Aliro™

The Fundamentals of Entanglement-based Key Distribution

Aliro

www.alirotech.com

The Fundamentals of Entanglement-based Key Distribution

Summary

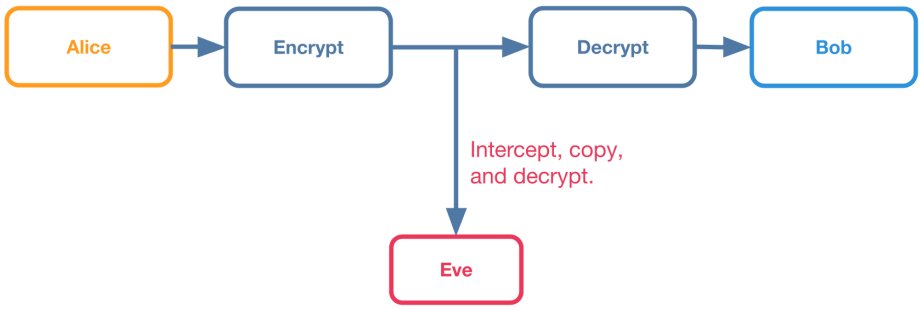
Key distribution protocols are used to securely establish a shared secret key between two parties. This white paper outlines how different key distribution protocols are implemented and scaled. As an example we compare the entanglement-based BBM92 protocol with the prepare-and-measure BB84 protocol. We discuss how the theoretical security of these two protocols is equivalent, and how they differ in implementation. Finally, we discuss how QKD can be deployed and scaled today using trusted relay nodes and why entanglement-based quantum networks that apply quantum repeaters offer a more secure solution at scale.

Introduction

Securing network communications is critical to our information technology infrastructure. Secure communication between two parties, Alice and Bob, is performed in three basic steps:

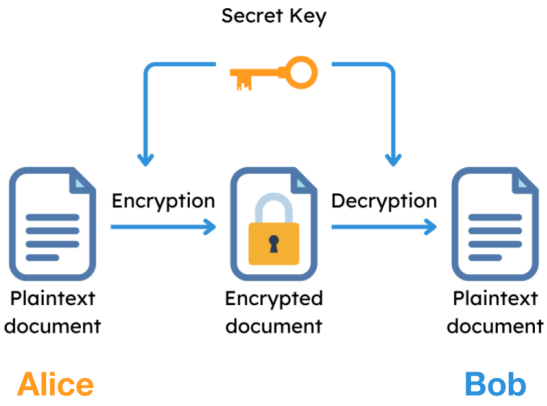
1. Alice encrypts a message.
2. Alice sends Bob the encrypted message via a public communication channel.
3. Bob decrypts the message in order to read it.

Since Alice's message is sent through a public communication channel, it could be intercepted by an eavesdropper, who we refer to as Eve. The encryption is crucial because it scrambles the message preventing its contents from being revealed to Eve.



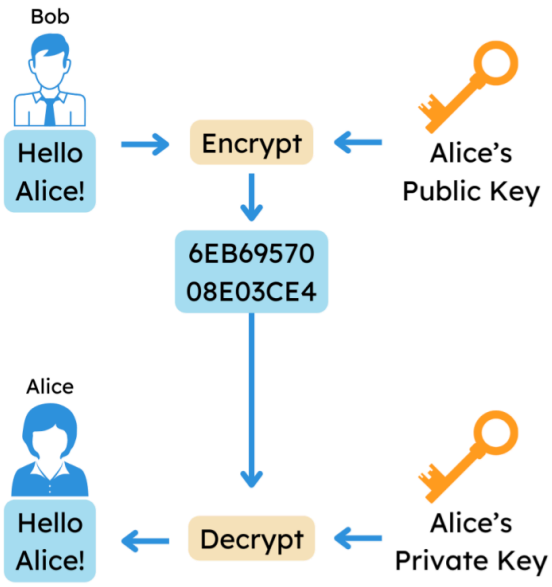
There are two main types of encryption algorithms, symmetric-key encryption and asymmetric-key encryption. Symmetric-key encryption is more efficient than asymmetric-key encryption, but it relies on Alice and Bob sharing a secret key. This presents a challenge in practical networks: how to securely share the key between Alice and Bob. Today's networks commonly apply symmetric-key encryption for bulk data traffic while asymmetric-key encryption is used for authentication and session key agreement. However, future quantum computers will be able to crack these asymmetric key-encryption schemes. New protocols that are resilient to attacks by quantum computers will need to be implemented to secure our data and communications.

Symmetric Key Encryption. Symmetric-key encryption uses a shared secret key to both encrypt and decrypt data. There are many types of symmetric key encryption algorithms such as the one-time pad, or the Advanced Encryption Standard (AES).



Despite its touted security and efficiency, there is a major problem with symmetric-key encryption. Namely, how do you securely distribute a secret key over a public network? If the key is intercepted during distribution, the security of the encryption is compromised. Today, symmetric-key encryption systems commonly apply classical communication protocols such as Diffie-Helman (DH) or elliptic-curve Diffie-Helman (ECDH) to share secret keys over an insecure channel. However, these key distribution schemes can be cracked by a quantum computer.

Asymmetric Key Encryption. Asymmetric-key encryption, also known as public-key cryptography, involves a pair of keys: a public key, which is shared openly, and a private key, which is kept secret. Anyone can use the public key to encrypt a message, but only the holder of the private key can decrypt it.



Since Alice does not share the private key with anyone an eavesdropper would need a different approach to decrypt the message. Typically, the security of asymmetric-key encryption relies on a math problem that is difficult to solve, but easy to check. Thus, an eavesdropper would just need to solve the math problem to decode the message. For example, the RSA algorithm is a commonly used asymmetric-key encryption method that relies on the difficulty of factoring large integers. However, when quantum computers become available at larger scales, Shor's algorithm can factor these large numbers, enabling eavesdroppers to crack RSA encryption.

Quantum Secure Protocols for Key Encryption. Protocols that leverage quantum physics for key distribution offer a solution for establishing shared secret keys, enabling broader use of symmetric-key encryption for securing bulk network traffic. These protocols help secure networks against the threat of quantum computers, and are an important cryptographic primitive on which entanglement-based quantum networks are built. Entanglement-based networks are multipurpose and hardware agnostic. These networks support the implementation of a wide variety of security measures, including post-quantum cryptography and entanglement-based key distribution protocols such as BBM92 and E91.

Secure networks in an evolving threat landscape

Two main approaches have emerged for securing systems against the impending threat of quantum computers: post-quantum cryptography and protocols for key distribution that leverage quantum mechanics.

Post-quantum cryptography (PQC) is a classical method of asymmetric-key encryption that is quantum resistant. This means that the security of PQC relies on foundational mathematical problems that cannot be solved efficiently using a quantum computer. PQC is favored by organizations such as the National Security Agency (NSA) because these algorithms are compatible with the existing communications and networking infrastructure. Implementing these quantum-resistant algorithms would mainly require a software update, however, it is worth pointing out that PQC requires more memory and computing power than existing approaches. This means that practical PQC will likely require the existing networking hardware to be upgraded. Moreover, the National Institute of Standards and Technology (NIST) is running a dedicated program to identify and evaluate viable algorithms for post-quantum cryptography. For more information on PQC, see the on-demand webinar "The Evolution of Network Security: Bridging Classical and Quantum Systems."

On the other hand, quantum secure protocols for key distribution securely establish a shared secret between two parties, Alice and Bob. The shared secret key can then be used for

symmetric key encryption, enabling secure communications between the two parties. Unlike PQC, whose security relies on the presumed computational difficulty of a math problem, protocols that leverage principles of quantum physics, such as no-cloning or entanglement monogamy, can be used to secure Alice and Bob's communications against eavesdroppers.

It is important to note that protocols for key distribution that leverage quantum physics are compatible with PQC solutions, it does not have to be one or the other. Both cryptographic primitives can be used to secure a diverse set of secure networking applications.

