Protocols for Key Distribution

Aliro

www.alirotech.com

Protocols for Key Distribution

Summary

This white paper focuses on the critical role of key distribution in secure communications and explores the transition from classical to quantum-based encryption methods. This is a necessary transition, as current classical encryption protocols, such as RSA and Diffie-Hellman, are vulnerable to quantum computing technologies.

To counteract this threat, we introduce quantum key distribution (QKD), such as BB84, Quantum Secure Communication (QSC), and interference-based QKD (IQKD). These protocols leverage the unique properties of quantum mechanics to ensure secure key distribution, providing potential solutions to the threats posed by quantum computers. Hardware requirements for such protocols, as well as classical post-processing steps like error correction and privacy amplification, are discussed as part of using these protocols.

Introduction

The rapid advancement of quantum computing represents both exciting opportunities and significant challenges for the future of secure communications. Traditional encryption methods, which have long safeguarded sensitive information, are increasingly vulnerable to quantum-based attacks. Protocols like RSA and Diffie-Hellman, which rely on the computational difficulty of factoring large numbers, are at risk of being compromised by quantum computers running algorithms such as Shor's algorithm. This looming threat has sparked a critical need for new approaches to secure key distribution.

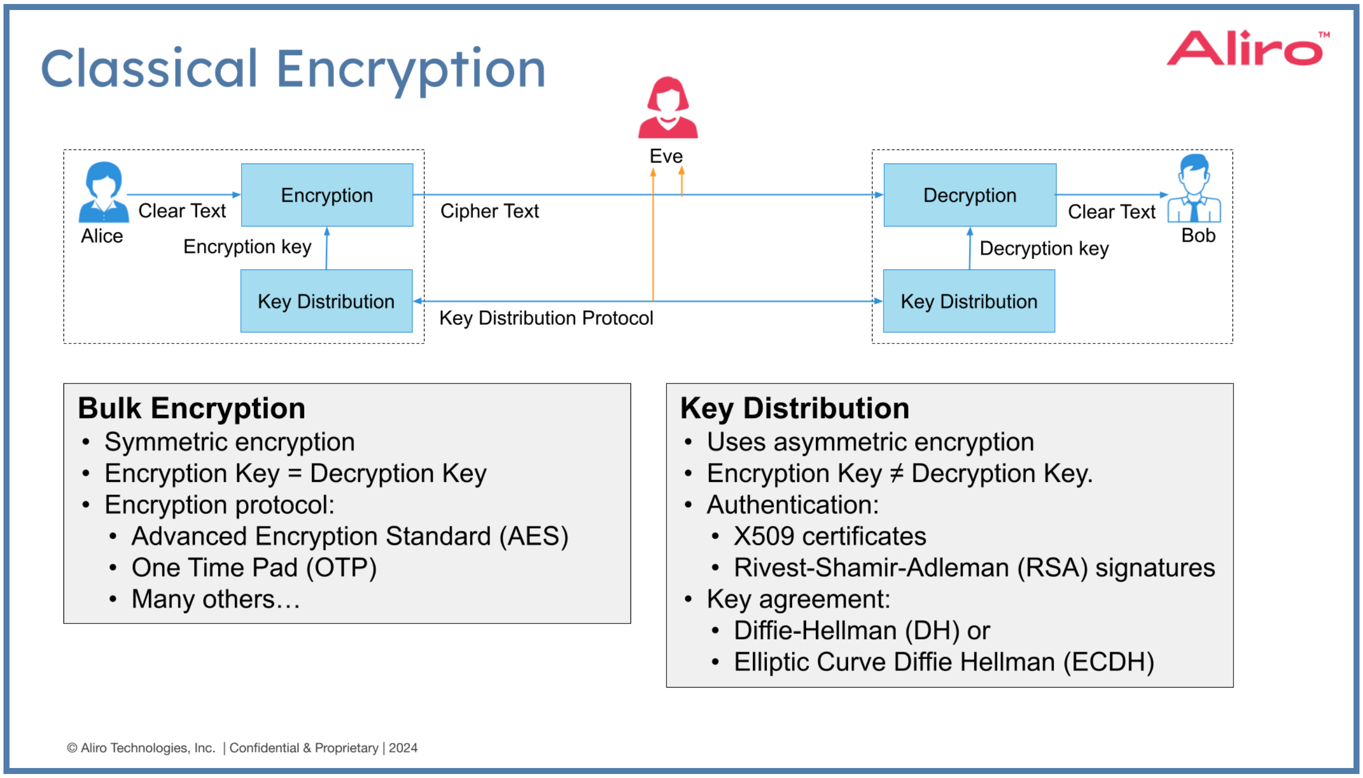
Quantum Key Distribution (QKD), Quantum Secure Communication (QSC), and Interference-based Quantum Key Distribution (IQKD) are emerging as potential solutions to this challenge, leveraging the principles of quantum mechanics to enable secure encryption key exchanges and transfer of data. Unlike classical methods, these methods ensure that any attempt to eavesdrop on the key exchange process is immediately detectable, providing a level of security that classical encryption cannot match. This paper explores these protocols and examines the practical requirements for implementing these advanced systems.

As organizations face the reality of a quantum-enabled future, the transition to quantum-secure networks is becoming imperative. By exploring the latest advancements in secure key distribution and providing insights into emerging technologies, this white paper aims to equip businesses and institutions with the knowledge needed to stay ahead of evolving cyber threats – today and into the future.

Classical Encryption

Two parties, traditionally called Alice and Bob, need to communicate with each other in a secure manner. However, there is an eavesdropper, Eve, who wants to discover or perhaps even change the messages between Alice and Bob.

Alice and Bob use symmetric encryption for encrypting the bulk transfer of large amounts of data.



Classical Encryption

Bulk Encryption

- Symmetric encryption
- Encryption Key = Decryption Key
- Encryption protocol:
  - Advanced Encryption Standard (AES)
  - One Time Pad (OTP)
  - Many others…

Key Distribution

- Uses asymmetric encryption
- Encryption Key ≠ Decryption Key.
- Authentication:
  - X509 certificates
  - Rivest-Shamir-Adleman (RSA) signatures
- Key agreement:
  - Diffie-Hellman (DH) or
  - Elliptic Curve Diffie Hellman (ECDH)

Symmetric encryption means that the encryption key is the same as the decryption key.

The main advantage of symmetric encryption is that it can be implemented in hardware relatively easily as it has a very high performance of up to terabits per second. The most widely used symmetric encryption protocol at this point in time is called the Advanced Encryption Standard or AES, but there are many other protocols as well.

Using a symmetric encryption protocol requires Alice and Bob to agree on an encryption key. This key is called a session key, and it is typically changed or rolled over periodically for increased security. Alice and Bob could agree on a pre-shared key a-priori, but this is very inconvenient in practice, so dynamic protocol for authenticating each other and for agreeing on a session key is used instead.

Currently, protocols based on X.509, RSA, Diffie-Hellman or Elliptic-Curve Diffie-Hellman are used for authentication and key agreement. For a deeper look at these protocols, watch the on-demand webinar The Evolution of Network Security: Bridging Classical and Quantum Systems.

The Quantum Threat to Classical Encryption

The challenge we face today is that classical key distribution will be vulnerable to attack by quantum computers. RSA, Diffie-Hellman and Elliptic Curve Diffie-Hellman all rely on the computational complexity of certain mathematical problems, such as factoring large numbers into prime factors.

In 1994, mathematician Peter Shor discovered what is now known as Shor's algorithm. Shor's algorithm enables a sufficiently powerful quantum computer to factor large numbers in *seconds* instead of the millions of years it would take today’s most powerful computers.

Fortunately, the quantum computers that exist today are still too small for running Shor's algorithm. However, the expectation is that in the not-too-distant future we *will* have a quantum computer that is powerful enough to run Shor's algorithm. That day is referred to as Q-day.

From Q-Day on, existing protocols such as RSA, Diffie-Hellman, and Elliptic Curve Diffie-Hellman will no longer be secure. This renders much of our secure communications vulnerable to attacks on Q-Day, but also to Harvest Now Decrypt Later (HNDL) attacks. In HNDL attacks, an adversary harvests data and saves it until sufficient resources exist to crack the encryption. This means data could be collected today, and decrypted by quantum computers at a future date - placing today’s secure communications at risk.

What can be done today to prepare for Q-Day?

There are several possible approaches:

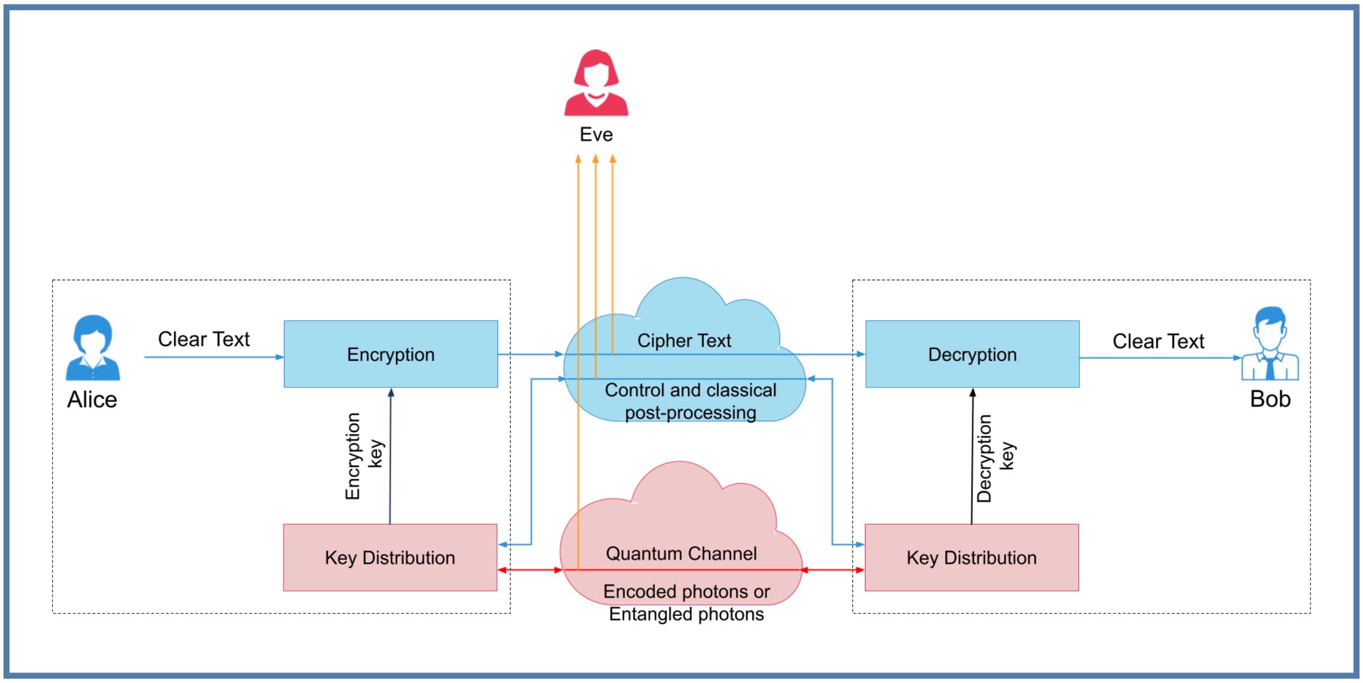
- Post Quantum Cryptography (PQC). PQC uses different mathematical problems, such as lattice-based cryptography, to protect data. PQC mathematical encryption algorithms are believed to be safe against attack even by quantum computers.
- Quantum Key Distribution (QKD). QKD uses certain quantum mechanical properties of photons to protect the key exchange from attack by an eavesdropper. Alice encodes the key bits into individual photons, sends them to Bob and Bob decodes the photons back into key bits. 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). QSCleverages principles of quantum entanglement to establish secret keys between two parties. In QSC, a mid-point station sends pairs of entangled photons to Alice and Bob. Alice and Bob then each decode their received photons into key bits. An example of this type of protocol for key distribution would be BBM92.

- Interference-based Quantum Key Distribution (IQKD). IQKD relies on the principles of quantum interference to establish a secure key between two parties. Thistypically involves a central measurement device that detects interference patterns to extract key information. An example of this type of protocol for key distribution is MDI-QKD.
- Quantum Secure Direct Communications (QSDC). QSDC also uses certain quantum mechanical properties of photons, but it is not used to protect the key exchange. Instead, QSDC protects the data itself that is being transferred.

This white paper explores QKD, QSC, and IQKD.

Network Channels for Key Distribution

QKD, QSC, and IQKD all utilize two channels to dynamically distribute session keys to Alice and Bob.



The first channel is the quantum channel shown in red at the bottom of this slide. Here we encode the keys into the quantum mechanical properties of photons. This is done in such a manner that Eve cannot steal the key without being detected. Later in this seminar we will describe how the encoding of key bits into the quantum properties of photons works. And we will explain why this protects against Eve stealing the keys.

The second channel is a normal classical channel shown in blue at the top of this slide. When we say classical channel, we mean just a normal network as it exists today, such as an Ethernet network. The classical channel is used for some control and post-processing steps such as parameter estimation, error correction, and privacy amplification. Later in this white paper we will describe these classical post-processing steps in detail.

Both the quantum channel and the classical channel can use normal telecom fibers, or they can use free-space optical connections, such as satellite connections.

Key Distribution Hardware

Single Photon Sources (SPS)

The security of key distribution protocols that leverage quantum mechanics relies on the fact that trying to decode a single photon in the wrong basis leads to random results. This only works for single photons, not for strong classical light pulses. Thus, for QKD to work properly, it is necessary to work with single photons.

QKD requires light sources that can emit very weak pulses of light containing only one individual photon. There exist many different single photon source technologies, including attenuated photon diode lasers, quantum dots, and nitrogen vacancy centers.

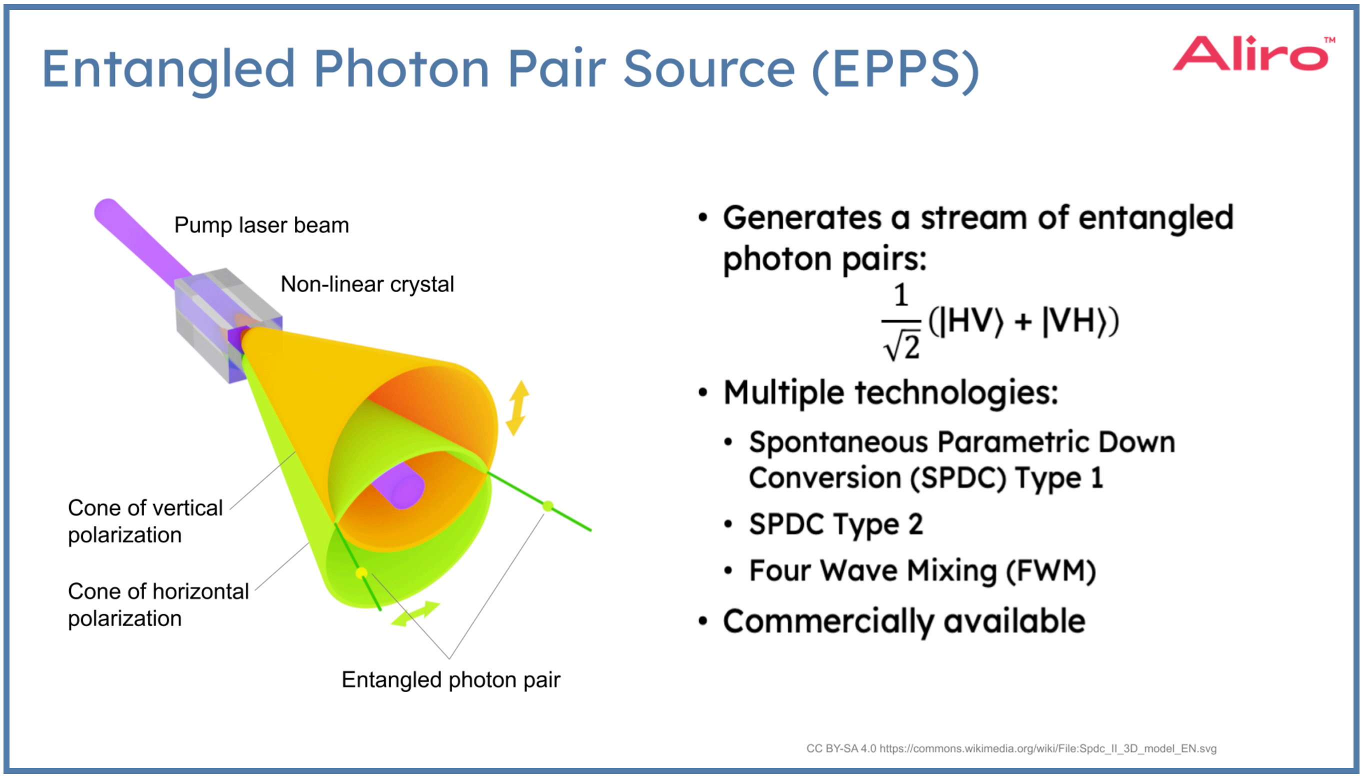
Different technologies and vendors vary in terms of performance characteristics. Some example characteristics include the light pulse wavelength and bandwidth, the light pulse rate, and whether the light pulses are produced on-demand or randomly. There are also practical considerations such as the cost of the device and whether or not it needs cryogenic cooling.

