Quantum Cryptography

5

The Internet can be thought of as a channel of information being sent from you to everyone else connected to the Internet. If you wanted to transmit your sensitive information (such as bank account numbers or military secrets) over the Internet, then you have to ensure that only the persons you intend to read your information have access to your sensitive data. Otherwise, everyone would be able to read your information, e.g., access to your bank account details and transfer money out of your account. Therefore, one needs to encrypt any data sent over the Internet. Encryption, in this context, ensures that only the intended sender and receiver can understand any message being sent over an Internet channel.

5.1 Cryptography Fundamentals

Encryption relies on the sender and receiver sharing a secret key (that no one else has) and using that to encrypt and decrypt messages. In this way, since no one else has the secret key, no one else can understand the shared information. Because no one else understands the shared information, they cannot misuse it for their own benefit.

The only type of encryption protocol known to be perfectly secure is the One-Time Pad, also known as the Vernam Cipher.[1] It is assumed that two people exchange a shared key at least as long as the message in a completely secure way. The shared key encrypts the message to create the cipher, and the cipher is decoded by decrypting with the shared key. The protocol is best understood by trying it out with the associated worksheets in Sect. 10.7. In practice, due to not having a secure channel to share such a complicated key, despite being unbreakable, this method

[1] Shannon, Claude (1949). "Communication Theory of Secrecy Systems." *Bell System Technical Journal*. 28 (4): 656–715. https://doi.org/10.1002/j.1538-7305.1949.tb00928.x.

C. Hughes et al., *Quantum Computing for the Quantum Curious*,
https://doi.org/10.1007/978-3-030-61601-4_5

is usually not employed.[2] Here we see the fundamental caveat with encryption: you require a secure channel to share the secret key (if you do not have a secure channel then someone random can just take the secret key and encryption would be pointless), but if you have a secure channel then why do you need to encrypt your data? You need a way around this issue. How do you share a secret key in an insecure channel, where anyone can be listening?

5.2 Classical Cryptography

The way around sharing a secret key in an insecure channel in the majority of online communications is called public key cryptography.[3] A person called Alice makes two keys such that each key knows that only the other key is related to it (think of the keys as siblings). They are called the private and public key. Alice then gives the public key to everyone in the world but importantly keeps the private key for herself. Anybody else, say Bob, who wants to send a private message to Alice has to encrypt their message with the public key that Alice generated. There are many different types of encryption protocols that one can use. The special part of public key cryptography is that *only* Alice's private key can decrypt the message that was encrypted using its sibling public key. In this way, only Alice can read the message from Bob. As no one else has Alice's private key, no one else can read Bob's message. However, if Bob did not use Alice's public key but used a different public key to encrypt his message, then Alice cannot decrypt that message, as her private key is not a sibling key of the different public key. This whole cryptography scheme relies on the fact that no one can break the encryption protocol. If they could break it, then they could read Alice's message even if they did not have Alice's private key.

The most commonly used modern Internet encryption protocol is RSA encryption. RSA encryption relies on encrypting messages with keys that are made out of very large integers. To break the encryption protocol, an eavesdropper would need to factorize this very large integer into its (prime) factors. Factorizing a large integer into its (prime) factors is known to be a problem that classical computers cannot solve in any reasonable amount of time.[4] For example, given two large prime numbers p and q, it takes just a fraction of a second to multiply these two prime numbers together to produce a large integer $c = pq$. However, finding the two prime numbers p and q given just the integer c would take a classical supercomputer thousands of years.

RSA encryption works by encrypting the message with the public key. Decrypting the message by brute force requires factorizing a large integer in the public

[2] https://en.wikipedia.org/wiki/One-time_pad.

[3] https://en.wikipedia.org/wiki/Public-key_cryptography.

[4] https://en.wikipedia.org/wiki/Integer_factorization.

key, which would take thousands of years. However, the private key related to the public key knows how to check the prime factors of the public key and can decrypt the message easily. Because the encryption protocol is so difficult to break, no one would even attempt to do so. Instead, the eavesdropper may attempt to steal your private key by hacking into your computer, which Internet firewalls protect against. As such, nearly all Internet encryption relies on a computer not being able to factor large integers in a short amount of time.

However, in 1995, Peter Shor proposed a quantum computing algorithm, based on superposition and interference, that drastically speeds up the factoring process. A 4000-digit number, which would take a classical computer longer than the lifetime of the universe to factorize, would take less than a day on a large, stable quantum computer. Shor's Algorithm[5] can theoretically break modern encryption schemes, although quantum hardware is not sufficiently advanced yet to make this decryption practical. If it were, all your bank details, military secrets, and industrial secrets could be easily hacked. The details of Shor's algorithm are beyond our scope, so we will instead discuss how the same quantum computer could be used to ensure a key is shared over a secure channel.