Overview of the key distribution process

The goal of protocols for key distribution is to generate a random secret key that is shared between Alice and Bob. Once the secret key is generated, Alice and Bob can communicate securely using symmetric key encryption, such as the one time pad. To perform a protocol that leverages quantum physics, Alice and Bob communicate over authenticated public channels and make use of a quantum resource, such as quantum communication from Alice to Bob or entanglement shared between Alice and Bob. Although an eavesdropper could intercept the classical message or tamper with the quantum communication resources, these protocols are secure against eavesdroppers due to the underlying quantum physics.

How does this work? Key distribution is essentially a two-step process. In the first step, Alice and Bob use quantum communication or entanglement to generate correlated raw keys that they each keep private. This process involves many rounds of key generation where each round consumes a qubit of communication or an entangled pair shared between Alice and Bob. In general, the generated raw keys are not the same, however some of the bits in Alice and Bob's raw keys are correlated. In the second step, Alice and Bob apply classical processing and communication over public channels to reduce their correlated raw key strings into shared key strings that are secret from any eavesdroppers. To produce shared secret keys, the classical processing applies a series of algorithms that we will discuss in more detail later: key sifting, parameter estimation, information reconciliation, and privacy amplification. Also note that the quantum resources are only used to produce the raw keys. Once the raw key has been generated, Alice and Bob proceed with the same classical processing, regardless of which protocol - BB84 or BBM92 - was used.

For key distribution to be successful, there are a few assumptions that must be true:

1. Alice and Bob can authenticate each other over the public classical channel. Alice can verify that they are talking to Bob and vice versa.
2. The hardware and software that Alice and Bob use must be characterized and trustworthy. Since these devices have access to Alice and Bob's secret keys, compromised hardware could leak secret keys to an adversary.
3. The randomness that Alice and Bob use to produce their raw keys must be private to themselves and uniformly distributed. If an eavesdropper knows Alice's random numbers, then they can learn the key by listening to Alice and Bob's public communication.

Shared Key Generation using Quantum Technology

As we've been discussing, protocols for key distribution can be performed using two different quantum technologies. We refer to key distribution protocols using quantum technologies as follows:

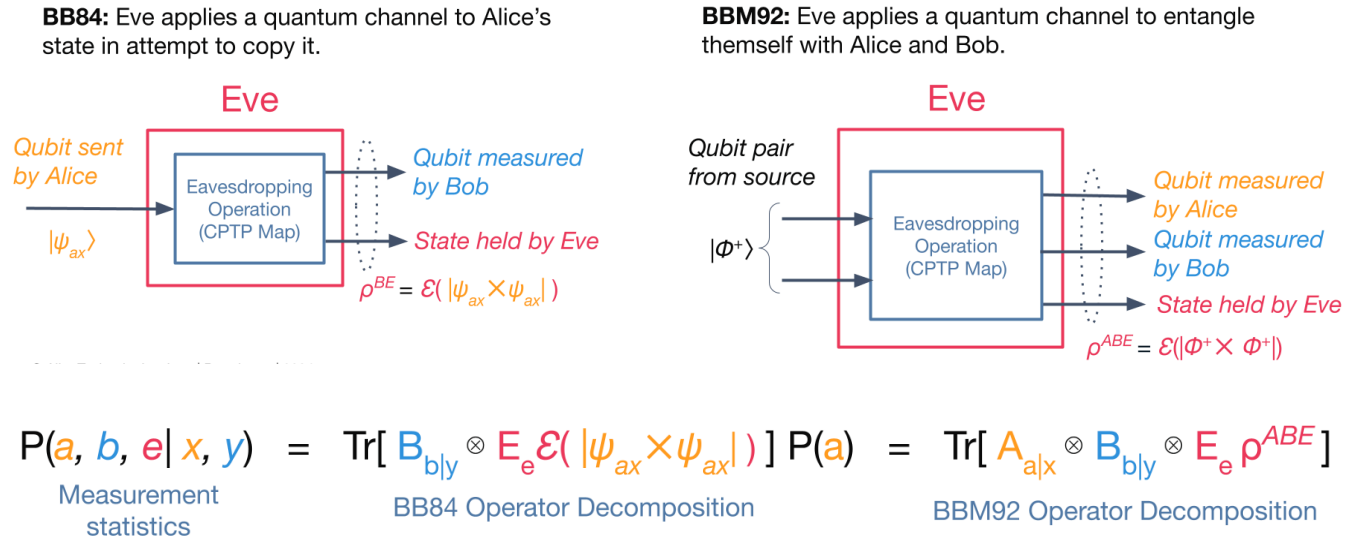
- Quantum Key Distribution (QKD): Alice and Bob generate raw key strings in a setting where Alice prepares a qubit and transmits it to Bob for measurement. This is sometimes referred to as prepare-and-measure key distribution. An example of this type of protocol for key distribution would be BB84.
- Quantum Secure Communication (QSC): Alice and Bob generate raw key strings by leveraging the principles of quantum entanglement. Entangled qubits are distributed to Alice and Bob prior to the key generation process, and when a shared secret key is needed these qubits are measured to create that key. An example of this type of protocol for key distribution would be BBM92.

In both of these cases, the key generation process requires many rounds to produce a key of sufficient length for encrypting messages. In each round, two bits of a correlated raw key are generated while one quantum resource is consumed, either a qubit of communication or an entangled qubit pair. After many rounds take place, a key sifting process is then applied to remove the uncorrelated bits from the raw key. If no errors or eavesdropping occurs then Alice and Bob will share the same bit string after key sifting. Note that key sifting is the same for both the BB84 and BBM92 protocols, but other protocols for key distribution may have different procedures.

Quantum communication and entanglement each serve as quantum resources for generating shared secret keys. Is there any notable difference between these two approaches? As it

turns out, QKD systems are mathematically equivalent to QSC systems and vice versa [Wilde2013]. Thus, as far as quantum theory is concerned, BB84 and BBM92 should not be thought of as separate protocols, but rather as dual protocols that can each be understood in terms of the other. Whether Alice and Bob share entanglement, or have a qubit channel connecting them, doesn't change the theoretical security of either protocol.

In general, the security of both QKD and QSC relies on being able to detect eavesdroppers. Entanglement-based protocols may seem more secure because no quantum information is sent from Alice to Bob, but this is not true. An eavesdropper can just as easily intercept one or both of the entangled qubits as they travel from the source to Alice and Bob allowing eavesdropping to be applied similarly in both QKD and QSC protocols. In the QKD case, eavesdropper detection results from the no-cloning principle of quantum mechanics, which enforces that an eavesdropper cannot copy an unknown qubit without introducing errors into a bit string. In the QSC case, eavesdropper detection relies on the monogamy of entanglement, which enforces that if any eavesdropper is entangled with Alice and Bob, the correlation between Alice and Bob decreases, introducing errors. Unsurprisingly, entanglement monogamy and no-cloning are related physical principles [Leifer2006].



Caption: (Left) An eavesdropping attack on a QKD system. (Right) An eavesdropping attack on an QSC system. Applying channel-state duality, equivalence can be drawn between the measurement statistics of the two key generation approaches. This means that any interception attack on a QKD protocol has an equivalent attack on the QSC protocol and vice-versa.