In practice, most QKD devices use attenuated photo diode lasers. They are also known as weak coherent sources. These are relatively inexpensive, small, robust, and don't require complex cryogenic cooling. The main disadvantage is that the light pulses that they produce don't always contain a single photon; sometimes they contain zero or multiple photons. This has important implications for the security of networks running QKD protocols, which we will discuss in more detail later.

Entangled Photon Pair Source (EPPS)

QSC requires a special type of photon source that produces a stream of entangled photon pairs instead of a stream of individual photons.

There are multiple technologies that can be used to implement Entangled Photon Pair Sources. The illustration on the next page shows one particular implementation, called Type 2 Spontaneous Parametric Down Conversion or SPDC.[EPPS]



Entangled Photon Pair Source (EPPS)

Pump laser beam
Non-linear crystal
Cone of vertical polarization
Cone of horizontal polarization
Entangled photon pair

- Generates a stream of entangled photon pairs:

$$\frac{1}{\sqrt{2}}(|HV\rangle + |VH\rangle)$$

- Multiple technologies:
  - Spontaneous Parametric Down Conversion (SPDC) Type 1
  - SPDC Type 2
  - Four Wave Mixing (FWM)
- Commercially available

CC BY-SA 4.0 https://commons.wikimedia.org/wiki/File:Spdc_II_3D_model_EN.svg

The purple line is a pump laser that shines a stream of pump photons into a particular type of crystal. The crystal causes some small fraction of the pump photons to split up into two separate photons, called the idler photon and the signal photon. One of these photons is always horizontally polarized and the other is always vertically polarized. All the horizontally polarized photons always exit somewhere on the orange cone. All the vertically polarized photons always exit somewhere on the green cone.

Where the two cones overlap, if a pair of photons exits on these two lines, they will be entangled with each other. It is not possible to predict which photon has which polarization, but when measured they will always have opposite (horizontal or vertical) polarizations.

Single Photon Detectors (SPD)

Single photon detectors are also required for QKD, QSC, and IQKD. The two most commonly used technologies are SPADs and SNSPDs.

Most QKD devices use Single Photon Avalanche Diodes or SPADs. SPADs are relatively inexpensive, small, robust, and don't require complex cooling. However, they have inefficiencies: the probability that they actually detect an arriving photon is relatively low. This limits the maximum distance of the quantum links.

For longer links, Superconducting Nanowire Single Photon Detectors or SNSPDs are better. This is mainly because they have a better detection efficiency. However, they are also more expensive and cumbersome because they require liquid helium cooling and vacuums.

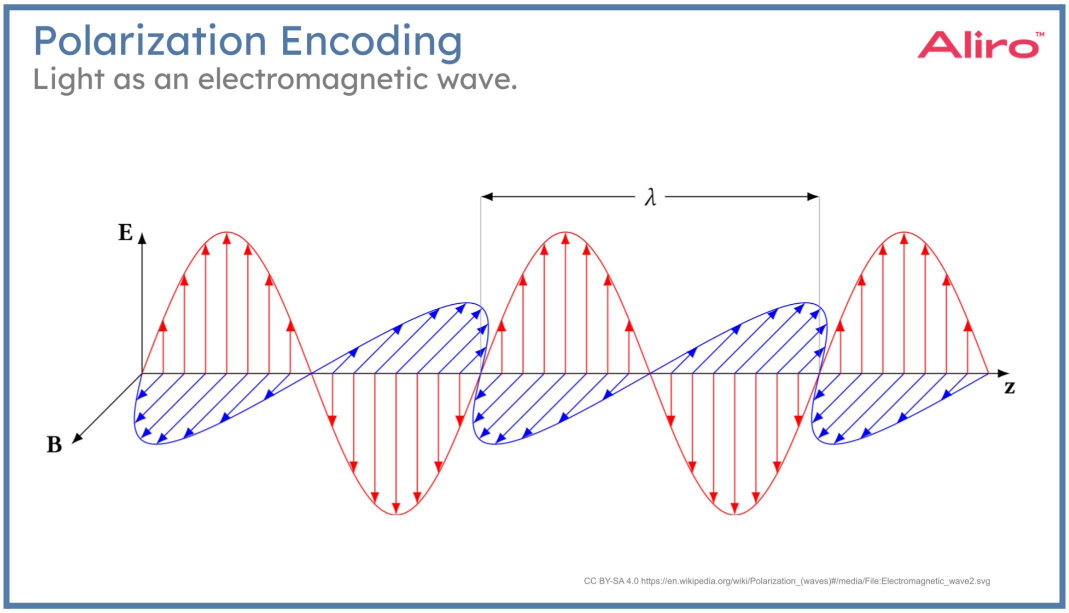
SPADs and SNSPDs also differ in various other performance characteristics which affect the design, performance, and security analysis of the quantum system for key distribution. Two of the most important examples of such performance characteristics are the dark count rate and the dead interval. The dark count rate is the rate at which the detector clicks even though no photon actually arrived. And the dead interval is the amount of time after a click during which the detector is blind in the sense that it cannot detect another photon.

QKD Protocols

QKD protocols, such as BB84, encode information into single photons (also known as qubits), and then send these photons over the quantum channel in a network. The most common and easiest to understand method for encoding qubits is polarization encoding.

Polarization Encoding

Light can be envisioned as an electro-magnetic wave.



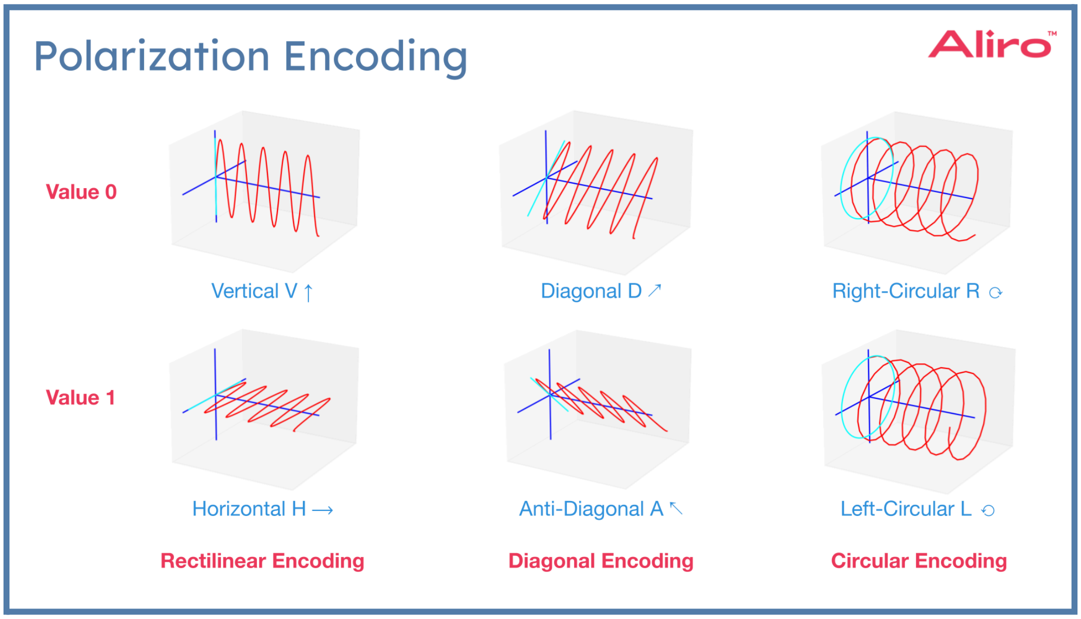
Polarization Encoding

Light as an electromagnetic wave.

CC BY-SA 4.0 https://en.wikipedia.org/wiki/Polarization_(waves)#/media/File:Electromagnetic_wave2.svg

In this diagram the red wave is the oscillating electric field. The blue wave is the oscillating magnetic field.[POLARIZATION]

The oscillating electric field can point in different directions. This is called the polarization of light. Key bits can be encoded into this polarization, and this is called polarization encoding. For example, an up-down vertical oscillation can be used to represent a zero bit and a left-right horizontal oscillation can be used to represent a one bit.



Polarization Encoding

Aliro™

Value 0

Vertical V ↑

Diagonal D ↗

Right-Circular R ⟳

Value 1

Horizontal H ⟶

Anti-Diagonal A ↖

Left-Circular L ⟲

Rectilinear Encoding

Diagonal Encoding

Circular Encoding

This is called rectilinear encoding as shown in the first column in the diagram above.

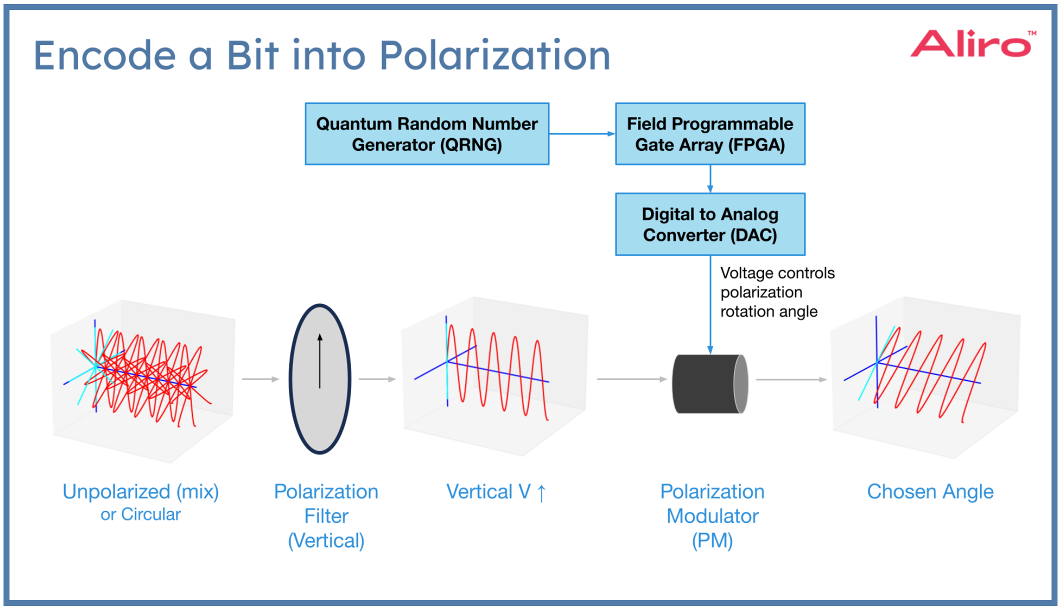
Alternatively, the diagonal polarization slanted to the right can be used to represent a zero bit and an anti-diagonal polarization slanted to the left can be used to represent a one bit. This is called diagonal encoding, as shown in the second column.

There is also the possibility that the polarization twists around the direction of propagation like a cork-screw. In this clockwise twist to represent a zero bit and we can use an anti-clockwise twist to represent a one bit. This is called circular polarization as shown in the third column.

It is relatively easy to implement the encoding of a bit into the linear polarization of light.

Start with a laser that emits light pulses. If the light coming out of the laser is not already polarized, pass it through a polarization filter to produce vertically polarized light. Finally, rotate the polarization by a given amount to produce light of the desired polarization. This is accomplished with a device called a polarization modulator.

The amount of rotation is determined by a voltage which is typically controlled using an FPGA and a Digital to Analog Converter. Using an FPGA allows the polarization to be set for each individual light pulse, even if the laser pulses at a very high rate, up to around a billion pulses per second.



Encode a Bit into Polarization

Quantum Random Number Generator (QRNG)

Field Programmable Gate Array (FPGA)

Digital to Analog Converter (DAC)

Voltage controls polarization rotation angle

Unpolarized (mix)
or Circular

Polarization Filter (Vertical)

Vertical V ↑

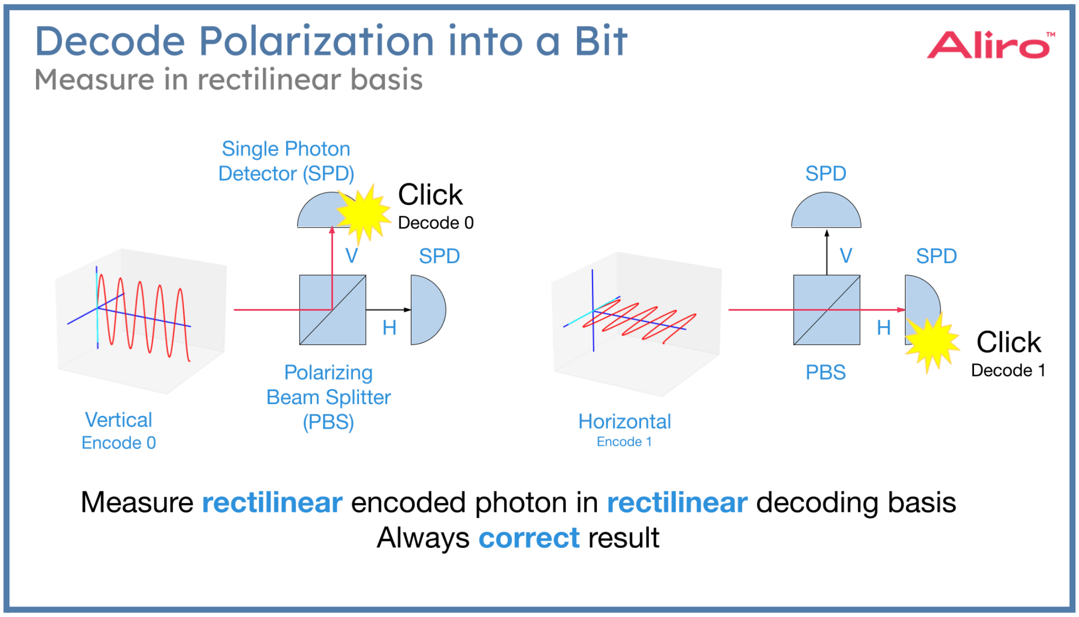
Polarization Modulator (PM)

Chosen Angle

It is also easy to decode a polarized light pulse back into a classical bit.