Together, the one-time pad and **quantum key distribution** (QKD) would be a formidable combination. The BB84 QKD[6] simulation demonstrates how one could create a shared key using electrons and a Stern–Gerlach apparatus. The BB84 protocol is summarized below.

5.3 BB84 Quantum Key Distribution

5.3.1 Before Sending the Message

The sender (Alice) and receiver (Bob) publicly agree to the relationship between spins and bit value shown in Table 5.1.

Table 5.1 Table for the relationship between spin and bit values for quantum cryptography

Spin	↑	←	↓	→
Bit value	0	0	1	1

[5] https://quantum-computing.ibm.com/docs/guide/q-algos/shor-s-algorithm.

[6] https://www.st-andrews.ac.uk/physics/quvis/simulations_html5/sims/cryptography-bb84/Quantum_Cryptography.html.

5.3.2 Quantum Part

1. Alice randomly chooses either the x- or z-basis (horizontal or vertical Stern–Gerlach apparatus).
2. Alice sends an electron in superposition in the chosen basis through the SGA, measures the spin, and records the corresponding bit value as 0 or 1. The electron is sent to Bob.
3. Bob randomly chooses either the x- or z-basis.
4. Bob measures the spin of the electron and records whether it was 0 or 1.
5. Repeat steps 1–4 until desired level of security is achieved.

5.3.3 Example

Alice sends five electrons to Bob. When Alice sends an electron prepared in one basis and Bob measures in the same basis, they measure the same spin. However, if Bob measures in a different basis than Alice, then the electron will be in a superposition state and there will be a 50% probability of the state collapsing into 0 or 1. Example values for the first three bits of a BB84 experiment are shown in Fig. 5.1. Can you fill in the last two bits?

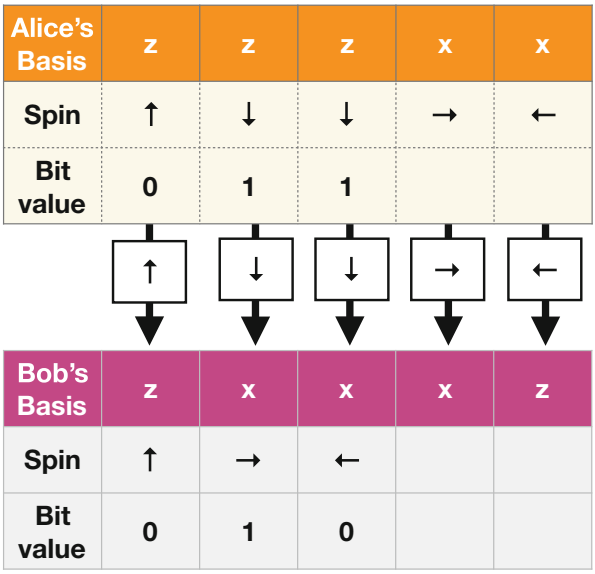

Fig. 5.1 Alice's and Bob's measurements of the BB84 protocol

Alice's Basis	z	z	z	x	x
Spin	↑	↓	↓	→	←
Bit value	0	1	1	1	0

Bob's Basis	z	x	x	x	z
Spin	↑	→	←	→	↓
Bit value	0	1	0	1	1

Key	0			1	

Fig. 5.2 Alice and Bob's measurements of the BB84 protocol completed from Fig. 5.1. The discarded bits are grayed out, and the key is 01

5.3.4 Classical Post-processing

1. Alice and Bob publicly share the basis used for each bit measurement *without revealing the actual bit value they measured.*
2. If they measured in the same basis, they keep that bit. If they measured in a different basis, they discard that bit. This is shown in Fig. 5.2. For the measurements performed in the same basis, Alice and Bob are guaranteed to have the same string of bits *unless there was an eavesdropper.*
3. They publicly compare a subset of the bits, say 20 out of 100 bits. If all 20 are the same, then it is unlikely that there was an eavesdropper. The remaining 80 become the shared key.

5.4 Detecting an Eavesdropper

If an eavesdropper (Eve) overhears the post-processing part where Alice and Bob share the basis used for each bit measurement, Eve has no information about whether any bit was either a 0 or 1. As Eve has no information, public post-processing sharing is not a dangerous action for Alice and Bob to take. The only

way for Eve to determine the spin value of the qubits, and as a consequence acquire important information, is to measure the qubit with her own Stern–Gerlach *before* it gets to Bob. This can be potentially dangerous for Alice and Bob. However, as the basis is not shared during the transmission, Eve must randomly pick a basis to measure the qubit intercepted from Alice. If Alice and Bob randomly choose to measure in a different basis, they throw away all the bits and it does not matter which basis Eve chooses. If Alice and Bob randomly choose to measure in the same basis then there are two outcomes depending on what Eve does: (1) If Eve randomly chooses the same basis as Alice, then she does not alter the state. This is bad, as Eve has successfully eavesdropped information without Alice and Bob knowing. (2) If Eve randomly chooses a different basis than Alice, then she alters the state and puts it into a superposition. Even though Bob is using the same basis as Alice, due to Eve altering the state, Alice and Bob can have a different spin measurement. This is how they can catch an eavesdropper.