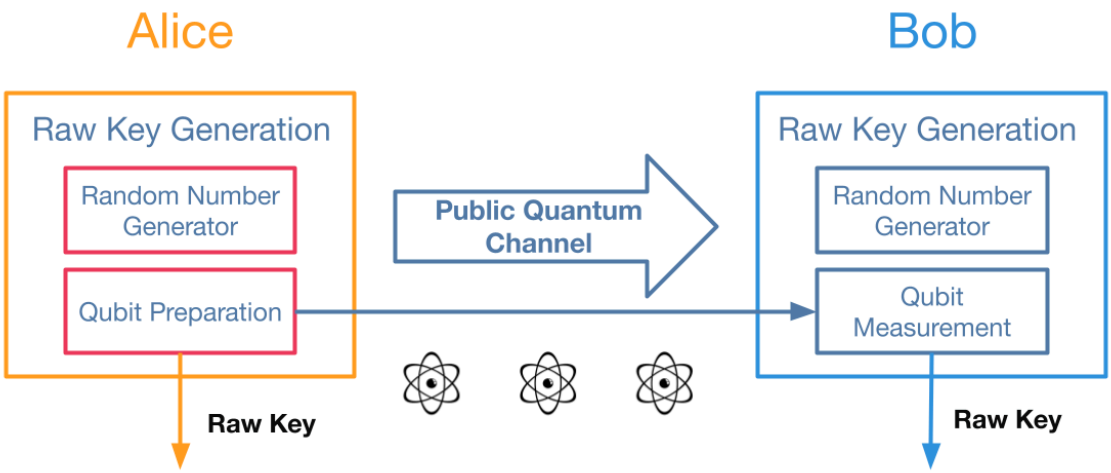
The main difference between these two approaches is their physical implementation. QKD offers higher raw key generation rates than QSC because single photon sources can emit qubits at rates of ~1e9 / sec whereas entanglement sources emit qubit pairs at rates of ~5e6 /

sec. On the other hand, QSC can be simpler to implement because passive basis selection can be incorporated into the quantum measurement. QKD is more complicated because the qubit states must be dynamically encoded at very fast rates, which is a challenging problem in precision control.

One challenge that affects both QKD and QSC is that qubits cannot be sent over long distances due to loss. Two modes of scaling have been developed for these protocols, trusted relay networks and quantum repeater-based networks. Both QKD and QSC are compatible with both modes of scaling, however, quantum repeater-based networks offer improved security over trusted relay networks. We'll discuss how these networks scale key distribution in more detail later.

Quantum Key Distribution (QKD): BB84

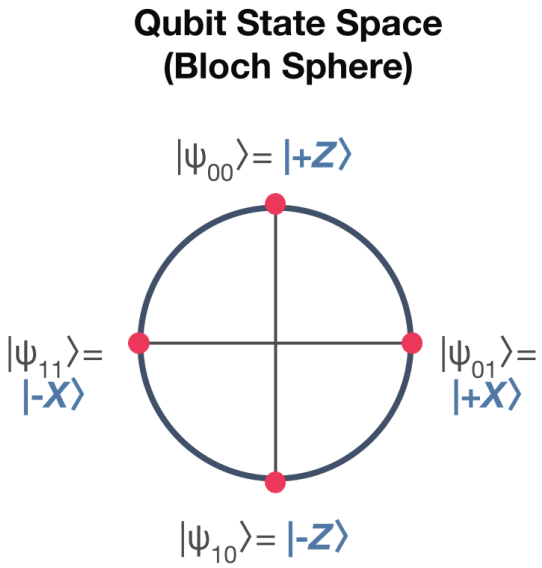
The BB84 protocol, proposed by Bennett and Brassard in 1984 [Bennett1984], leverages principles of quantum mechanics to distribute cryptographic keys between two parties. In this protocol, Alice prepares the key information by encoding it into qubits. These encoded qubits are then sent to Bob who measures the qubits immediately upon receipt. Hence the classification of BB84 as a so-called "prepare-and-measure" QKD protocol.



Caption: Alice and Bob generate correlated bit strings or raw keys using one-way communication over a public quantum channel. Alice prepares a qubit based on two random bits and sends this qubit to Bob, who measures the qubit in a random basis to obtain a one-bit measurement outcome. Alice and Bob's measurement bases and encoded/measured values correspond to the raw key bit string.

In the BB84 protocol, the first step in the key generation process is to prepare or encode the qubit. In this step, Alice generates and records two random bits, which we write as (a,x). Alice uses these two random bits to encode a qubit state that we'll call (Psi).

There are four possible states that Alice encodes in the qubit (Psi), which are conditioned upon Alice's random bit string (*a*,*x*). If Alice draws (0,0), they'll encode the +Z state. If Alice draws (0,1), they'll encode the +X state. If Alice draws (1,0), they'll encode -Z. Finally, if Alice draws (1,1), they'll encode the -X state. Note that the plus or minus corresponds to this value *a,* while the basis (Z or X) corresponds to the value *x*. Hence the bit *x* dictates the basis, while the bit *a* dictates the encoded value.



Caption: Alice's encoded qubit states (red dots) can be visualized as four points distributed equally about the circumference of a circle. The circle represents a two-dimensional slice of the Bloch sphere along the XZ-plane. The vertical axis is the Z basis while the horizontal axis is the X basis. These bases represent two different orientations in which a quantum system can be measured. Once Alice has prepared the qubits, they send the qubits to Bob for the next step: qubit measurement.

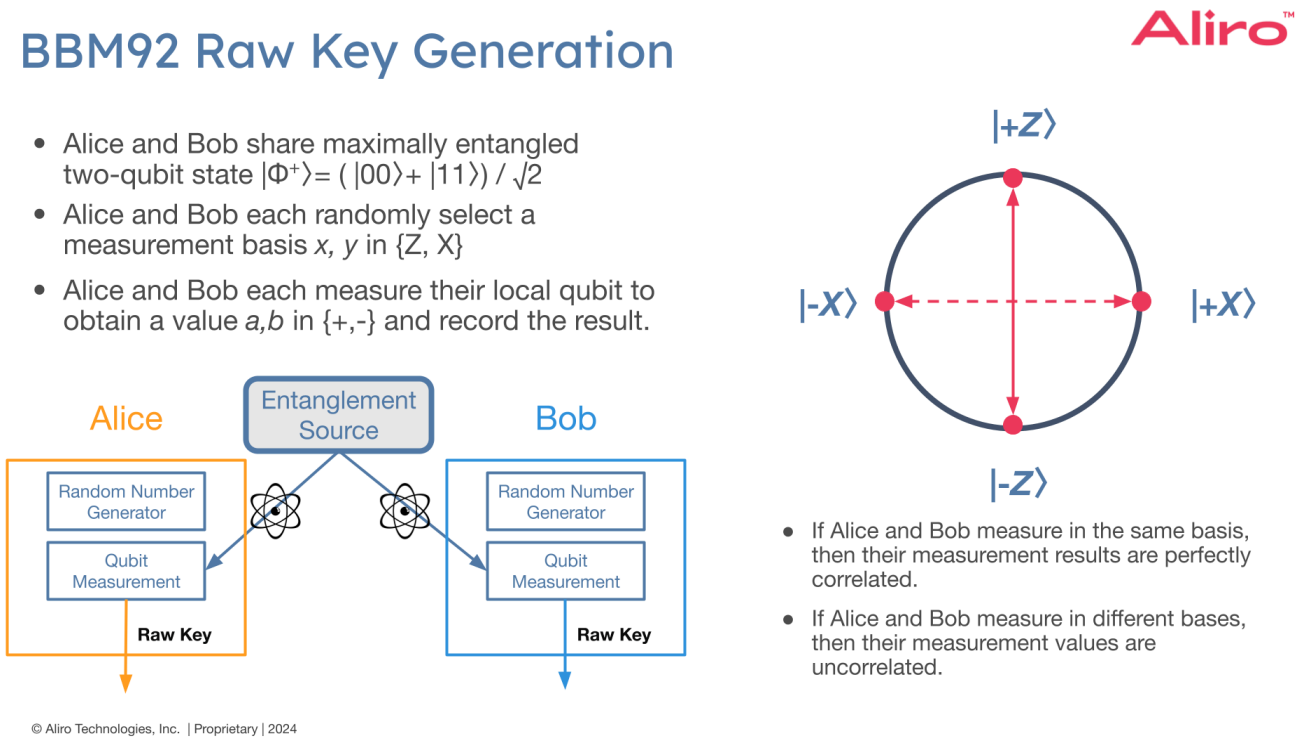
In the second step of the BB84 protocol, Alice's encoded qubit is measured by Bob. In this step, Bob uses a random number generator to produce one random bit, *y*. This random value selects one of the measurement bases, Z or X, where Bob measures the received qubit in this basis to obtain the bit *b* as the measurement outcome. Bob then records the bitstring (*b*,*y*) and notifies Alice that a qubit was measured.