For now, assume that the incoming light pulse uses rectilinear encoding: vertical for zero and horizontal for one.

The arriving light pulse is passed into a device which is called a polarizing beam splitter. This device has the special property that it sends vertically polarized light pulses in one direction (up in this diagram) and horizontally polarized light pulses in another direction (right in this diagram).



Decode Polarization into a Bit

Measure in rectilinear basis

Single Photon Detector (SPD)

Click
Decode 0

V

SPD

H

Polarizing Beam Splitter (PBS)

Vertical
Encode 0

SPD

V

SPD

H

PBS

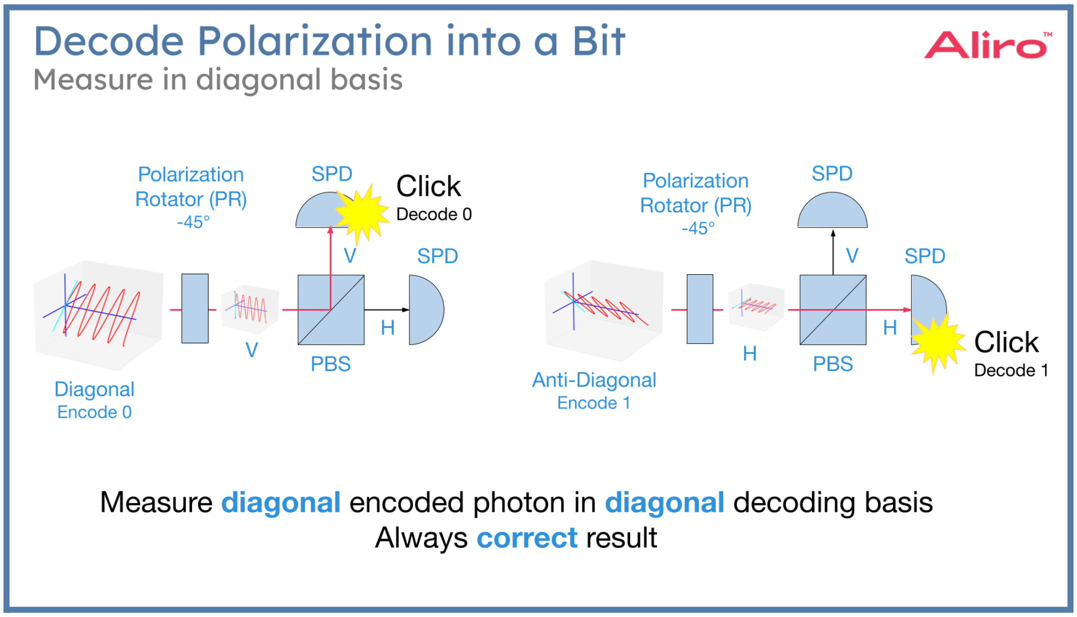
Click
Decode 1

Horizontal
Encode 1

Measure **rectilinear** encoded photon in **rectilinear** decoding basis
Always **correct** result

Single photon detectors are located on the two sides of the polarizing beam splitter. The polarization of the arriving light pulse is known based on which detector clicks, meaning which detector detects a pulse of light. This way of decoding the pulse is called rectilinear decoding or decoding in the rectilinear basis.

What if the arriving light pulse is diagonally encoded: diagonal for zero and anti-diagonal for one? How is it decoded into a classical bit?



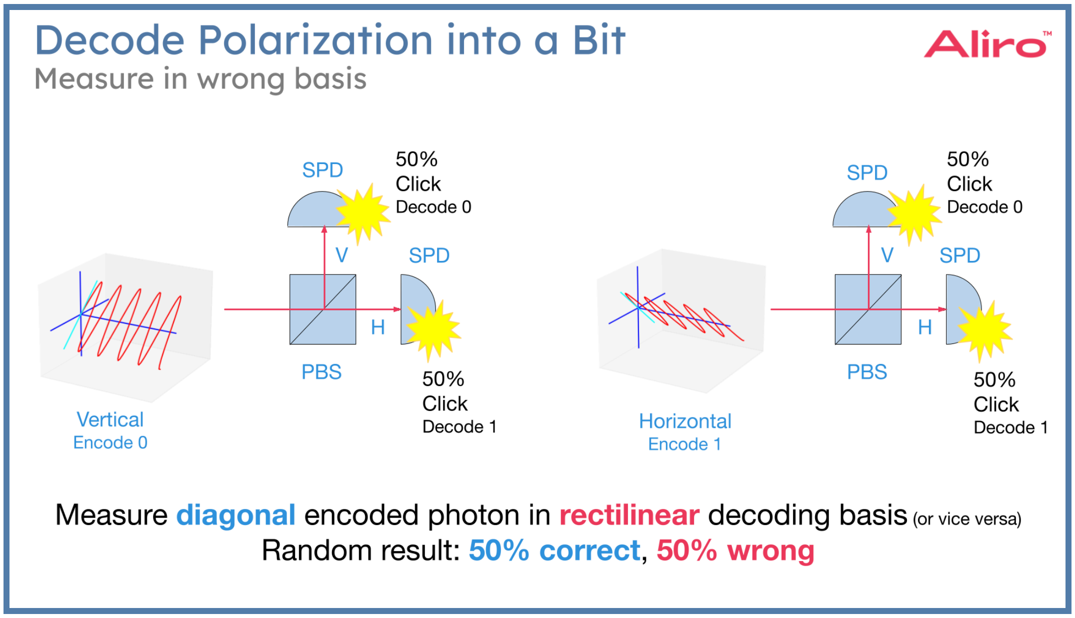
Decode Polarization into a Bit

Measure in diagonal basis

Measure **diagonal** encoded photon in **diagonal** decoding basis
Always **correct** result

In this case, the arriving light pulse first passes through a device called a polarization rotator to rotate the polarization by 45 degrees before it goes into the polarizing beam splitter. This turns diagonally encoded polarization into rectilinearly encoded polarization. The most common way to implement a fixed rotation is to use a half-way plate. This decoding scheme is referred to as diagonal decoding or decoding in the diagonal basis.

What happens if a diagonally encoded photon is received, but decoded in the wrong basis - for example, the rectilinear basis?



Decode Polarization into a Bit

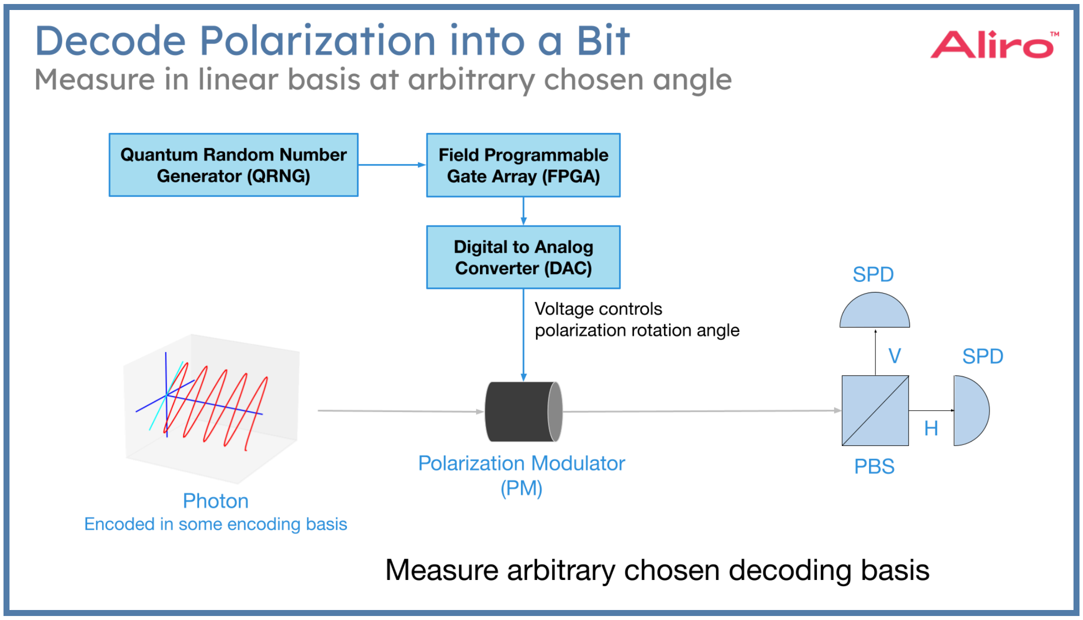
Measure in wrong basis

Measure **diagonal** encoded photon in **rectilinear** decoding basis (or vice versa)
Random result: **50% correct**, **50% wrong**

The decoder in the above diagram is missing the polarization rotator from the previous diagram. First consider the classical case, where the light is a strong laser beam. In the classical case, half of the optical power goes into one direction and the other half of optical power goes into the other direction but what happens in the quantum case, with a single photon instead of a strong beam of laser light?

Since a single photon is an individual particle of light, it cannot be split into two parts, one going in each direction. Instead, the photon randomly takes one direction or the other. There is a 50 percent probability that the photon goes right and a 50 percent probability that the photon goes up. The important thing to remember is that if this photon is decoded in the wrong basis, purely random bit values are the result. This randomness is a fundamental outcome of the laws of quantum physics and not some limitation of how good or bad the hardware is.

Many key distribution protocols that leverage quantum physics involve randomly choosing the decoding basis, rectilinear or diagonal, for each arriving photon. This can be achieved by using a polarization modulator controlled by an FPGA, as shown in the diagram below.



Decode Polarization into a Bit

Measure in linear basis at arbitrary chosen angle

Quantum Random Number Generator (QRNG)

Field Programmable Gate Array (FPGA)

Digital to Analog Converter (DAC)

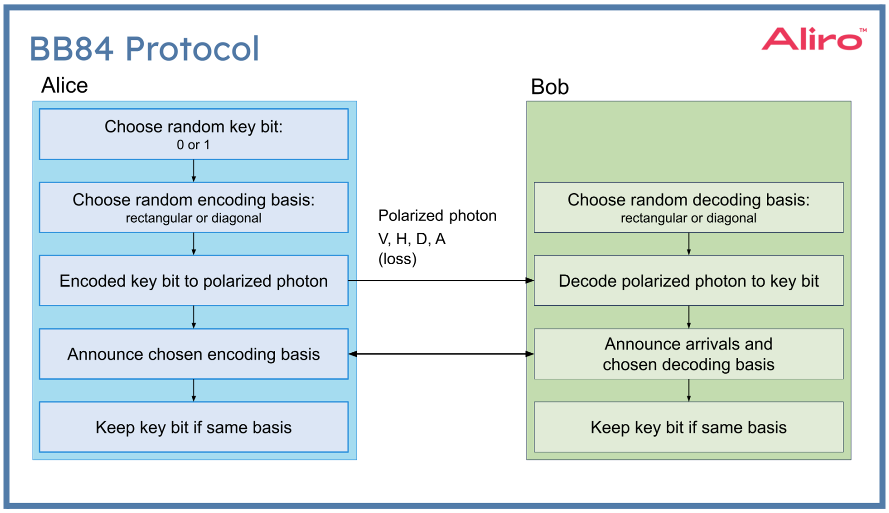
Voltage controls polarization rotation angle

Measure arbitrary chosen decoding basis

For each arriving light pulse, the polarization modulator rotates the polarization by either 0 degrees or by 45 degrees. A quantum random number generator is used to make the decoding basis choice unpredictable.

BB84 Protocol

There are other QKD protocols, but here the focus will be on BB84. Like many QKD protocols, BB84 is named after its inventors, Bennet and Brassard, and the year of invention, 1984. The protocol is fairly straightforward.



BB84 Protocol

Alice

Choose random key bit:
0 or 1

Choose random encoding basis:
rectangular or diagonal

Encoded key bit to polarized photon

Announce chosen encoding basis

Keep key bit if same basis

Polarized photon
V, H, D, A
(loss)

Bob

Choose random decoding basis:
rectangular or diagonal

Decode polarized photon to key bit

Announce arrivals and chosen decoding basis

Keep key bit if same basis

BB84 is generally implemented in the following steps:

- Alice sends a series of key bits to Bob over the quantum channel.
- For each key bit, Alice first randomly chooses the bit value, which is zero or one.
- Then Alice also randomly chooses the encoding basis, which is rectilinear or diagonal.

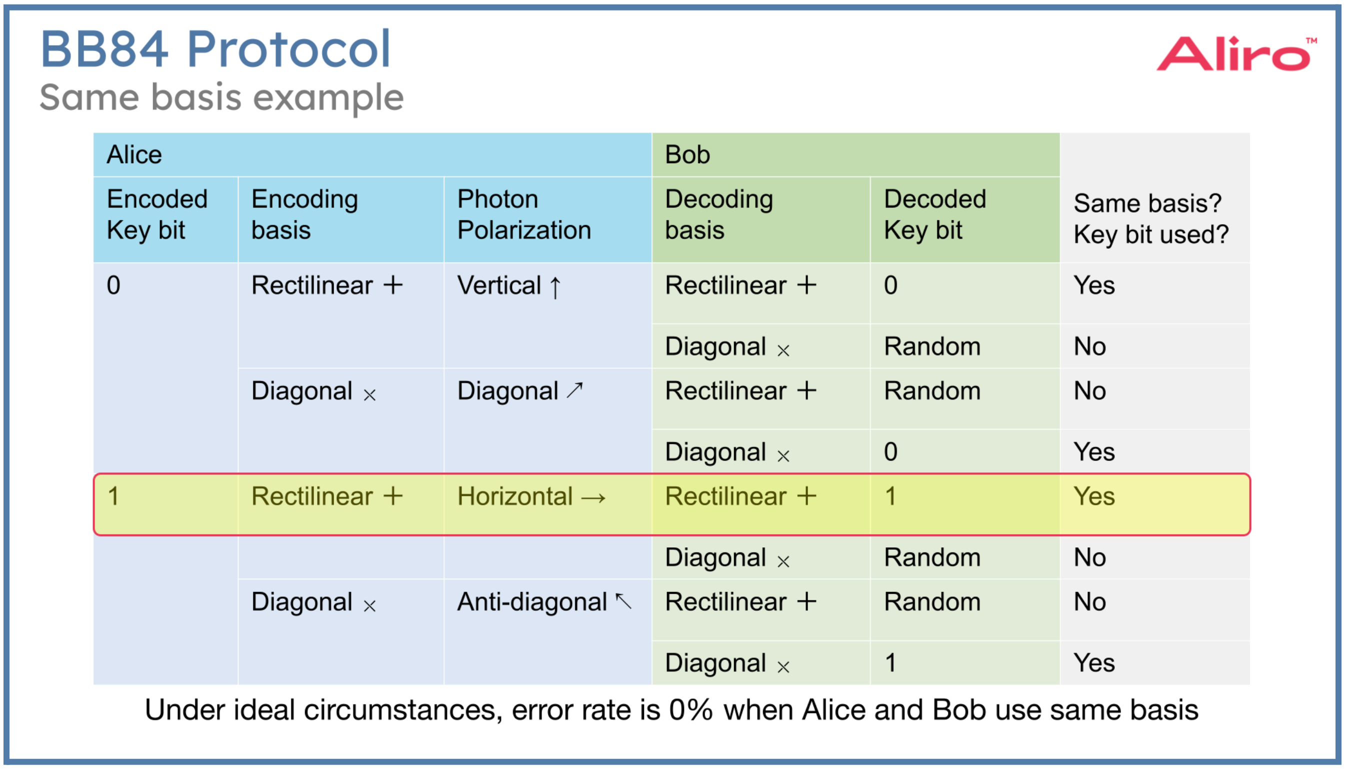
- Alice then uses polarization encoding to encode the bit value into a single photon, using the chosen encoding basis.
- This results in one of four possible polarizations of the photon: vertical, horizontal, diagonal, or anti-diagonal.
- Alice sends the encoded photon to Bob, over optical fiber or free space.
- Because of loss, only some of the photons sent by Alice actually arrive at Bob. For each photon that arrives at Bob, Bob randomly chooses a decoding basis: rectilinear or diagonal.
- Bob uses polarization decoding to decode the received photon into a bit value, using the chosen decoding basis.
- If Bob happens to choose the correct basis, meaning the same as Alice, then Bob gets the correct key bit. If Bob chooses the wrong basis, he gets random junk bit values.
- After Bob has received and decoded the photons, Alice and Bob exchange information about what polarization they used for each photon and which photons actually arrived. These announcements happen over a public, authenticated channel.
- The final step is that Alice and Bob discard all the key bits that either Bob never received or for which Bob used the wrong decoding basis. This process is called basis sifting.