5.4.1 Example

The eavesdropping situation is shown in Fig. 5.3. If Eve chooses the same basis as Alice, the spin is unchanged when it gets to Bob (bit #1). If Eve chooses a different basis than Alice, the spin could be different when it gets to Bob (bits #2 and #3). Eve could get lucky and Bob's bit could agree with Alice (bit #2). However, Bob is equally likely to measure something different from Alice (bit #3). Can you fill in what might happen with bits #4 and #5?

When Alice and Bob compare a portion of their key bits, a discrepancy would indicate the presence of an eavesdropper. If they compare a sufficient number of key bits and all of them match, they can be reasonably sure that the rest of it is secure. This statement will be quantified shortly in the questions.

5.5 Big Ideas

1. Classical RSA encryption assumes that factoring a large integer into its prime factors is prohibitively difficult. This assumption is true for classical computers, ensuring your information can be safe.
2. Shor's algorithm on a large and stable quantum computer could factor a large integer into prime factors, making classical encryption vulnerable.
3. New quantum encryption protocols are developed to keep information safe in the quantum era. The BB84 protocol is one way to share a secret key in a secure channel, that can then be used for encryption.

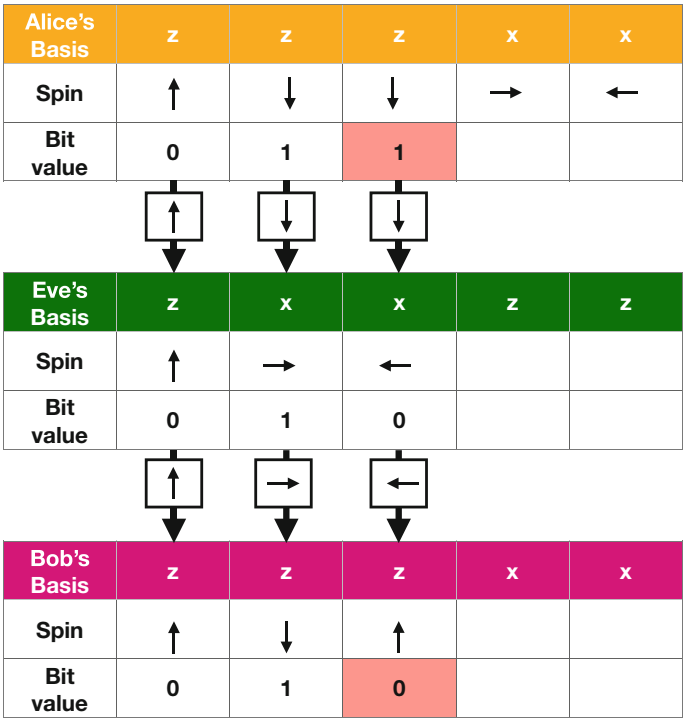

Fig. 5.3 An example of how to catch an eavesdropper using the BB84 protocol

5.6 Activities

- One-time Pad for Alice/Bob in Worksheet 10.7
- BB84 Quantum Key Distribution for Alice/Bob/Eve in Worksheet 10.8
- For those interested in hands-on experiments, see QuTools[7]

5.7 Check Your Understanding

1. If Alice and Bob exchange 1 million bits in order to use the BB84 quantum cryptography protocol, approximately how long will their bit-key string be? Assume they do not check for eavesdropping.
2. Alice and Bob share their lists of measurement basis, but do not share any more information about the bits. What is the probability that Eve will guess the correct bit for a single bit-key?

[7] https://www.qutools.com/quantenkoffer_science-kit/.

3. ■ Alice and Bob perform 20 bit-key measurements but do not share any information about the bits. What is the probability that Eve will guess the correct 20-bit key?
4. ● If Eve tries all possible key combinations with the one-time pad, can she crack the one-time pad?
5. ■ If Eve uses a Stern–Gerlach to measure the spin in between Alice and Bob's measurements, what percentage of the time will she be lucky and get the correct key-bit value without detection?
6. ■ If Alice and Bob measure in the same basis and compare 20 bits of their key, what is the probability that Eve could have eavesdropped all 20 bits without being detected?
7. ● Suppose that Eve discovers that the no-cloning theorem is wrong and finds a way to clone the state of each photon. How could she use a cloning machine to learn about the entire key without leaving any trace?