marc Robert. Privacy amplification by public discussion. *SIAM journal on Computing*, 17(2):210–229, 1988.

[11] Charles H Bennett, Gilles Brassard, Claude Crépeau, and Ueli M Maurer. Generalized privacy amplification. *IEEE Transactions on Information theory*, 41(6):1915–1923, 1995.

[12] Boris Korzh, Charles Ci Wen Lim, Raphael Houlmann, Nicolas Gisin, Ming Jun Li, Daniel Nolan, Bruno Sanguinetti, Rob Thew, and Hugo Zbinden. Provably secure and practical quantum key distribution over 307 km of optical fibre. *Nature Photonics*, 9(3):163–168, 2015.

[13] Louis Salvail, Momtchil Peev, Eleni Diamanti, Romain Alléaume, Norbert Lütkenhaus, and Thomas Länger. Security of trusted repeater quantum key distribution networks. *Journal of Computer Security*, 18(1):61–87, 2010.

[14] PAN Jianwei. Progress of the quantum experiment science satellite (quess) micius project. *KongJianKeXueXueBao*, 38(5):604–609, 2018.

[15] Thomas Länger and Gaby Lenhart. Standardization of quantum key distribution and the etsi standardization initiative isg-qkd. *New Journal of Physics*, 11(5):055051, 2009.