If there wasn't any noise, Alice and Bob would end up with identical keys, but in real life there is noise. One source of noise is just random fluctuations in the environment. Another source of noise is caused by attacker Eve trying to steal the key, as we will discuss later. We will also discuss how Alice and Bob use classical post-processing to discover and correct noise and detect Eve’s eavesdropping.

Examples of the BB84 Protocol

The two examples discussed here will clarify how the BB84 protocol works.

In the first example, the highlighted row shows that Alice and Bob happened to choose the same encoding basis, which is the rectilinear basis in this case.



BB84 Protocol

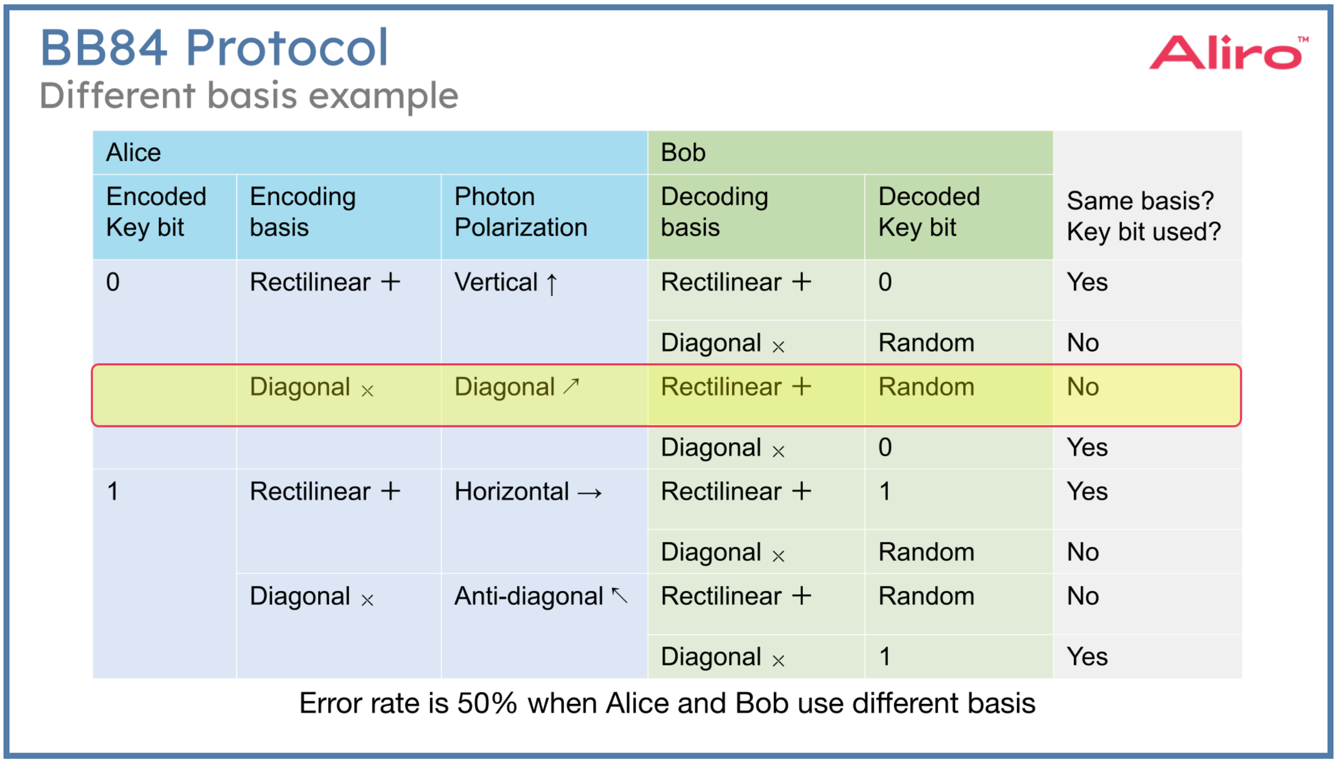
Same basis example

Alice			Bob		
Encoded Key bit	Encoding basis	Photon Polarization	Decoding basis	Decoded Key bit	Same basis? Key bit used?
0	Rectilinear +	Vertical ↑	Rectilinear +	0	Yes
			Diagonal ×	Random	No
	Diagonal ×	Diagonal ↗	Rectilinear +	Random	No
			Diagonal ×	0	Yes
1	Rectilinear +	Horizontal →	Rectilinear +	1	Yes
			Diagonal ×	Random	No
	Diagonal ×	Anti-diagonal ↖	Rectilinear +	Random	No
			Diagonal ×	1	Yes

Under ideal circumstances, error rate is 0% when Alice and Bob use same basis

As a result, Bob decodes the correct key bit value, which is key bit value one in this case. During classical post-processing (discussed later in more detail later in this white paper) it is publicly announced that Bob actually received the photon and used the correct decoding basis. As a result, Alice and Bob each decide not to discard the key bit and keep it as part of the key.

In the second example, the highlighted row shows that Bob happened to choose the wrong decoding basis, meaning the opposite basis from Alice.



BB84 Protocol

Different basis example

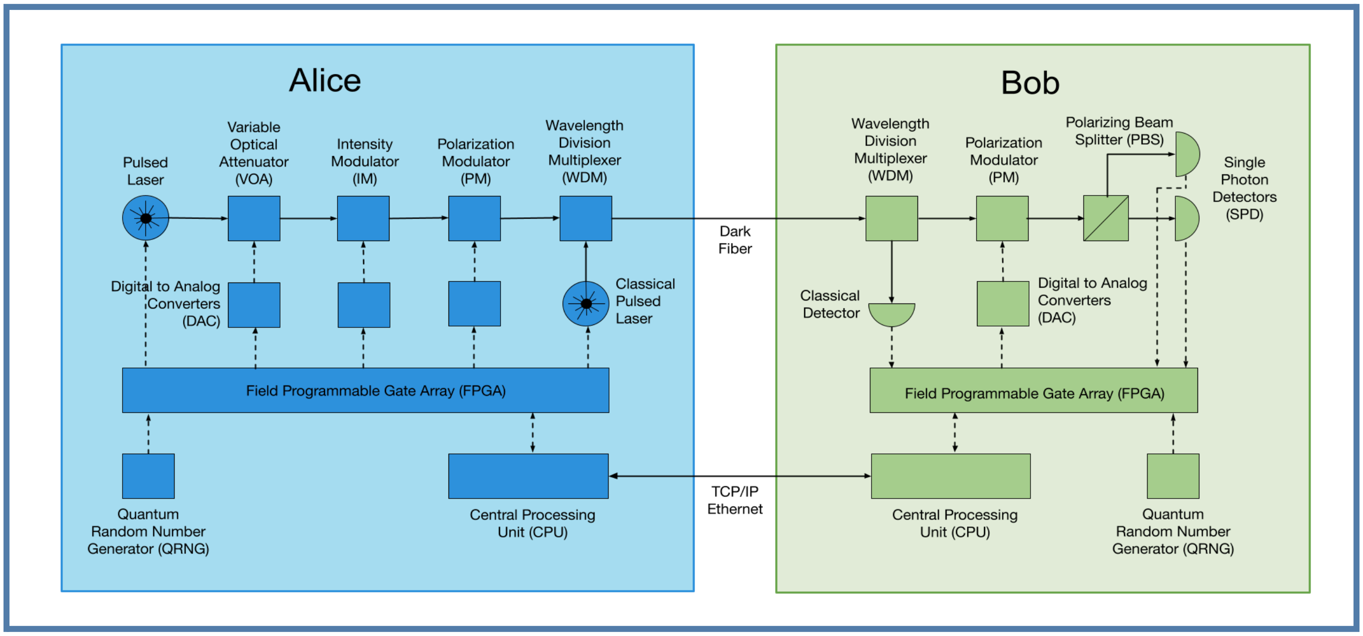
Alice			Bob		
Encoded Key bit	Encoding basis	Photon Polarization	Decoding basis	Decoded Key bit	Same basis? Key bit used?
0	Rectilinear +	Vertical ↑	Rectilinear +	0	Yes
			Diagonal ×	Random	No
	Diagonal ×	Diagonal ↗	Rectilinear +	Random	No
			Diagonal ×	0	Yes
1	Rectilinear +	Horizontal →	Rectilinear +	1	Yes
			Diagonal ×	Random	No
	Diagonal ×	Anti-diagonal ↖	Rectilinear +	Random	No
			Diagonal ×	1	Yes

Error rate is 50% when Alice and Bob use different basis

In this case Alice chose diagonal basis encoding and Bob chose rectilinear basis measurement. As a result, Bob decodes a random key bit value, which has nothing to do with the bit value that Alice sent. During classical post-processing it is publicly announced that Bob received the photon but used the wrong decoding basis. As a result, Alice and Bob each discard the key bit and don't use it as part of the key.

BB84 QKD Device Implementation

The diagram below shows a simplified representation of how a BB84 QKD device using polarization encoding could be implemented.



Alice
Pulsed Laser
Variable Optical Attenuator (VOA)
Intensity Modulator (IM)
Polarization Modulator (PM)
Wavelength Division Multiplexer (WDM)
Digital to Analog Converters (DAC)
Classical Pulsed Laser
Field Programmable Gate Array (FPGA)
Quantum Random Number Generator (QRNG)
Central Processing Unit (CPU)
Dark Fiber
TCP/IP Ethernet
Bob
Wavelength Division Multiplexer (WDM)
Polarization Modulator (PM)
Polarizing Beam Splitter (PBS)
Single Photon Detectors (SPD)
Classical Detector
Digital to Analog Converters (DAC)
Field Programmable Gate Array (FPGA)
Central Processing Unit (CPU)
Quantum Random Number Generator (QRNG)

On the left side of Alice's block, a pulsed laser is used to generate light pulses. It is followed by a Variable Optical Attenuator to weaken the laser pulse down to the single photon level. Next in line is an intensity modulator which is used to implement decoy states, a concept which will be discussed later. After that is a polarization modulator, which is used to choose one of the four possible polarizations to encode the chosen encoding basis and bit value.

On Bob's side, the received light pulses go through another polarization modulator to choose the decoding basis. After that, they go through a polarizing beam splitter and a pair of single photon detectors to do the actual measurement in the chosen basis.

Alice and Bob each have an FPGA, a CPU, and a quantum random number generator to control the various components and to make real random choices. Also shown here is a classical pulsed laser at Alice and a classical detector at Bob. These are used to implement the real-time classical communication used for synchronization, calibration, sifting, etc. The non-real time protocols such as information reconciliation (error correction), privacy amplification, management, and key delivery typically happen over a standard TCP/IP Ethernet interface. The classical channels may or may not be wave division multiplexed on the same fiber as the quantum channel.

BB84 Cloning Attack

How does BB84 protect against Eve stealing the key?

The formal security proof is rather complex, but this explanation will hopefully give you an intuitive understanding.

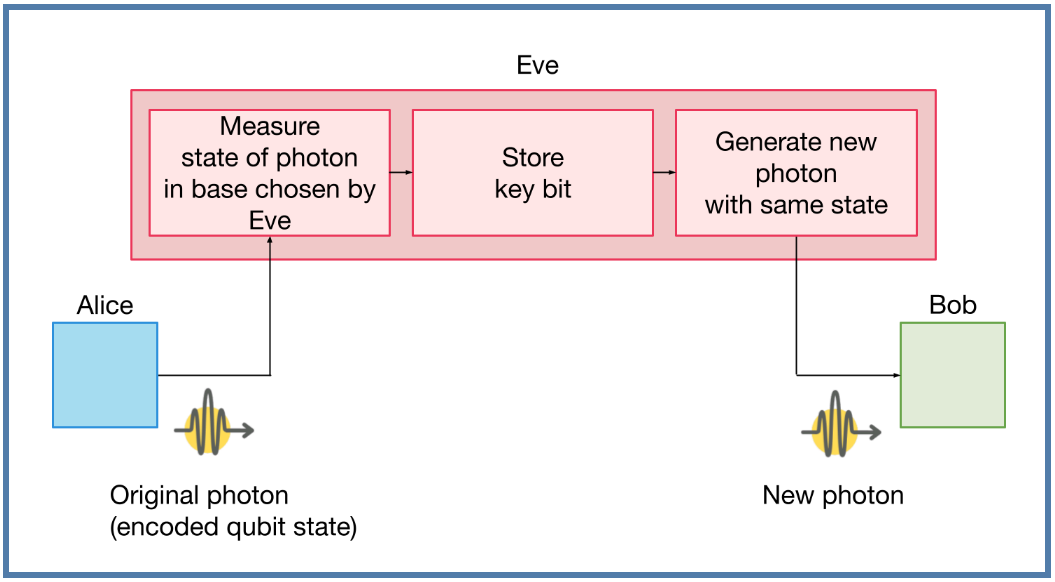
First, consider an attack where Eve makes a copy of each photon which Alice sends to Bob. Eve stores these copied photons in quantum memories. Eve then waits for Alice and Bob to publicly announce which photons were received and which encoding basis they used. Finally, Eve uses that information to measure the correct set of copied photons in the correct basis.

Such a scheme, if it actually worked, would allow Eve to discover the key. Fortunately, this attack does not actually work because of a law in quantum mechanics which is called the no-cloning theorem.