If Bob measures in the same basis that Alice encoded in, then Alice's and Bob's values are perfectly correlated, meaning that their bit strings (*a*,*x*) and (*b*,*y*) are the same. If Bob measures in a different basis than Alice encoded in, then Alice's and Bob's measurement values are uncorrelated, and the resulting value *b* could be + or -, with equal probability. This correlation between matching bases is essential for distilling a shared secret key during the key sifting stage of the classical processing step.

Quantum Secure Communication (QSC): BBM92

The BBM92 protocol, proposed by Bennett, Brassard, and Mermin in 1992 [Bennett1992], leverages principles of quantum entanglement to establish secret keys between two parties. Unlike BB84, Alice and Bob both perform random qubit measurements on a shared entangled state. In BBM92, an entanglement source emits pairs of entangled qubits where each party receives one qubit of the pair. In this case the entanglement serves as a shared resource that Alice and Bob use to produce correlated bit strings. Hence, the classification of BBM92 as an entanglement-based protocol.

Although no information is sent between Alice and Bob during the BBM92 key generation phase, this does not mean that the protocol is secure from eavesdropping. Indeed, the eavesdropper can still attempt to learn the key by entangling their local qubits with the qubits sent to Alice and Bob.



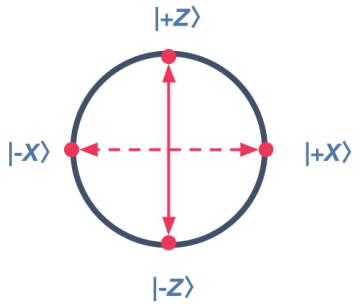
Caption: Alice and Bob share a pair of entangled qubits. They each measure their local qubit in a random basis, X or Z, to obtain the measurement outcome, + or -. These binary values are recorded as the raw key.

To generate keys using the BBM92 protocol, Alice and Bob measure their entangled qubits. As in the BB84 protocol, Alice and Bob each measure in a randomly selected basis, X or Z, and record the measurement result as (a,x) and (b,y) respectively where the bits x and y

respectively encode Alice and Bob's measurement bases, and the bits *a* and *b* respectively encode Alice and Bob's measurement results.

If Alice and Bob each measure their qubit in the same basis, they get the same result with certainty. In other words, their measurement results are perfectly correlated. Otherwise, If Alice and Bob measure in different bases, their measurement results are independent from each other or uncorrelated.

Alice measurement result	+Z	-Z	+X	-X	+Z	-Z	+X	-X
Bob measurement result	+Z	-Z	+Z	+Z	-X	+X	+X	-X
Alice Raw Bits	00	10	01	11	00	10	01	11
Bob Raw Bits	00	10	00	00	11	01	01	11



Caption: Sample data for the BBM92 protocol. Each column corresponds to one round of key generation. The first two rows show Alice and Bob's measurement bases and results while the second two rows show Alice and Bob's raw key strings.

Key Sifting

Alice and Bob's raw keys contain both correlated bits and uncorrelated bits. In each round of key generation, Alice and Bob each produce two bits (a,x) and (b,y) respectively. As was noted earlier, whenever Alice and Bob's bases align (i.e. $x = y$), it follows that $a = b$, hence Alice and Bob's raw keys align for these bits. However, when Alice and Bob's bases are different there is no correlation.

The procedure known as key sifting exploits the correlation between Alice and Bob's raw keys to filter out the uncorrelated bits while leaving the correlated bits. First, Alice uses the classical channel to send Bob the basis x used in each round of key generation. In response, Bob uses the classical channel to send Alice the rounds in which their bases agree (*i.e.,* whether $x = y$). Alice and Bob discard the bits generated in all rounds that do not share the same basis, which roughly amounts to about half of the raw key string (N bits). Furthermore, since the bases were communicated over the public classical channel, an eavesdropper would know these values. As a result, the bits corresponding to the basis x and y must also be discarded for all key generation rounds.

Once the key sifting procedure concludes, Alice and Bob are left with sifted key strings that are about *N/2* bits long where *N* is the number of key generation rounds. Therefore, each use of a quantum resource, entanglement or qubit communication, yields only ½ bit of sifted key on average.

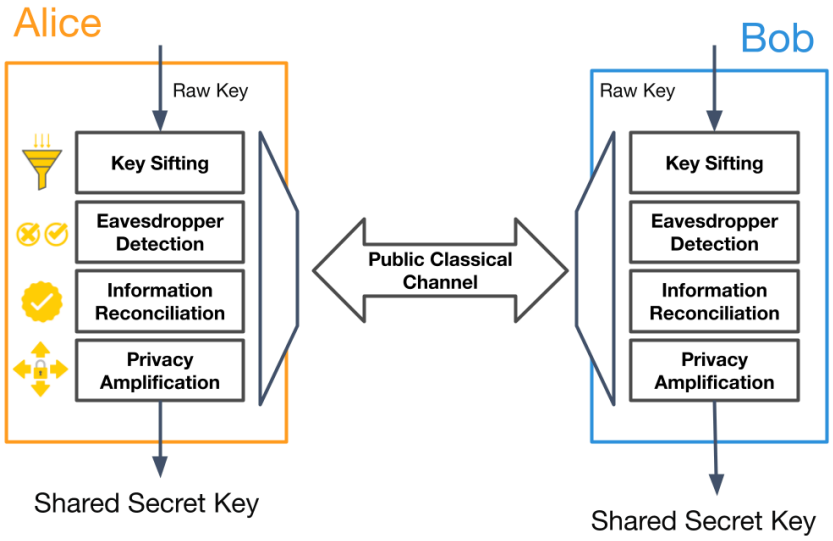
Alice States	-Z	+X	-X	-Z	+X	-X
Bob Measurements	-Z	+Z	+Z	+X	+X	-X
Alice Raw Bits	10	01	11	10	01	11
Bob Raw Bits	10	00	00	01	01	11
Sifted bits	1	reject	reject	reject	0	1

Caption: Key sifting on sample raw keys produced using a BB84 key generation protocol. Note that the table could also apply to BBM92. Each column corresponds to a round of key generation. The green columns designate the key generation rounds that yield correlated raw keys and are identified by key sifting. The top two rows show Alice and Bob's bases and values, the third and fourth row show Alice and Bob's respective raw key strings, and the last row shows the sifted key or if the round was rejected during sifting.

Classical Processing to Establish Shared Secret Keys

After the key generation procedure, Alice and Bob each hold a raw key string containing *2N* bits where two bits are produced in each key generation round. It does not matter whether BBM92 or BB84 were used to generate the raw key, the following classical processing is the same in both cases. Note that this stage of the protocol is entirely classical where quantum physics is only used in the key generation stage.

The classical processing involves four steps: key sifting, parameter estimation, information reconciliation (also referred to as error correction), and privacy amplification. Each of these steps are critical for transforming Alice and Bob's raw keys into a shared secret key suitable for encryption and other cryptographic applications. Hence the goal of these steps is to establish identical keys between Alice and Bob, without sharing any information about the key with an eavesdropper.



Caption: Alice and Bob communicate over a public classical channel to coordinate their processing and operations upon their raw keys.

Eavesdropper detection

After shared key generation, Alice and Bob each hold the same bit string, assuming that no errors occurred. However, how do we know for certain that the sifted key is private between Alice and Bob? Furthermore, what would happen if there were errors in the key generation process?