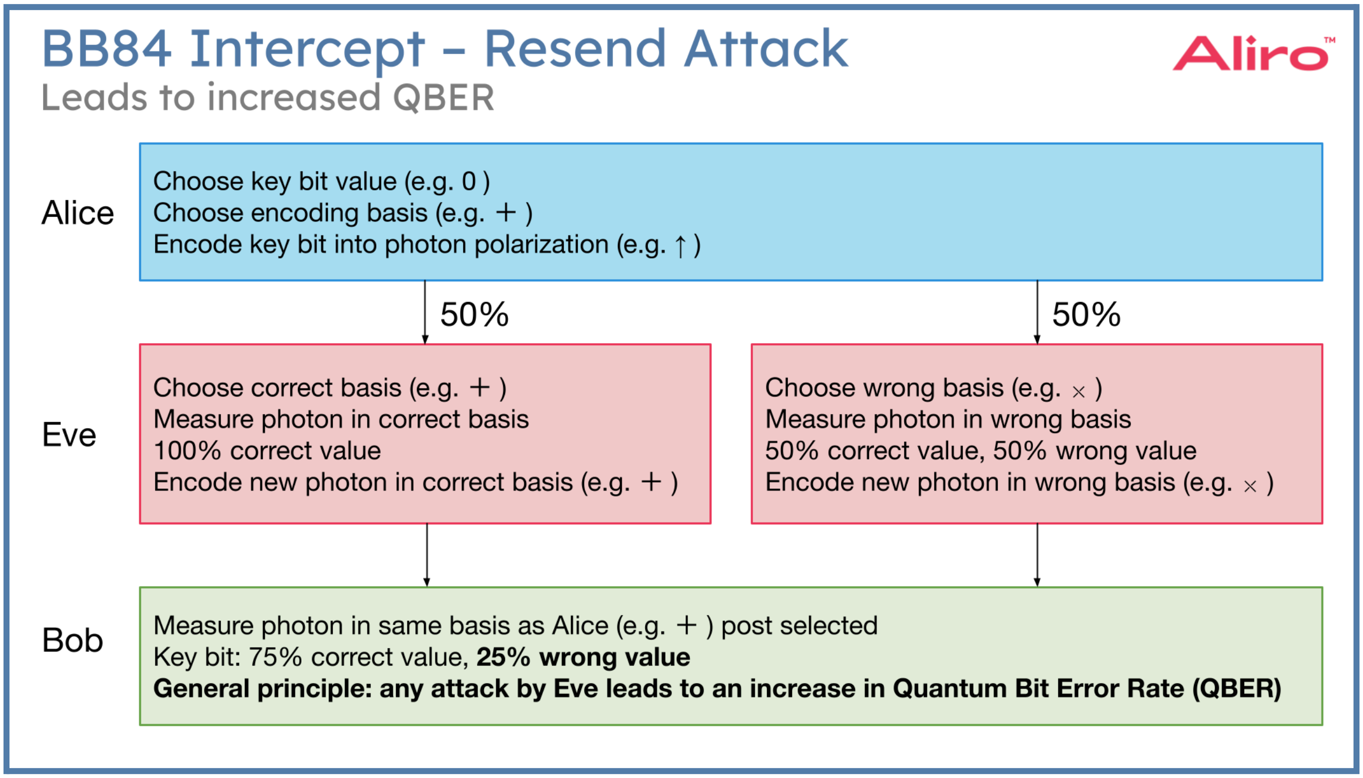
According to the no-cloning theorem, it is impossible to reliably make a copy of a photon in some arbitrary unknown polarization state. Note that this law is only applicable if Eve does not know the encoding basis of the photon. If Eve does know the encoding basis, then she is able to make a copy. For this reason, it is important for Alice and Bob to only make their public announcements after the photons have already been received and decoded by Bob.

BB84 Intercept-Resend Attack

Now consider a more sophisticated attack by Eve: the so-called intercept-resend attack.


Eve
Measure state of photon in base chosen by Eve
Store key bit
Generate new photon with same state
Alice
Bob
Original photon (encoded qubit state)
New photon

Here, instead of trying to copy the photons, Eve measures the photons. At this moment in time, Eve doesn't have any information about which basis Alice used to encode the photon: rectilinear or diagonal. The most Eve can do is to choose the decoding basis randomly. Eve stores the result of the measurement, which is a classical bit. After that, Eve creates a new photon and encodes this classical bit using the same randomly chosen basis that was used for decoding. This newly created photon is sent to Bob.



BB84 Intercept – Resend Attack

Leads to increased QBER

Aliro™

Alice

Choose key bit value (e.g. 0)
Choose encoding basis (e.g. +)
Encode key bit into photon polarization (e.g. ↑)

50%

50%

Eve

Choose correct basis (e.g. +)
Measure photon in correct basis
100% correct value
Encode new photon in correct basis (e.g. +)

Choose wrong basis (e.g. ×)
Measure photon in wrong basis
50% correct value, 50% wrong value
Encode new photon in wrong basis (e.g. ×)

Bob

Measure photon in same basis as Alice (e.g. +) post selected
Key bit: 75% correct value, **25% wrong value**
General principle: any attack by Eve leads to an increase in Quantum Bit Error Rate (QBER)

The diagram above shows what happens when Eve uses the intercept-resend attack to try to steal the keys. When Eve measures Alice's photons, Eve does not yet know what the correct measurement basis is, because it has not yet been publicly announced by Alice and Bob. As a result, Eve has a 50% chance of guessing the correct basis, the same basis that Alice originally encoded.

If Eve guesses the wrong basis, Eve will get a completely random bit value. Eve will also send a photon encoded in the wrong basis to Bob, which means that Bob will also get a completely random bit value, even when Bob has chosen the same basis as Alice.

Going through all the details and the math will show that Bob ends up with a Quantum Bit Error Rate of 25%.

Looking at this scenario from a different perspective, if Alice and Bob notice a Quantum Bit Error Rate of 25%, they must conclude that Eve is present and trying to steal the key. The formal security proof, which takes all possible attacks into account, not just a few example attacks, is quite complex but it does exist for the BB84 protocol. It turns out that the actual cut-off Quantum Bit Error Rate (above which no usable keys can be produced) is somewhere around 11%.

The important point here is that the laws of quantum physics guarantee that the Quantum Bit Error Rate increases in a well-defined and detectable manner when Eve is present and attempting to steal the keys.

Once this process of encoding, sending, and decoding has been completed, classical post-processing is performed. All key distribution protocols discussed in this white paper undergo classical post-processing. This topic is discussed in more detail later in the paper.

QSC Protocols

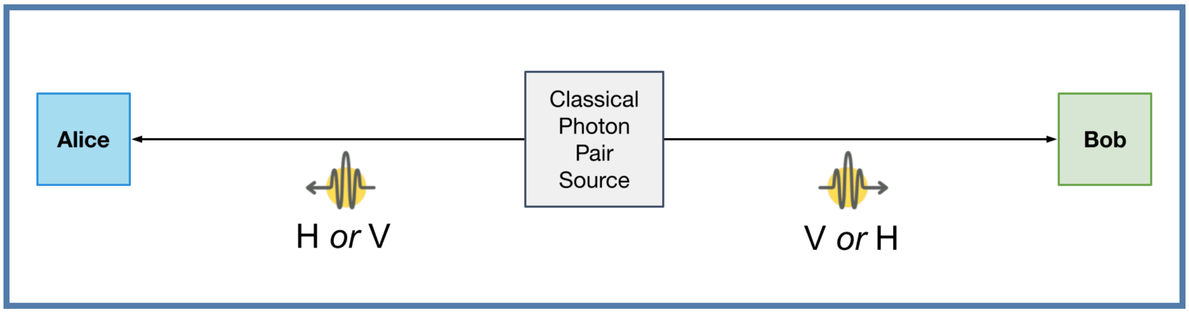
QSC protocols, such as E91 and BBM92, use entanglement to generate secure keys. Entangled photons are sent over the quantum channel in a network, one to Alice and one to Bob. Entanglement occurs when two or more particles become interconnected such that the state of one particle instantaneously affects the state of the other, regardless of distance. This property is exploited in QSC to generate secure cryptographic keys, ensuring that any eavesdropping attempt disrupts the entangled state and is detectable.

Classical Correlation versus Entanglement

Classical Correlation

Quantum entanglement is a difficult concept to fully grasp. In particular, it should not be confused with classical correlation.

Let's consider the classical source that is shown in the diagram below.

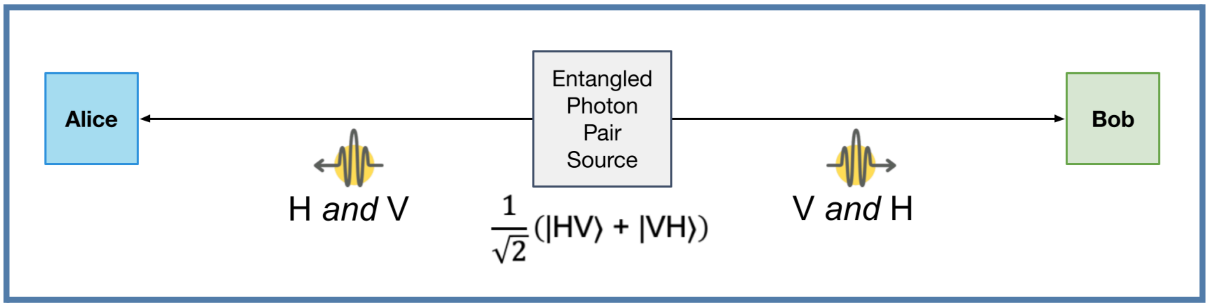


When it emits a photon pair, there is a 50 percent probability that the photon going left is horizontally polarized and the photon going right is vertically polarized. There is also a 50 percent probability that it is the other way around: the photon going left is vertically polarized and the photon going right is horizontally polarized. The important concept here is that as soon as the photon pair leaves the source it is already one or the other. Of course, Alice and Bob don't know which it is until they actually measure the photon after it arrives, but that doesn't change the fact that it was already one or the other as soon as the photon pair left the source and before Alice and Bob did the measurement.

Here, the state is simply hidden until Alice and Bob do the measurement. This is not quantum entanglement. This is classical correlation.

Entanglement

What, then, is entanglement? In the diagram below, there is a source that produces entangled photon pairs.


Alice
Entangled Photon Pair Source
Bob
H *and* V
$\frac{1}{\sqrt{2}}(|HV\rangle + |VH\rangle)$
V *and* H

In the previous diagram, each photon was *either* horizontally polarized *or* vertically polarized as soon as it left the source. In this diagram, each photon is in some sense *both* horizontally polarized *and* at the same time also vertically polarized. It only becomes one or the other when someone, either Alice or Bob, does the first measurement.

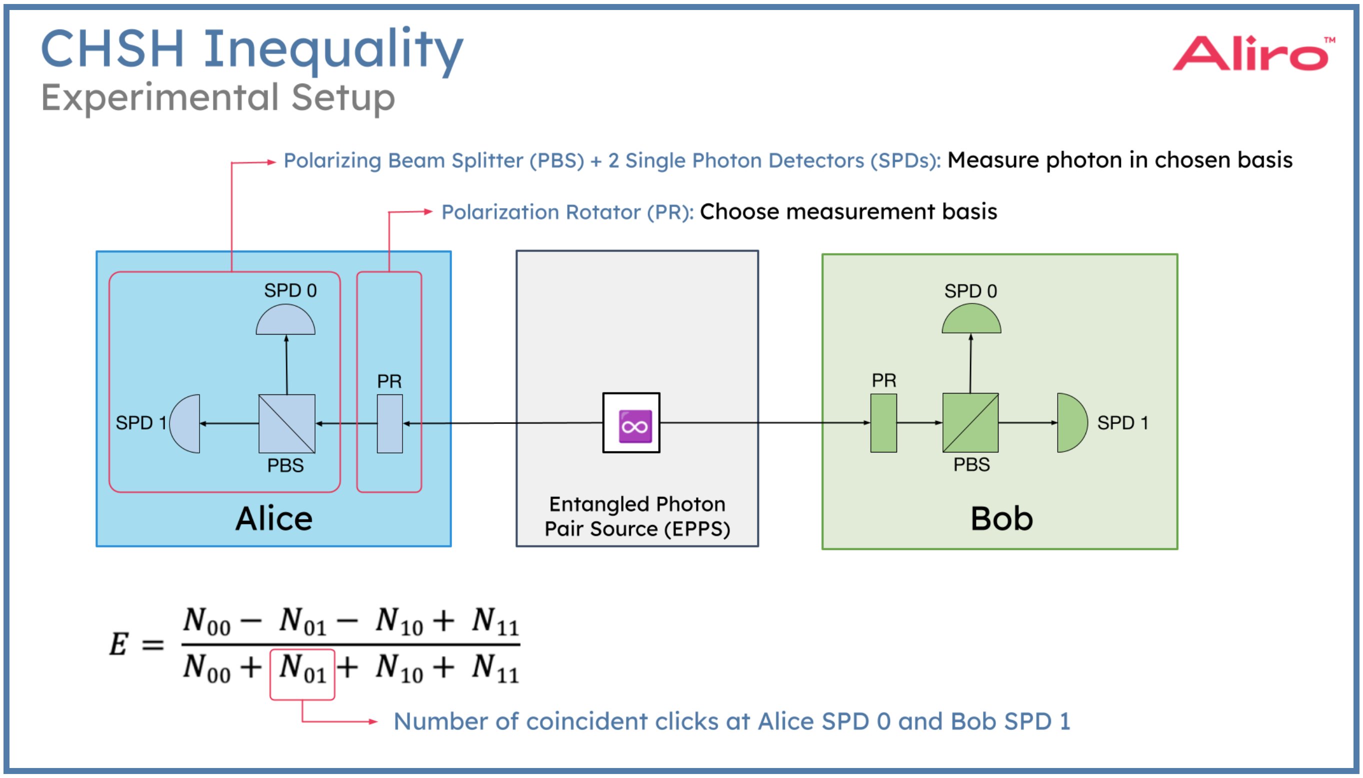
Here, the state of the photons is described by a wave function, and the wave function collapses into a specific state when an observation (a measurement) is made. When Alice measures their photon, it causes that photon to randomly collapse into one or the other polarization. This then causes the photon traveling to Bob to immediately collapse into the orthogonal polarization. This happens instantaneously.

Distinguishing between classical correlation and quantum entanglement: the Bell Test

In 1964, physicist John Steward Bell devised an experiment that can distinguish classical correlation from quantum entanglement. This experiment can determine whether quantum entanglement really exists or not. The original experiment is called a Bell test. Later, a simpler version of the experiment was discovered called the CHSH inequality after its inventors.

CHSH Inequality

The experimental setup for the CHSH Inequality is very simple and shown below.



CHSH Inequality

Experimental Setup

Polarizing Beam Splitter (PBS) + 2 Single Photon Detectors (SPDs): Measure photon in chosen basis

Polarization Rotator (PR): Choose measurement basis

$$E = \frac{N_{00} - N_{01} - N_{10} + N_{11}}{N_{00} + \boxed{N_{01}} + N_{10} + N_{11}}$$

Number of coincident clicks at Alice SPD 0 and Bob SPD 1

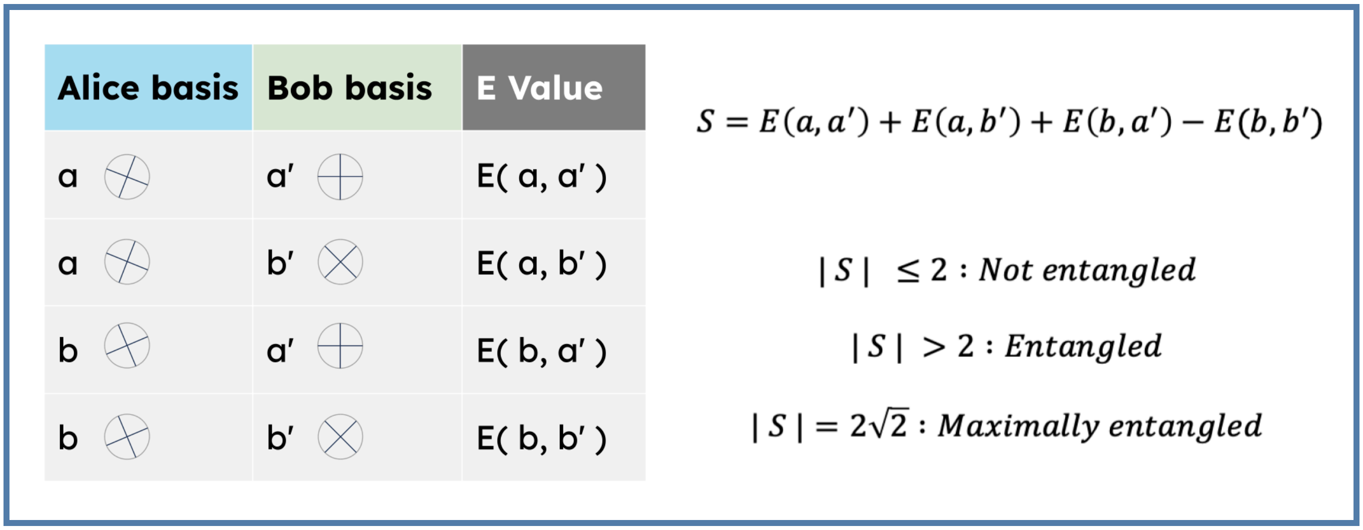
There is a photon pair source in the middle, and the goal is to determine whether the photons that it produces are entangled or not. To find out, Alice and Bob each perform a series of Bell state measurements using a polarizing beam splitter and two single photon detectors.

Alice and Bob each also have a Polarization Rotator which allows them to independently choose their measurement basis. For each series of measurements, the *coincident* click counts are tracked. For example, N_{10} is the number of times that the 1-Detector at Alice clicked *at the same time* as the 0-Detector at Bob.

At the end of a series of coincident clicks, the value E is calculated using the formula shown at the bottom of the diagram above.

The polarization rotators in the diagram allow Alice and Bob to independently choose their measurement basis. Alice randomly chooses a measurement basis for each arriving photon: either 22.5 degrees or minus 22.5 degrees. Bob also randomly chooses a measurement basis for each arriving photon, but selects either 0 degrees or 45 degrees.

Given that Alice and Bob each randomly choose between two measurement bases, there are four possible combinations.



Alice basis	Bob basis	E Value
a ⊗	a′ ⊕	E(a, a′)
a ⊗	b′ ⊗	E(a, b′)
b ⊗	a′ ⊕	E(b, a′)
b ⊗	b′ ⊗	E(b, b′)

$$S = E(a,a') + E(a,b') + E(b,a') - E(b,b')$$

$$|S| \leq 2 : \mathit{Not\ entangled}$$

$$|S| > 2 : \mathit{Entangled}$$

$$|S| = 2\sqrt{2} : \mathit{Maximally\ entangled}$$

The E value can be computed separately for each possible combination of bases. Then, those four values of E can be plugged into this formula for S shown on the right. If the absolute value of S is two or less, it means that the photons are not entangled, they are only classically correlated. If the absolute value of S is greater than two, it proves that the photons are quantum entangled. The higher the value of S above 2, the more entangled the photons are. When S is two times the square root of 2, the photons are maximally entangled.

In 2022 three gentlemen, Alain Aspect, John Clauser, and Anton Zeilinger, were awarded the Nobel prize in physics for performing the CHSH experiment proving that entanglement is real.

Monogamy of Entanglement

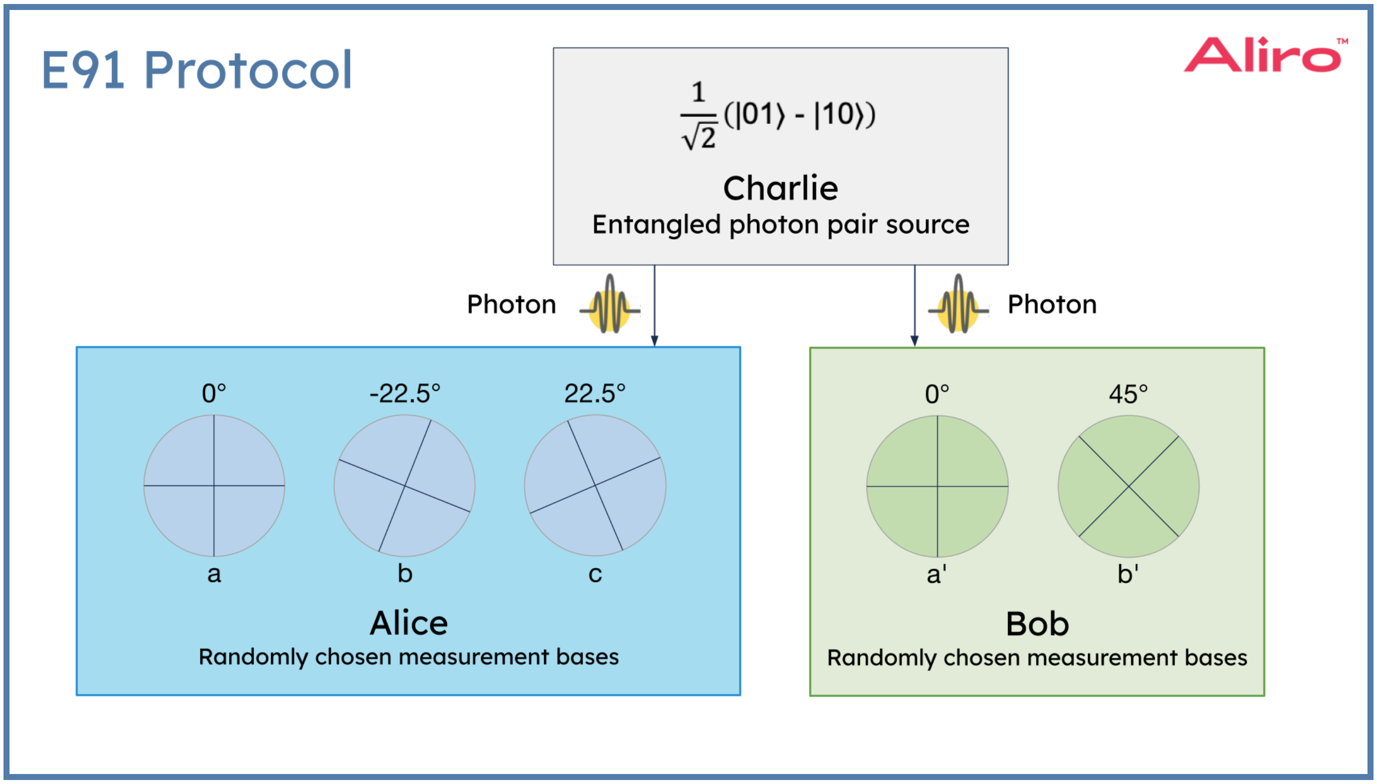
There is one more concept that is central to QSC protocols: monogamy of entanglement.

Earlier, we stated that two photons are *maximally* entangled if the S value in the CHSH experiment is equal to 2 times the square root of 2. This is also referred to as maximal bi-partite entanglement. The four Bell states are examples of such entanglement.

The concept of monogamy of entanglement says that if two photons are maximally entangled with each other, then it is impossible for a third photon to be entangled as well. This is central to the security of QSC protocols.

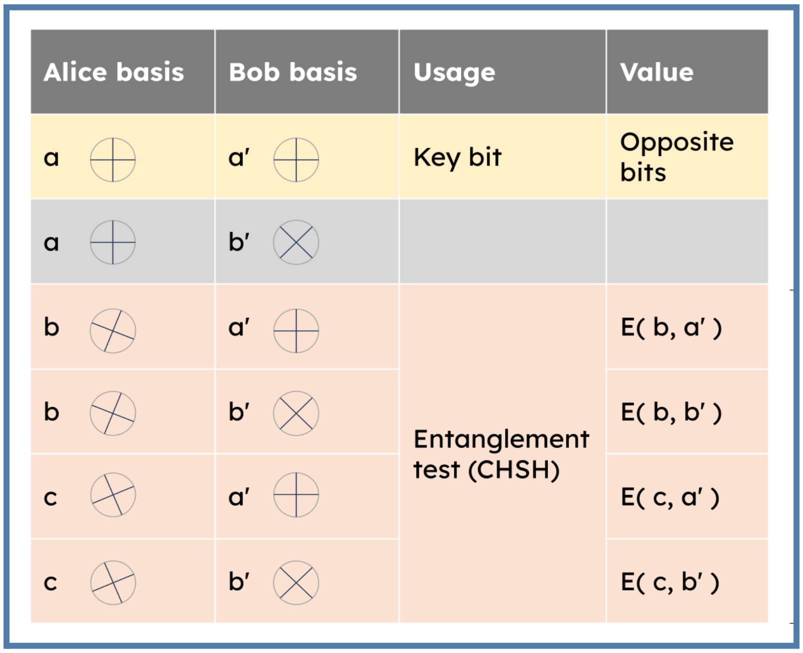
E91 Protocol

In Ekert 1991 protocol, or E91 protocol, Alice and Bob do something very similar to the CHSH experiment, except that Alice chooses from three random measurement bases instead of two.


E91 Protocol
Aliro™
$\frac{1}{\sqrt{2}}(|01\rangle - |10\rangle)$
Charlie
Entangled photon pair source
Photon
Photon
0°
-22.5°
22.5°
a
b
c
Alice
Randomly chosen measurement bases
0°
45°
a'
b'
Bob
Randomly chosen measurement bases

This leads to six possible combinations of measurement bases at Alice and Bob. In the table below, Alice and Bob are using the E91 protocol. In the first row, in yellow at the top of the table, Alice and Bob choose the same measurement basis. Because this example is using the psi-minus state, Alice's and Bob's measurements will always be opposite. If Bob flips their bit, Alice and Bob can agree on a key bit

Four of the basis choice combinations, those shown in red at the bottom of the table below, are exactly the CHSH experiment that was described above.



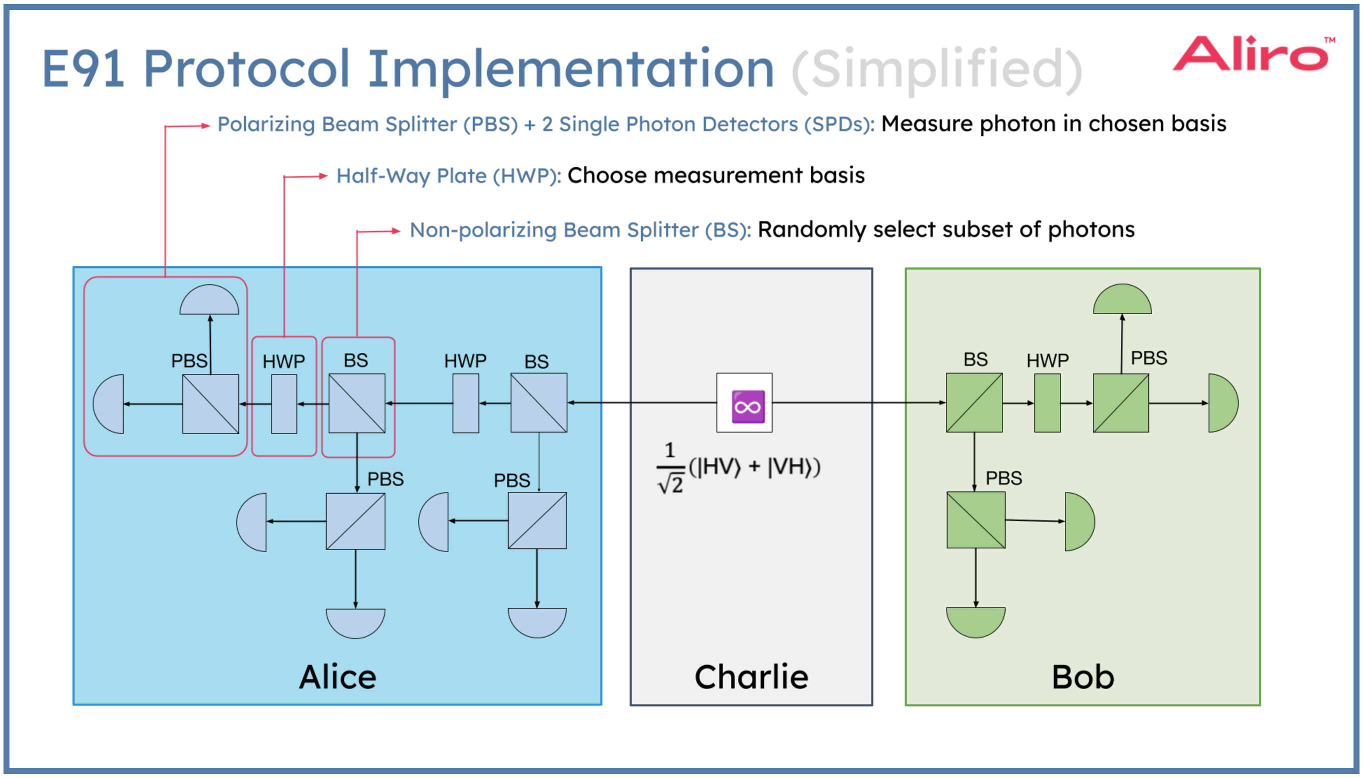
Alice basis	Bob basis	Usage	Value
a ⊕	a′ ⊕	Key bit	Opposite bits
a ⊕	b′ ⊗		
b	a′ ⊕	Entanglement test (CHSH)	E(b, a′)
b	b′ ⊗		E(b, b′)
c	a′ ⊕		E(c, a′)
c	b′ ⊗		E(c, b′)

This allows Alice and Bob to compute the S value and determine whether the photons they receive are entangled or not. If the S value is two times the square root of two, it shows that the photons are maximally entangled. Because of the monogamy of entanglement this implies that Eve cannot participate in the entanglement and hence the key is safe. If the S value is greater than two but less than two times the square root of two, it shows that the photons *are* entangled but not *maximally* entangled.

The amount of information that was leaked to Eve can be determined using parameter estimation. Depending on how much information was leaked, Alice and Bob can either erase the leaked information using privacy amplification or conclude that the key cannot be used.

E91 QSC Device Implementation

The diagram below shows how the E91 protocol can be implemented using polarization encoding.



E91 Protocol Implementation (Simplified)

Polarizing Beam Splitter (PBS) + 2 Single Photon Detectors (SPDs): Measure photon in chosen basis

Half-Way Plate (HWP): Choose measurement basis

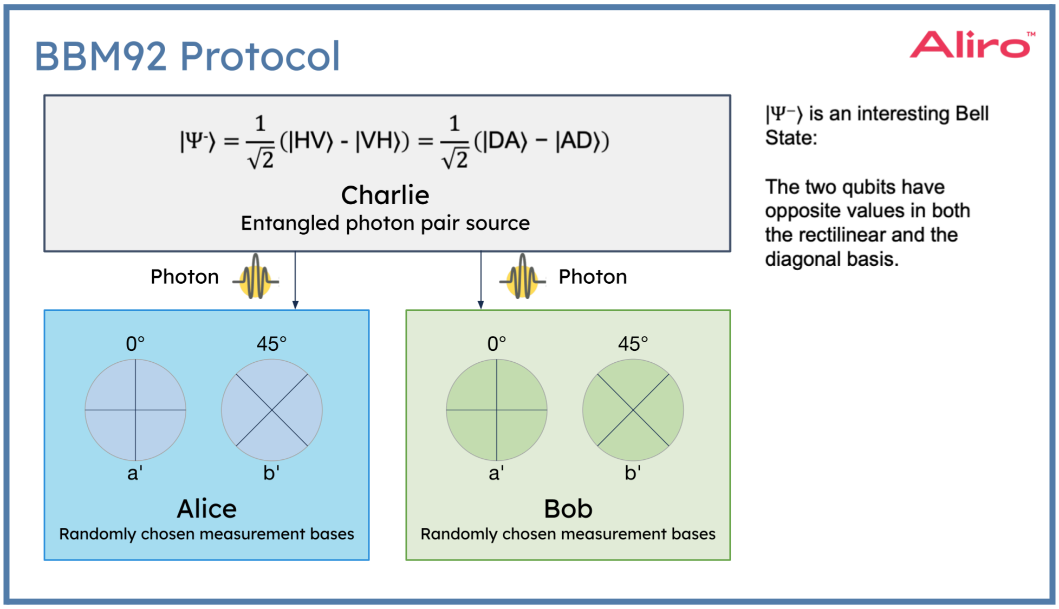
Non-polarizing Beam Splitter (BS): Randomly select subset of photons

At Alice, two normal non-polarizing beam splitters are used to randomly split the arriving photons into three groups. Similarly, at Bob one beam splitter is used to randomly split the arriving photon into two groups.