Suppose an eavesdropper Eve were to listen in on the BB84 protocol by intercepting the qubits sent from Alice to Bob. Ideally, Eve would like to make copies of the intercepted qubit, and then send it to Bob undisturbed for measurement. This qubit-copy attack would allow Eve to decode the basis and value encoded by Alice while giving no indicator that the qubit had been intercepted. Fortunately, this attack is not possible due to the no-cloning principle of quantum mechanics. That is, an unknown qubit cannot be copied, and attempting to do so introduces errors into the shared key with non-negligible error.

As an example, a basic eavesdropping algorithm that Eve could use is to randomly measure Alice's qubit in the X or Z basis to obtain a measurement result (e,z) where bit e is the measurement outcome and bit z indexes the basis. However, Eve's measurement consumes Alice's qubit, and Eve must prepare a new qubit to send to Bob. The best choice being the

state encoded by the two bit string (*e,z*). Hence Eve forwards Bob a new qubit encoding Eve's measurement result.

Supposing that Eve applies this basic eavesdropping algorithm, there are a few cases that must be analyzed. The first case is when Eve measures in the same basis that Alice encoded in (*i.e. x = z*). It follows that Eve correctly obtains Alice's encoded value, and that Eve prepares the same state that Alice sent. Hence, if Eve measures in the correct basis, and so does Bob, then Eve will obtain a bit of the sifted shared key. The second case is when Eve measures in a different basis than what Alice encoded in (*i.e. x != z*). Eve's measurement result *e* will then be uncorrelated with Alice's encoded value *a*. Furthermore, Eve will then proceed to encode a new qubit in the incorrect basis. If Bob is to measure this manipulated qubit, Bob's result will be uncorrelated with Alice's, even when they measure in the same basis. Hence, if the eavesdropper measures in the incorrect basis a bit error is introduced into the shared key string with 50% chance.

Alice State Encoding	+Z	+Z	-Z	+Z	+X
Eve measurement Result / encoding	+Z	-X	-Z	+X	-Z
Bob measurement result	+Z	-Z	-Z	+Z	+X
Alice Raw Key	00	00	10	00	01
Eve Raw Key	00	11	10	01	10
Bob Raw Key	00	10	10	00	01
Sifted bits	0	error	1	0	0

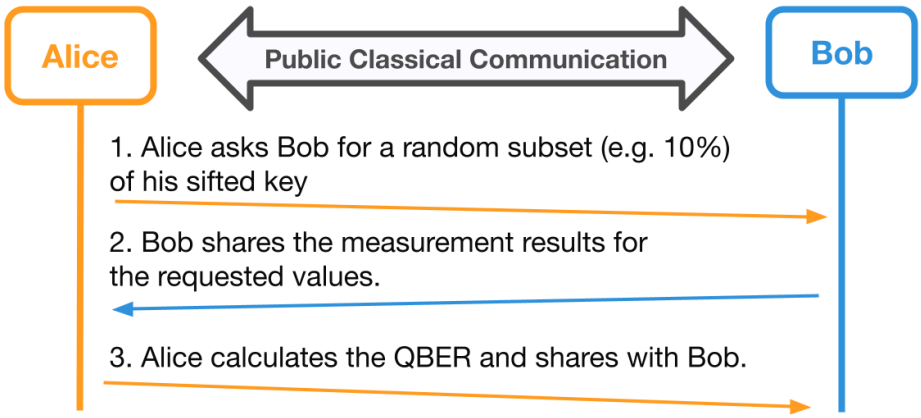
Caption: The table shows example data obtained from the key generation stage where an eavesdropper applies the basic eavesdropping strategy described above. Each column designates an independent round of key generation where the red columns show the cases where Eve learns a bit of sifted key while the green columns show the cases where Eve does not learn the secret key or Alice and Bob obtain a bit error on shared key. The top three rows show the bases used by Alice, Eve, and Bob while the next three rows show each party's raw key prior to sifting. The final row shows the shared key after sifting. Note that Eve is able to learn 3 out of 5 bits of the shared key while only introducing a single error.

In the example table above, we see each of the eavesdropping cases. The first and third columns show the case where Eve measures in the same basis as used by Alice and Bob. As a result, Eve is able to obtain the corresponding bit of sifted key with certainty and without introducing any errors. In the second, fourth, and fifth columns we see the second case occur where Eve's basis does not match Alice and Bob's bases. With 50% chance, Bob's measurement outcome will not align with Alice's resulting in a bit error between their sifted keys (see column two). In the other cases, Alice and Bob happen to get the same

measurement result although their measurements are uncorrelated. In this case, Eve will only have the correct bit with 50% chance. The fourth column is red because Eve obtains the same result as Alice and Bob while the fifth column is green because Eve obtains a different result than Alice and Bob and, therefore, does not share the same bit.

The keen observer will notice that the majority of eavesdropping incidents go undetected. In fact, considering only the cases where Alice and Bob use the same basis, only about one quarter of eavesdropping attempts on these cases will be detectable. Furthermore, this eavesdropping attack would allow Eve to have about 62.5% of the sifted key.

Although it may seem like Eve has a lot of information about the shared key, the introduced errors are detectable. If the quantum bit error rate (QBER), the ratio of error bits to key length, is too large, then there could be an eavesdropper present and the shared key should be discarded. Otherwise, if the QBER is within the tolerable range, Alice and Bob discard the subset of the key that was used to calculate the QBER and continue the protocol. Note that this mechanism for eavesdropper detection results directly from the no-cloning principle of quantum mechanics.



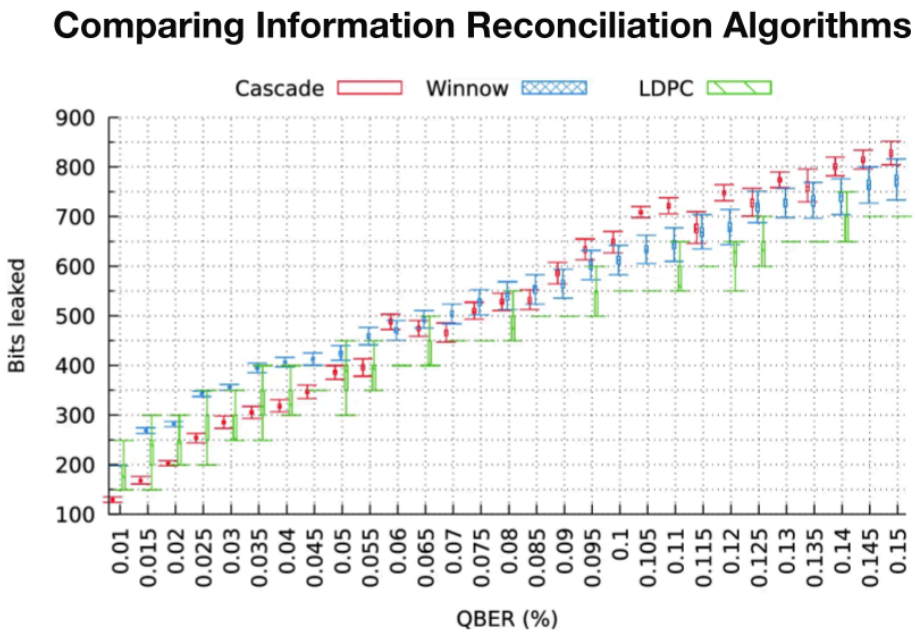
Caption: Alice and Bob calculate their QBER. Alice asks Bob for a random subset of their sifted key to test for errors. Bob responds with the request bits of the sifted key. Using their own key, Alice computes the QBER and shares the value with Bob. If the error is too large, then the key is insecure and should be discarded.

The next question of course is what QBER can be tolerated? Ideally this number should be as small as reasonably possible, however it cannot be too small because the real-world is noisy. Noise also introduces errors in the bit string taking on the appearance of eavesdropping. Unfortunately, it is not possible to distinguish between noise and eavesdropping. This means that a non-negligible amount of error must be tolerated where the threshold depends on the noise characteristics of the hardware.