For each group, Alice and Bob use a half-way plate to rotate the polarization into the chosen measurement basis. They then use a polarizing beam splitter with two single photon detectors to make the actual measurement in the chosen basis. There are many practical details including calibration and synchronization that are not shown in this simplified design.

BBM92 Protocol

The BBM92 protocol uses an entangled photon pair source that produces photons in a particular entangled state, namely the Psi-minus state. The Psi-minus state is a very interesting state.



BBM92 Protocol

$$|\Psi^{-}\rangle = \frac{1}{\sqrt{2}}(|HV\rangle - |VH\rangle) = \frac{1}{\sqrt{2}}(|DA\rangle - |AD\rangle)$$

Charlie

Entangled photon pair source

Alice

Randomly chosen measurement bases

Bob

Randomly chosen measurement bases

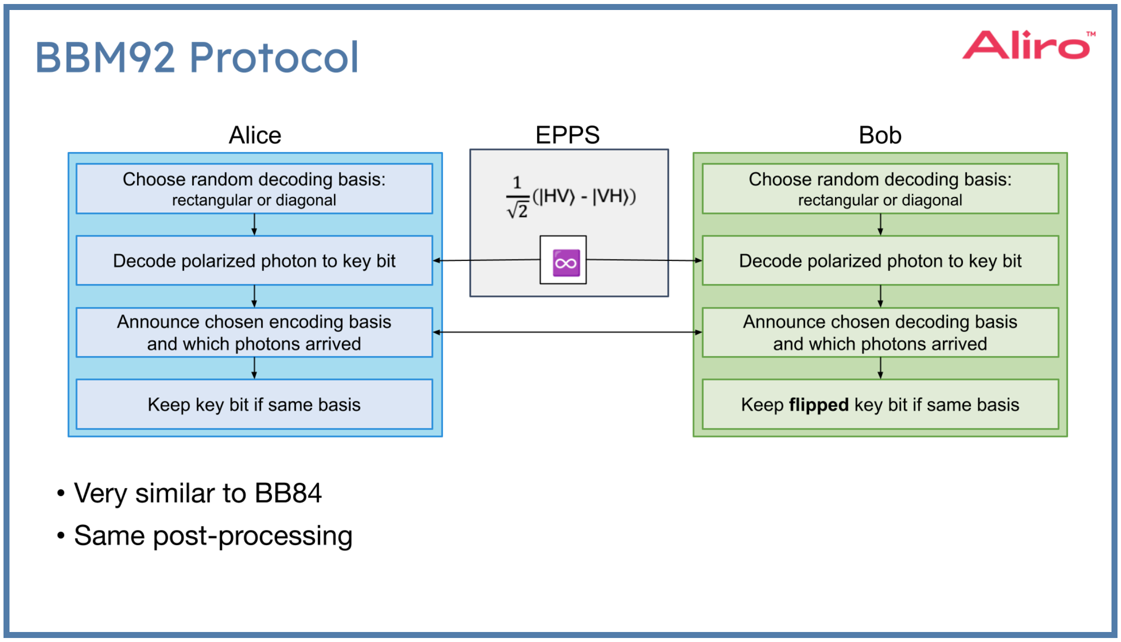
$|\Psi^{-}\rangle$ is an interesting Bell State:

The two qubits have opposite values in both the rectilinear and the diagonal basis.

If Alice and Bob both measure their photon in the rectilinear basis, they always get opposite bit values. If Alice and Bob both measure their photon in the diagonal basis, they *still* always get opposite bit values.

Alice and Bob randomly choose between the rectilinear and diagonal basis when they measure arriving photons.

The BBM92 protocol can be thought of as an entanglement-based variation of the BB84 prepare-and-measure protocol. (See diagram on next page.) After Alice and Bob have done their measurements, they perform coincidence and basis sifting. They announce their chosen measurement bases over the public classical channel and discard all key bits for which they chose a different decoding basis.



BBM92 Protocol

Aliro™

Alice

Choose random decoding basis: rectangular or diagonal

Decode polarized photon to key bit

Announce chosen encoding basis and which photons arrived

Keep key bit if same basis

EPPS

$\frac{1}{\sqrt{2}}(|HV\rangle - |VH\rangle)$

∞

Bob

Choose random decoding basis: rectangular or diagonal

Decode polarized photon to key bit

Announce chosen decoding basis and which photons arrived

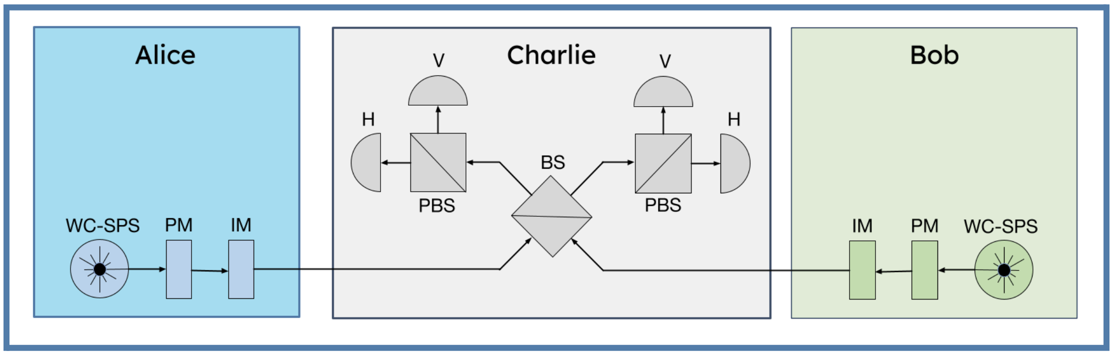
Keep **flipped** key bit if same basis

- Very similar to BB84
- Same post-processing

Interference-Based Protocols

Measurement Device Independent (MDI) QKD Protocol

In MDI-QKD, Alice and Bob each send a stream of light pulses to a midpoint station which is typically called Charlie. Alice and Bob each choose a random key bit value and a random encoding basis for each sent pulse. The light pulses go through a polarization modulator to encode the chosen bit value and encoding basis into one of four polarizations: horizontal, vertical, diagonal, or anti-diagonal. There are also intensity modulators to implement decoy states.[MDI1, 2]



The light pulses, one sent by Alice, and one sent by Bob, arrive at the midpoint station Charlie, which performs a Bell state measurement. If Charlie sees two clicks on two different detectors at the same time, then both photons arrived. This is called a coincident click. Charlie is looking for a particular coincident click pattern: namely one click on a vertical detector, and the other click on a horizontal detector. Such a click pattern is called a successful Bell state measurement.

Charlie reports all successful Bell state measurements back to Alice and Bob. Alice and Bob use basis sifting to only keep the Bell state measurement for which they used the same encoding basis. It is relatively easy to see that this results in a stream of opposite bit values at Alice and Bob. If Bob flips their bits, Alice and Bob will end up with a shared key.

Charlie knows which Bell state measurements were successful, but also only knows that Alice and Bob have opposite bit values; Charlie has no knowledge of whether Alice has the zero bit or Bob has the zero bit. Hence, Charlie doesn't know anything about the key.

One important detail is that for the Bell state measurement at Charlie to work properly, the Hong-Ou-Mandel effect must occur. This, in turn, requires that the two photons arriving at Charlie from Alice and Bob are identical in every aspect, except for possibly the polarization. They must have the exact same frequency. They must arrive at the exact same time, down to the sub-nanosecond. For this reason, implementing calibration and synchronization for MDI QKD systems is quite challenging.

One interesting aspect of MDI QKD is that there is no need to trust the mid-point Charlie. If attacker Eve has compromised Charlie, this does not affect the security of the total system. How is this possible? First, Charlie does not have any information about the key bits. Second, if Eve manipulates Charlie to do anything other than a real Bell state measurement, or sends any type of fake results, this can be detected by Alice and Bob based on the statistics of the success reports.

This is where the name "Measurement Device Independent" QKD comes from. MDI QKD is immune against any side-channel attack on the detectors, because the detectors are only located at Charlie and not at Alice and Bob. There are also two variations of MDI QKD that promise further improvements, but that are still in the research and development stage: Twin Field QKD and Device Independent QKD.

Twin Field QKD is a variation on MDI QKD which greatly increases the maximum distance of the QKD link. Twin Field QKD systems have been demonstrated in proofs of concept, but commercial products are still under development because the calibration is much more challenging than MDI QKD.

Device Independent QKD, or DI QKD, is not only immune to side-channel attacks on the single-photon detectors but also immune to side-channel attacks against the single-photon sources. The theory around fully Device Independent QKD is well developed, but as of this writing, there are no implementations yet that achieve useful key rates.

Classical Post Processing

Up to this point, the focus has been on the quantum aspects of QKD, QSC, and IQKD.

Regardless of the protocol used, classical post-processing is performed. This is done over the classical channel of the network.

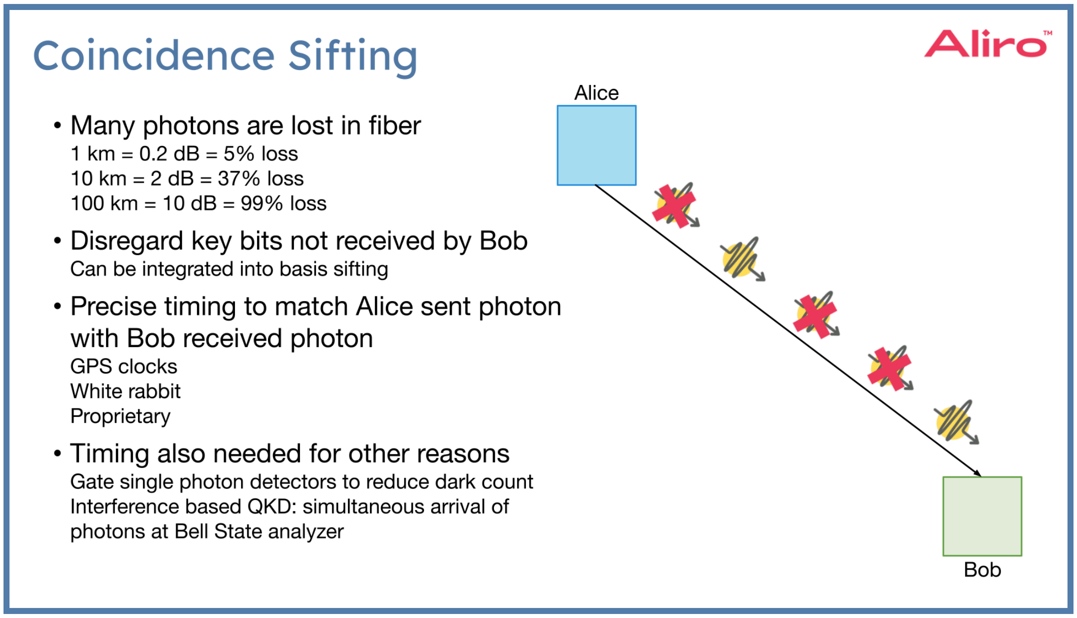
Classical post-processing consists of five steps:

1. Coincidence sifting, where we disregard all key bits for which Bob failed to receive the photon due to loss in the fiber.
2. Basis sifting, where we disregard all key bits for which Alice and Bob chose different encoding bases. Academic literature typically combines coincidence and basis sifting into one single step, but in real implementations they are sometimes separated for practical reasons.
3. Parameter estimation, which is also known as error estimation. Here Alice and Bob sacrifice and publicly announce a subset of the sifted key bits. This allows them to estimate the quantum bit error rate. This is needed to detect the presence of eavesdropper Eve. It is also useful to select the optimal codes for the next two steps.
4. Information reconciliation, which is also known as error correction. Alice and Bob detect and correct any bit errors in the key.
5. Privacy amplification, where any information that was leaked to Eve is erased.

Next we’ll look at these steps in more detail.

Coincidence Sifting

In this diagram, Alice sends a sequence of photons to Bob. In a more realistic example, Alice would be sending many more photons to Bob at a very high rate, possibly a billion photons per second.



Coincidence Sifting

Aliro™

- Many photons are lost in fiber
  1 km = 0.2 dB = 5% loss
  10 km = 2 dB = 37% loss
  100 km = 10 dB = 99% loss
- Disregard key bits not received by Bob
  Can be integrated into basis sifting
- Precise timing to match Alice sent photon with Bob received photon
  GPS clocks
  White rabbit
  Proprietary
- Timing also needed for other reasons
  Gate single photon detectors to reduce dark count
  Interference based QKD: simultaneous arrival of photons at Bell State analyzer

Due to loss in the fiber, which increases exponentially with distance, many or even most of the photons don't make it from Alice to Bob. For example, in 10 kilometers of fiber, 37% of the photons are lost. In 100 kilometers of fiber, 99% of the photons are lost.

In this example, Alice sends 5 photons, 3 are lost, and Bob receives only 2 photons. Alice and Bob need to agree on which photons were received and which were lost. This is needed because they both need to discard the key bits which correspond to the lost photons.

This is more difficult than it sounds, because unlike classical messages, the photons don't have header fields to carry a sequence number. Instead, Alice and Bob typically use time slots, very precise clocks, and very precise synchronization to identify the photons. A variety of technologies are used to implement the clock synchronization, including GPS clocks, standard protocols such as White Rabbit, or proprietary protocols over the real-time classical channel. We will skip basis sifting, as that process has already been discussed in detail previously in this white paper, and move on to information reconciliation, also known as error correction.

Information Reconciliation (Error Correction)

After Alice and Bob perform coincidence sifting and basis sifting, they have what is commonly referred to as a raw key. This raw key still contains some errors due to noise on the fiber. Some of this noise is caused by random fluctuations. It is also possible that some of this noise is caused by an eavesdropper, Eve, attempting to steal the key.

Alice and Bob cannot distinguish these two sources of noise and must conservatively assume that all observed noise is caused by Eve. Alice and Bob run an error correction protocol to detect and correct these remaining errors. The two error correction protocols that are most widely used in QKD are Cascade and LDPC.

Cascade

Cascade is an error correction protocol that was popular in early key distribution research. In the Cascade protocol, the raw key is split up into blocks. Bob sends a parity bit for each block to Alice, and Alice checks the parity bit. If Alice finds that the parity bit is incorrect, she knows that there are an odd number of errors in the block.

In this case, Alice and Bob recursively split the block in half, to find the exact location of the error. If Alice finds that the parity bit is correct, it could be that there are no errors in the block.

However, it could also be that there are an even number of errors in the block. To handle this case, Alice and Bob perform multiple rounds of Cascade and shuffle the way raw keys are split into blocks in each round.

The Cascade protocol is efficient in the sense that it can correct errors while minimizing the amount of information that is leaked to Eve, but the Cascade protocol is very interactive: there are many back-and-forth classical messages between Alice and Bob, each taking a round-trip delay. This makes the Cascade protocol slow.

Low Density Parity Check (LDPC)

Low Density Parity Check or LDPC codes are more widely used for key distribution these days. LDPC codes are an error correction mechanism that is widely used in industry, including satellite communications, 5G mobile networks, and DVD disks. Here’s how LDPC works:

Alice computes a checksum over the raw key and sends it to Bob. The “PC” in LDPC refers to the fact that the checksum consists of parity check bits. The LD in LDPC refers to the fact that these parity bits are low density which means that they can be computed efficiently.

When Bob receives the checksum from Alice, not only can Bob detect whether there are any errors in the raw key, they can actually be corrected if the error rate is within certain limits. The algorithm that Bob uses to do the error correction is called belief propagation and uses a data structure that is known as a Tanner graph.

There are many flavors of LDPC codes, each optimized to deal with a particular error rate. Alice and Bob can dynamically choose which LDPC code to use, based on the estimated Quantum Bit Error Rate (QBER) as determined in the parameter estimation phase.

Information Reconciliation

During all of the steps described so far, some information about the key may have been leaked to eavesdropper Eve. In other words, Eve may have discovered some partial information about the key which could significantly speed up the brute force search for the correct key.

As mentioned before, a conservative assumption would be that all noise manifested in the observed Quantum Bit Error Rate is an indication of information leaked to Eve. On top of that, the parity bits exposed in the error correction protocol are another source of information leakage to Eve.

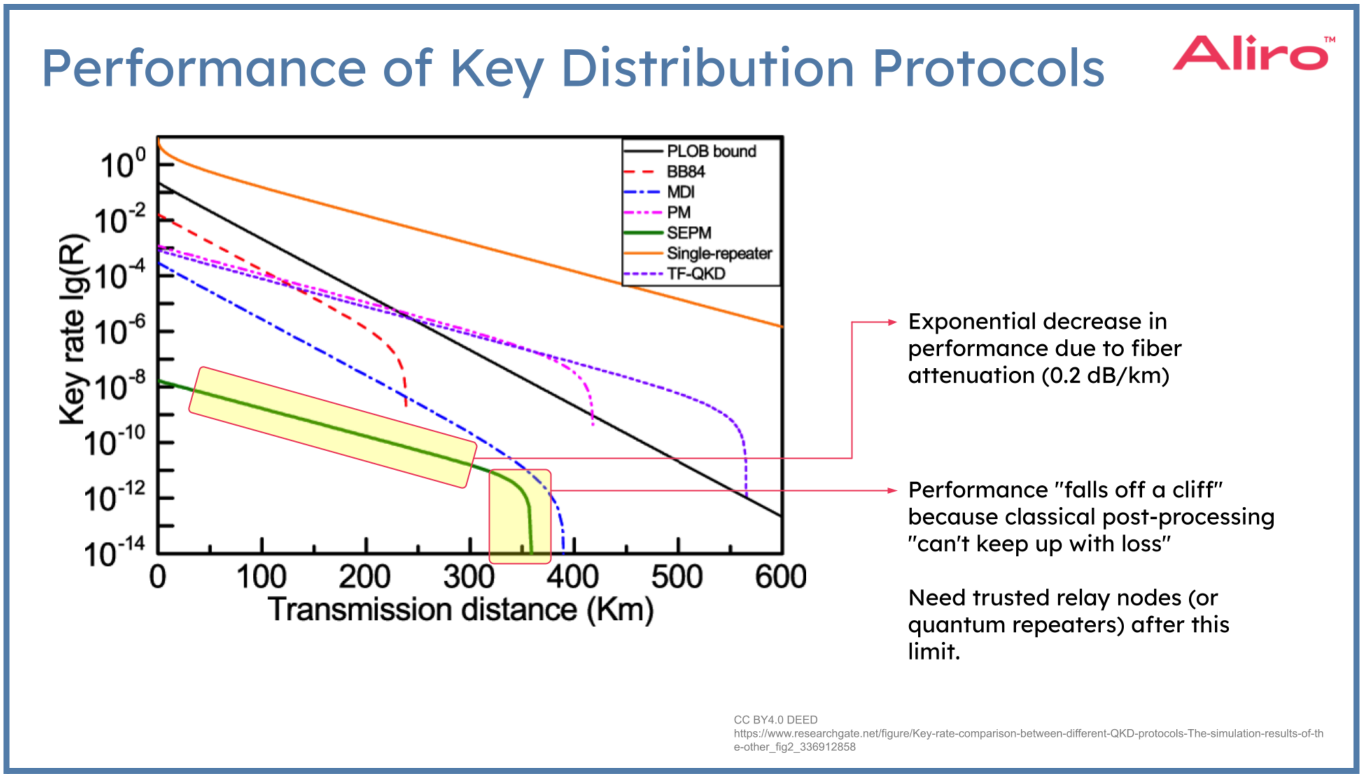
The goal of privacy amplification is to erase the information that was leaked by Alice.

This is achieved in two steps:

1. Using some rather complicated math, Alice and Bob can compute an upper bound on how many key bits have been leaked to Eve. This math is even more complex when taking finite-key effects into account.
2. Alice and Bob erase the information in the leaked key bits by essentially compressing the key. In practice this is achieved using a particular hash function called a Toeplitz matrix.

Impact on performance

Classical post-processing has a strong impact on the performance of key distribution protocols as shown in the graph below.[COMPARISON]



Performance of Key Distribution Protocols

Exponential decrease in performance due to fiber attenuation (0.2 dB/km)

Performance "falls off a cliff" because classical post-processing "can't keep up with loss"

Need trusted relay nodes (or quantum repeaters) after this limit.

CC BY4.0 DEED
https://www.researchgate.net/figure/Key-rate-comparison-between-different-QKD-protocols-The-simulation-results-of-the-other_fig2_336912858

On the Y-axis is the performance of the protocol, expressed in produced key bits per second. Note that this is a logarithmic scale. On the X-axis is the distance of the link in kilometers. This is directly and linearly related to the loss on the fiber in decibels.

The first thing to notice is that different protocols, represented by differently colored graphs, have different performances, but all the graphs have essentially the same shape. At first the graph is a straight line with a downward slope. This is due to the fact that the photon loss increases exponentially with the length of the fiber. Then, at the end of the graph, the performance drops off a cliff.

This creates a hard limit on the maximum distance of the link. This cliff is caused by the fact that the classical post-processing has to work harder and harder as the Quantum Bit Error Rate increases. Eventually, so many raw key bits have to be sacrificed for privacy amplification that nothing remains for the key.

Side Channel Attacks

In general, a side channel attack is a vulnerability in a security device which is due to some hardware imperfection or due to some behavior that is not considered in the security analysis. One should always consider the threat of side channel attacks for all types of security devices, both classical security devices and quantum security devices.

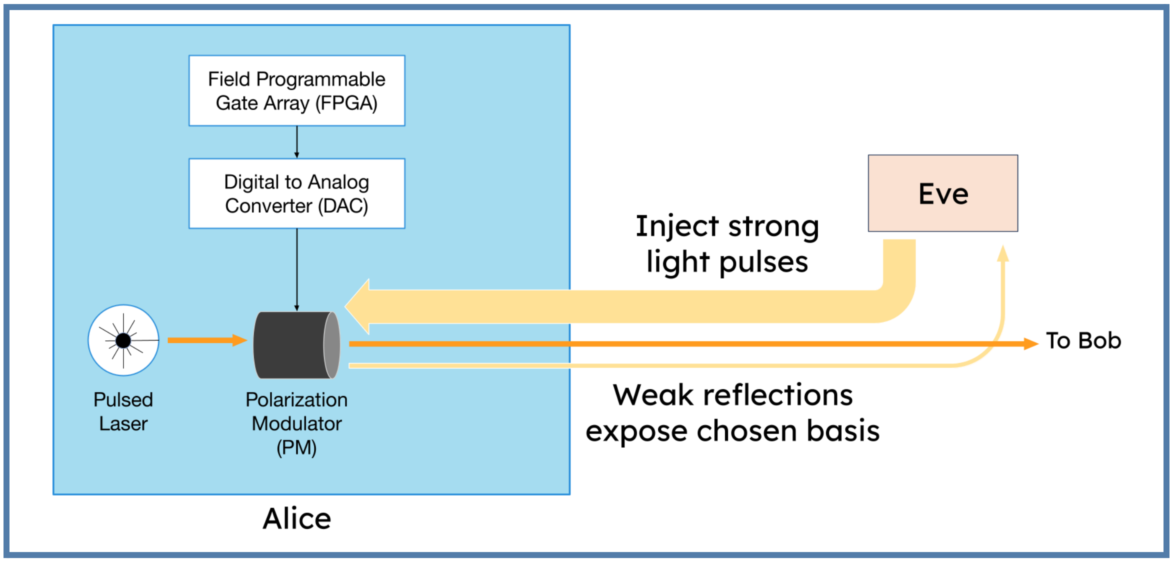
That said, there are also some side channel attacks that are unique to quantum devices. These attacks unique to quantum devices are listed in the two white papers referenced at the end of this white paper.[IMPLEMENTATION1, 2] Here we describe a few example side channel attacks to give you a flavor of what they look like.

TEMPEST

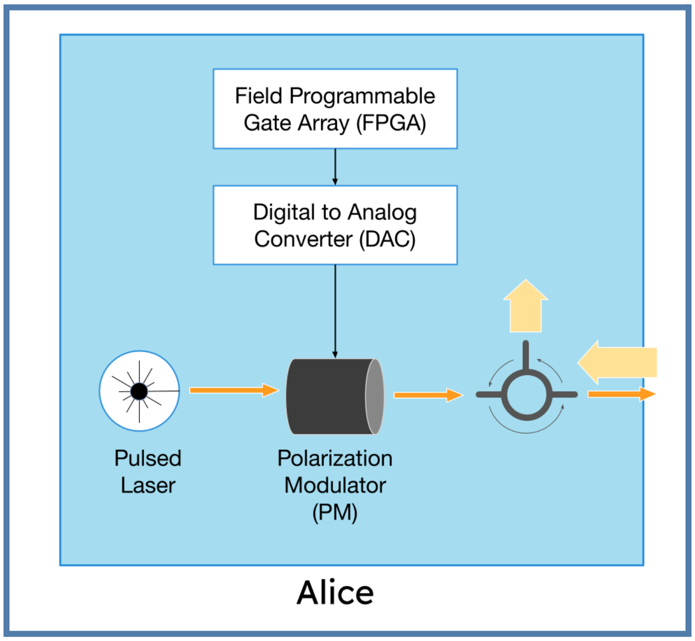
TEMPEST is the NATO code name for spying on information systems through leaking radio or electrical signals. This is an example of a side-channel attack that applies to both existing classical devices as well as new quantum devices. One can easily obtain a software defined radio gadget and the associated software to analyze radio signals leaking from a device. These leaking signals could give an attacker insight into what is happening inside the device.

Trojan Horse Attack

Another example of a side-channel attack against quantum systems is the so-called Trojan Horse attack. Here, attacker Eve, sends a strong laser light into Alice's quantum device. The state preparation hardware, in this case the polarization modulator, will reflect some small amount of this injected light. This allows Eve to determine the encoding basis of the key bits.


Field Programmable Gate Array (FPGA)
Digital to Analog Converter (DAC)
Pulsed Laser
Polarization Modulator (PM)
Alice
Eve
Inject strong light pulses
To Bob
Weak reflections expose chosen basis

The countermeasure against the Trojan Horse attack is to add an optical circulator. The optical circulator allows light to exit Alice's quantum device but blocks light from entering. There might still be a small amount of reflected light, but this can be handled by privacy amplification.



Field Programmable Gate Array (FPGA)
Digital to Analog Converter (DAC)
Pulsed Laser
Polarization Modulator (PM)
Alice

Weak Coherent Source Attacks

Many quantum systems use an attenuated photon diode laser, also known as a weak coherent source, as the single photon source. In such a source, the number of photons in a light pulse has a Poisson distribution. As a result, each light pulse doesn't always contain exactly one single photon; it may also contain zero photons or multiple photons.

If we make the pulse very bright, there is a high probability of having multiple photons. On the other hand, if we make the pulse very dim, there is a high probability of having zero photons. An example of an attack that takes advantage of this scenario is the Photon Number Splitting (PNS) Attack. Here, using a weak coherent source exposes the quantum system to a so-called Photon Number Slitting or PNS side-channel attack. When Alice sends a light pulse that contains more than one photon, Eve siphons off one of the photons and stores it in a quantum memory. Later, when Alice and Bob publicly announce the encoding basis, Eve measures the stored photon in the correct basis and discovers the key bit.

There are two possible counter measures against the photon number splitting attack. The simplest counter measure is for Alice to send very weak light pulses with a mean photon number well below one. However, this causes the key rate to be very low because most light pulses won't contain any photon at all.

A much better counter measure is to use decoy states. Here, Alice sends the light pulses in one of multiple (typically three) randomly chosen intensities: vacuum, signal, and decoy. By observing the statistics of the different Quantum Bit Error Rates of the different intensities, Bob can detect whether Eve is doing a Photon Number Splitting attack.

Conclusion

Quantum Secure Communication (QSC) represents the most robust and reliable way to protect and secure sensitive communications in an increasingly vulnerable digital landscape. By leveraging the principles of quantum mechanics, particularly entanglement, QSC offers unparalleled security against both classical and quantum-based attacks, while also offering flexibility. Entanglement-based quantum networks are the key enabler of this advanced level of protection, ensuring that any attempt to intercept or tamper with communications is immediately detectable. These networks are not only highly secure but also multipurpose and hardware-agnostic, making them adaptable to a wide range of applications and capable of integrating seamlessly with existing infrastructure while simultaneously supporting future quantum technologies

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 quantum networks working with telecommunications companies, national laboratories, intelligence organizations, and systems integrators.

Building 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 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^{SM} powered by Qubitekk in Chattanooga, Tennessee.

AliroNet™, the world's first full-stack entanglement-based network solution, consists of the software and services necessary to ensure that customers fully meet their quantum networking goals. Each component of AliroNet™ is built to be compatible with entanglement-based networks of any scale and architecture. AliroNet™ is used to simulate, design, and manage 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 Advanced Secure 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 quantum network designs, (2) Pilot Mode for implementing a small-scale quantum network testbed, and (3) Deployment Mode for scaling quantum networks and integrating end-to-end applications.

To get started on your quantum networks journey, reach out to the Aliro team for additional information on how AliroNet™ can enable secure communications and more for your organization.

info@alirotech.com

www.alirotech.com

References

[COMPARISON]. "Key Rate Comparison between Different QKD Protocols." ResearchGate, CC BY 4.0, https://www.researchgate.net/figure/Key-rate-comparison-between-different-QKD-protocols-The-simulation-results-of-the-other_fig2_336912858.

[EPPS]. "SPDC II 3D Model EN." Wikimedia Commons, CC BY-SA 4.0, https://commons.wikimedia.org/wiki/File.svg.

[IMPLEMENTATION1]. "Implementation Attacks on Quantum Key Distribution Systems." Federal Office for Information Security (BSI), https://www.bsi.bund.de/EN/Service-Navi/Publikationen/Studien/QKD-Systems/Implementation_Attacks_QKD_Systems_node.html.

[IMPLEMENTATION2]. Quantum Key Distribution: A Secure Implementation Framework. ETSI, https://www.etsi.org/images/files/ETSIWhitePapers/etsi_wp27_qkd_imp_sec_FINAL.pdf.

[MDI1]. Slater, Paul B. Recent Progress on the SIC Question. QCrypt 2014, https://qcrypt.github.io/2014.qcrypt.net/wp-content/uploads/Slater.pdf.

[MDI2]. "Measurement Device Independent Quantum Key Distribution." YouTube, uploaded by QCrypt 2014, 17 Oct. 2014, https://www.youtube.com/watch?v=WL7OPSO0s_s.

[POLARIZATION]. "Electromagnetic Wave." Wikipedia, Wikimedia Commons, CC BY-SA 4.0, https://en.wikipedia.org/wiki/Polarization_(waves)#/media/File.svg.