Establishing Shared Secret Keys

Given that there will be errors between Alice and Bob's sifted keys, even when no eavesdropper is present, there must be a secure mechanism by which to reconcile these errors. In a standard information reconciliation algorithm, Alice will share an error syndrome with Bob, or some information about the key's bits that doesn't reveal the bits themselves. The idea is that Eve will have access to the error syndrome and could use it to try and correct their own key. Using the error syndrome, Bob can decide whether his key string is compatible with Alice's or if an error is present. Through this information reconciliation procedure, Alice and Bob can correct the errors in their bit string while leaking minimal information to an eavesdropper. However, the number of bits leaked to the eavesdropper increases with the quantum bit error rates.

There are three main algorithms that could be used for information reconciliation. There's the cascade algorithm, the WIN algorithm, and the low density parity check algorithm. In the graph below, we can see how many key bits are being leaked, with respect to the quantum bit error rate of the sifted key.



Caption: Three standard information reconciliation algorithms are compared: the Cascade algorithm, the Winnow algorithm, and the Low Density Parity Check (LDPC) algorithm. For a key of fixed length, the average number of bits leaked during the algorithm is plotted with respect to the QBER [Mehic2020].

Privacy amplification

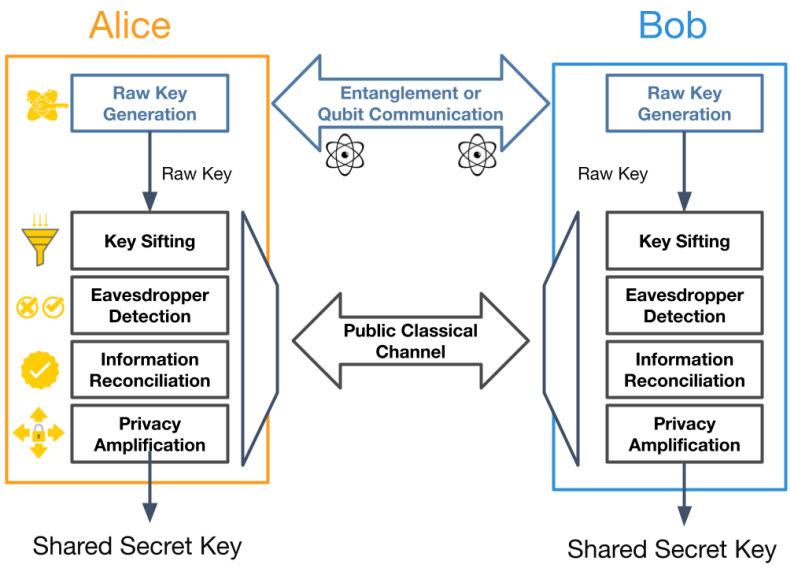
Although information reconciliation allows Alice and Bob to correct their errors, it also leaks additional key information to Eve who might already have intercepted a significant portion of the raw key. For this reason, Alice and Bob's shared key is referred to as a weak secret key. Ideally, the secret should be strong such that Eve has no information about the secret key. To increase the strength of the shared secret key, Alice and Bob can use a privacy amplification algorithm to decouple their key from Eve.

To amplify the privacy of their shared secret key, Alice and Bob apply what is called a randomness extractor. A randomness extractor accepts an input bit string *r* and a random seed bit string *s*. The input bit string is mapped to an uncorrelated bit string using the seed where the new string is totally random to an outside observer. This means that if Eve does not have complete knowledge of the original sequence *r*, applying the randomness extractor with the same seed will result in a bit string that is uncorrelated with Alice and Bob's shared secret. Thus, Alice simply shares the seed with Bob and the two use the randomness extractor to decouple their shared key from Eve.

It is important to note that the output of the randomness extractor will be a shorter bit string than the input *r*. Indeed, the resulting bit string is about half the length of the input. Although the key has become shorter, it has also become more secure because the shared key is decoupled from Eve. Hence, Alice and Bob establish a shared secret key.

Key Rate

The performance of a key distribution protocol is quantified by the key rate, the rate at which bits of the shared secret key are produced. The time it takes to produce a bis of shared secret key is highly dependent on the hardware used for the protocol. For the sake of our discussion, we can equivalently quantify the key rate in terms of rounds of key generation, in which either a single qubit of communication or entangled pair is used. Then for a given hardware platform, time it takes to complete a round of key generation can be calculated and the key rate can be determined.



Caption: Accumulated losses to the shared secret key in each step of the key distribution protocol. For N rounds of key generation, 2N random bits are produced as raw key strings. Key sifting removes about ¾ of these bits resulting in a sifted shared key of length N / 2. To evaluate the QBER, about 10% of the remaining bits should be used, hence the remaining key has about 9N / 20 bits per key generation round. Finally, privacy amplification reduces the shared key's length by about 50% where the established shared secret key has the length of about N / 4 bits, meaning that, on average, only about one quarter of a bit is produced of shared secret key per round of key generation.

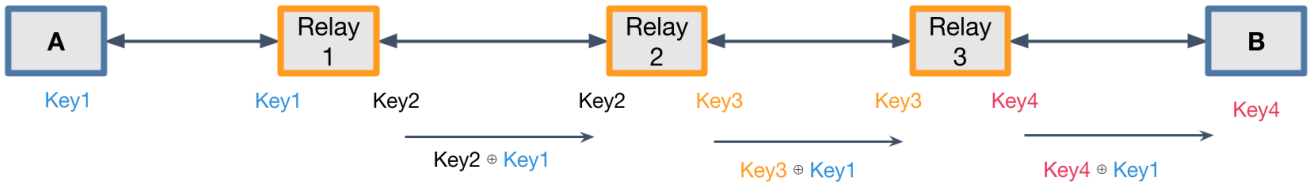
Securely Scaling Key Distribution in Networks

There are two standard solutions for scaling key distribution: trusted relay nodes or quantum repeaters. QKD networks employ trusted relay nodes to scale, while entanglement-based quantum networks use quantum repeaters to scale.

Trusted Relays Nodes

In a network that uses trusted relay nodes, prepare-and-measure communication, like that of QKD, is performed over short distances to mitigate losses. As a result, intermediate relay nodes are needed to transfer the secret key between Alice and Bob. The advantage of these networks is that they can be implemented using commercially available hardware. The drawback, however, is that the relay nodes must be secured because they have access to the secret key as it is relayed between the communicating parties. This means that as the secret key is being shared between Alice and Bob, all relay nodes along the path also possess the key, increasing the security threat that must be managed.

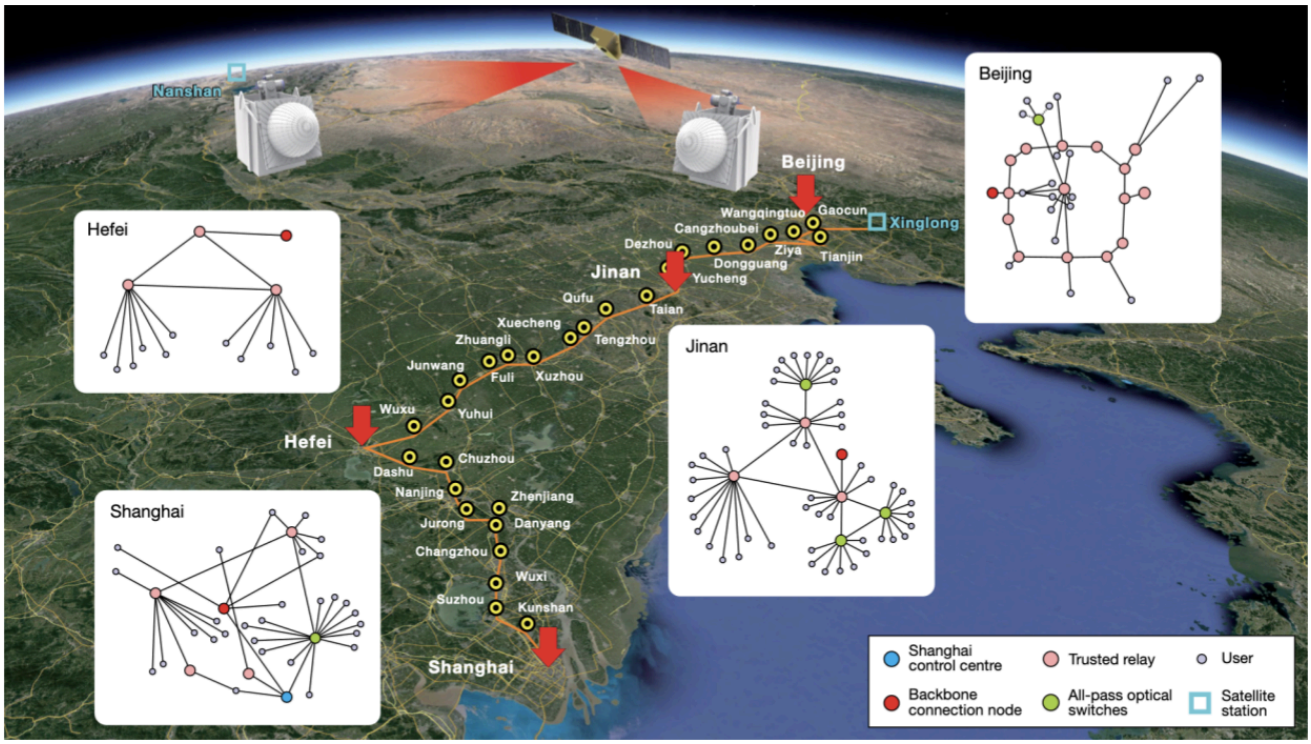
In the first step, each one of these links, as illustrated below, performs BB84 to construct a secret key between each pair of neighboring nodes.



Caption: An example of a QKD network where Alice and Bob are separated by three trusted relay nodes. Alice and Bob want to distribute a shared secret key, but they don't have a quantum channel connecting them. However, there is a quantum communication channel connecting neighboring nodes. In a trusted relay key distribution scheme, a secret key is generated between each pair of neighboring nodes. The secret key generated between Alice and Relay 1 can then be securely communicated to Bob via the relay nodes. Each node relays Alice's key to its neighbor using secret key and symmetric-key encryption. At the end of the relay protocol, Alice, Bob, and all of the relay nodes possess the same secret key.

In the QKD protocol, Alice and Bob must presume trust in the relay nodes. In reality, it may not be possible to verify whether or not a given relay node has been compromised by an adversary. Since all of the relay nodes also possess the secret key material, risk for a security breach increases. As a network scales, the secret key is held by more relay nodes creating more vulnerabilities in the network and increasing the risk of insider threat.

Although the insider threat is a concern, trusted relay nodes can be implemented using existing technologies and these types of networks have been deployed and in use for many years. Since prepare-and-measure communication is used between neighboring nodes in a trusted relay network, there's no need to store the qubit for any length, allowing these networks to be implemented without quantum memories.

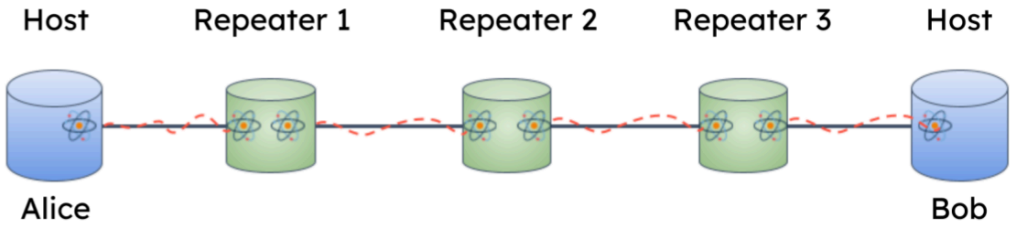


Chen, Yu-Ao, et al. "An integrated space-to-ground quantum communication network over 4,600 kilometres." *Nature* 589.7841 (2021): 214-219.

Caption: An example of a QKD network with trusted relay nodes deployed in China. The network uses both satellite and fiber optic cable interconnects between neighboring nodes. Using trusted relay nodes, many points throughout China can securely communicate for practical purposes [Chen2021].

Quantum Repeaters

Entanglement-based quantum networks use quantum repeaters to scale the network's reach. As shown in the image below, quantum repeaters perform entanglement swapping to distribute entanglement between Alice and Bob over great distances.



Caption: Alice and Bob are linked through a chain of intermediate quantum repeater nodes. Entanglement is shared between neighboring nodes. By performing entanglement swapping and entanglement distillation protocols, entangled qubits can be shared between Alice and Bob. Note that many entangled pairs between intermediate nodes are consumed to generate one entangled pair between Alice and Bob. Therefore, rates of entanglement distribution can be slow in entanglement-based networks.

Once Alice and Bob have established entanglement from end-to-end, they can either use the BBM92 protocol to produce a secret key, or they could use quantum teleportation to send a qubit from Alice to Bob and apply the BB84 protocol. Since teleportation adds an extra step, the entanglement-based BBM92 protocol is more efficient. Although key distribution protocols in entanglement-based quantum networks are not functionally different from QKD networks, entanglement-based quantum networks offer greater key distribution security because no intermediate relay nodes have access to the key material.

The improved security of entanglement-based quantum networksover QKD networks that use trusted relay nodes is due to the intermediate quantum repeater nodes only having access to the entangled qubits, not the secret key. In the worst case, one of these repeaters could be an eavesdropper trying to correlate their local data with Alice and Bob's secret key. However, the whole point of key distribution protocols is that they are able to detect the eavesdropper. Even if one of these repeaters were to be malicious and eavesdrop on Alice and Bob's secret key, they would be detected and the key would be discarded. Since only Alice and Bob know the secret key, entanglement-based quantum networks manage the insider threat posed by QKD networks using trusted relay nodes.

The main challenge preventing entanglement-based quantum networks from being scaled is that entanglement-based networking requires high fidelity quantum repeaters to efficiently distribute entanglement. Developing high-fidelity quantum repeaters is an active area of research and development, and we can hope to see major breakthroughs in coming years.

Comparing Key Distribution Networks

Each method of scaling key distribution networks has its own strengths and weaknesses. It is worth noting that comparing QKD with QSC is not about comparing BBM92 and BB84, because either protocol could be implemented in either networking scheme. The key differences between the two key distribution networks are their availability and their relative security risk at scale.

	Networks that use Trusted Relay Nodes	Networks that use Quantum Repeaters
Availability	Has been demonstrated and deployed at scale.	Improvements to quantum repeater technology required.
Security Risk at Scale	Increased insider threat risk due to the relay nodes having the secret key.	Managed insider threat risk because no relay nodes are given the secret key.
Key Rate Limitations	Limited by rate of qubit preparation, transmission, and measurement.	Limited by end-to-end entanglement generation rates.
Network Flexibility	Quantum protocols between neighboring nodes only.	QSC and QKD protocols between *any* network nodes. Enables other types of protocols for a variety of use cases.

Comparing QKD networks that use trusted relay nodes to scale with QSC on entanglement-based quantum networks that use quantum repeaters to scale. This table is a comparison on the qualities of availability, security risk at scale, key rate limitations, and network flexibility based on the technology used to scale. Cells are shaded green for positive, yellow for neutral, and red for negative where justification is given in each cell.

Availability is the main selling point for QKD networks because they have been demonstrated and standardized using trusted relay nodes. Networks that use trusted relay nodes can be deployed and scaled with off-the-shelf components. Entanglement-based quantum networks are being actively built, but are not built with off-the-shelf components. Scaling entanglement-based quantum networks requires high-fidelity quantum repeaters, which are still being researched and developed. Quantum repeater technologies will improve in the coming years, enabling entanglement-based quantum networks to be demonstrated and deployed at scale.

The security risk as the network scales presents a critical difference between the two key distribution network schemes. Namely, as entanglement-based quantum networks scale, their security threat profile remains constant because Alice and Bob perform key distribution with each other using their shared entanglement. On the other hand, the risk of security breach escalates as QKD networks scale. The reason is that each of the intermediate relay nodes is given access to the secret key. This means that each of the relay nodes can decrypt Alice and Bob's message. Although the relay nodes are presumed to be trusted and secure, there is always risk of a security breach due to an insider leaking information or other vulnerabilities. As a result, the security risk increases when the network scales and more trusted relay nodes are required.

The third point of comparison is key rate. Both types of networks face restrictions in this area largely due to loss and photon production rates. For QKD Networks and entanglement-based

quantum networks, the key rate is limited by the rate at which qubits can be prepared, transmitted, and measured. Since errors accumulate during entanglement distribution scaling these key distribution networks require high-fidelity repeaters and high-rate entanglement sources such that impact error correction and entanglement distillation are offset.

Flexibility is another area where these two networks differ significantly. QKD networks that use trusted relay nodes are limited in their applications, supporting only QKD protocols between neighboring nodes. Entanglement-based quantum networks offer remarkable flexibility because they enable entanglement between any two nodes in the network, regardless of proximity. As a result, these networks can do much more than key distribution including a wide range of applications such as distributed quantum computing, distributed quantum sensing, as well as a variety of security protocols.

While entanglement-based quantum networks exhibit superior security and flexibility, they do face some practical implementation challenges. QKD networks, on the other hand, benefit from immediate availability and ease of construction using existing components, but are limited in capability and vulnerable to insider threats. As technology advances, the advantages of entanglement-based quantum networks will become more accessible, enabling secure and versatile communication in the future as well as other applications beyond secure communication.

Designing a key distribution network

Designing a key distribution system involves multiple factors, including the choice of protocol, physical hardware, scalability, post-processing algorithms, and security measures. Each decision impacts the overall performance and security of the network, and thorough planning and evaluation are necessary to build a robust and reliable key distribution system.

The first consideration when designing a key distribution system is selecting the appropriate key distribution protocol. In this white paper, we used BBM92 and BB84 to respectively represent Quantum Secure Communication (QSC) and QKD protocols, however, there are other variations such as continuous variable QKD, device-independent QKD, and measurement device-independent QKD. Since each protocol has its own merits and drawbacks, the requirements and constraints of your application can help identify the most suitable protocol.

The second consideration involves selecting the hardware for the key distribution system. The choice of protocol can help dictate the hardware needed for the protocol. This may require selecting appropriate entanglement sources, identifying the appropriate frequencies for

emission, and deciding on the physical qubit encoding. Hardware decisions impact not only the performance of the network, but also the compatibility with existing infrastructure and the complexity of the implementation. Thoroughly vetting and evaluating available technologies is essential to the hardware selection process. A quantum network simulation can be helpful in working through these complex choices and how they will affect a network environment.

Scalability is another important aspect to consider in your key distribution network design. This requires identifying the number of users the network must support, the distances between nodes, and the losses in the network. Scaling a key distribution network involves many logistical and technical challenges, managing the loss of quantum states, synchronizing devices, and routing entanglement or trusted relays between nodes.

Another design question that needs to be addressed is the selection of algorithms used for information reconciliation and privacy amplification, which are essential to ensuring the security and reliability of the key distribution process. Different algorithms have various strengths and are suited to different levels of noise and error rates. Understanding the expected quantum bit error rate (QBER) and the specific requirements of the application can guide the selection of the most effective algorithms.

Finally, the security of the network implementation must be evaluated against potential attacks. Every network has a unique threat profile based on its hardware, connectivity, and algorithms. It is essential to identify these vulnerabilities and develop strategies for mitigating these risks.

Conclusion

Advancements in quantum computing threaten the security of our information technology systems and networks. To counter this threat, post-quantum cryptography (PQC) and Quantum Secure Communication (QSC) protocols can be applied. Both QSC and PQC offer solutions to these emerging threats, and they can be used together to enhance network security depending on specific use cases. The combined use of QSC and PQC provides a comprehensive security solution, called Advanced Secure Networking, which balances immediate feasibility with long-term resilience against quantum computing advancements. As research and technology progress, these methods will play a critical role in safeguarding future communication networks. Advanced Secure Networks uniquely enable the highest level of security with end-to-end entanglement provided by quantum networks. These networks are multipurpose, hardware agnostic, are used to implement QSC, and are compatible with PQC.

Entanglement-based quantum networks are being developed by a variety of organizations for a variety of use cases – benefiting organizations internally, as well as providing great value to an organization's customers. Aliro is helping to leverage the capabilities of entanglement-based quantum networks working with telecommunications companies, national laboratories, intelligence organizations, and systems integrators.

Building entanglement-based quantum networks that use entanglement is no easy task. It requires:

- Emerging hardware components necessary to build the network.
- Software for network design, simulation, and management.
- Expertise in both classical networks and quantum information science and technology.

Aliro is uniquely positioned to help clients build entanglement-based quantum networks. Aliro will ensure that your organization is ready to meet the challenges and leverage the benefits of the quantum revolution. Our unified solution is already at work in the EPB Quantum Network℠ powered by Qubitekk in Chattanooga, Tennessee.

AliroNet™, the world's first full-stack entanglement-based quantum network solution, consists of the software and services necessary to ensure that customers fully meet their secure networking goals, including developing Advanced Secure Networks that leverage PQC as well as entanglement-based quantum networks. Each component of AliroNet™ is built to be compatible with entanglement-based quantum networks of any scale and architecture. AliroNet™ is used to simulate, design, and manage entanglement-based quantum networks as well as test, verify, and optimize quantum hardware for network performance. AliroNet™

leverages the expertise of Aliro personnel in order to ensure that customers get the most value out of the software and their investment in entanglement-based quantum networks.

Depending on where customers are in their quantum networking journeys, AliroNet™ is available in three modes that create a clear path toward building full-scale entanglement-based quantum networks: (1) Emulation Mode, for emulating, designing, and validating entanglement-based quantum networks, (2) Pilot Mode for implementing a small-scale entanglement-based quantum network testbed, and (3) Deployment Mode for scaling entanglement-based quantum networks and integrating end-to-end applications. AliroNet™ has been developed by a team of world-class experts.

To get started on your quantum networkingg journey, reach out to the Aliro team for additional information on how AliroNet™ can enable secure communications, networking quantum computing, and networked quantum sensing.

info@alirotech.com www.alirotech.com

References

[Bennett1984] C. H. Bennett and G. Brassard. "Quantum cryptography: Public key distribution and coin tossing". In *Proceedings of IEEE International Conference on Computers, Systems and Signal Processing*, volume 175, page 8. New York, 1984.

[Bennett1992] Bennett, Charles H., Gilles Brassard, and N. David Mermin. "Quantum cryptography without Bell's theorem." *Physical review letters* 68.5 (1992): 557.

[Chen2021] Chen, Yu-Ao, et al. "An integrated space-to-ground quantum communication network over 4,600 kilometres." *Nature* 589.7841 (2021): 214-219.

[Leifer2006] Leifer, Matthew S. "Quantum dynamics as an analog of conditional probability." *Physical Review A—Atomic, Molecular, and Optical Physics* 74.4 (2006): 042310.

[Mehic2020] Mehic, Miralem, et al. "Error Reconciliation in Quantum Key Distribution Protocols." (2020): 222-236.

[Wilde2013] Wilde, Mark. *Quantum information theory*. Cambridge university press, 2013.