Comparison of Classical and Quantum Cryptography

Kathy Ngo

May 11, 2021

Abstract

Cryptography is a method of protecting sensitive information and providing secure communication. While classical cryptography relies on mathematical computation, quantum cryptography is based on the laws of quantum mechanics. In this paper, we will discuss the different cryptography algorithms as well as comparing the advantages and disadvantages of both schemes.

1 Introduction

Cryptography is the study of secure communication techniques that allows only the intended and authorized users to access the data transmitted. The purpose of cryptography is to ensure confidentiality, integrity, and authenticity. A cryptosystem consists of encryption algorithms that convert plaintext to ciphertext and decryption algorithms that convert ciphertext back to the original plaintext. The piece of information used for encryption and decryption is called a key. In classical cryptography, security is based on the secrecy of the key. However, as technology becomes faster and more efficient, it is no longer secure to only depend on the secret key. As a result, quantum cryptography was developed to overcome the shortcomings of classical cryptography and provide a better way to protect valuable information.

2 Background

2.1 Classical Cryptography

2.1.1 Symmetric Encryption - Private Key

The first method is private key cryptography which uses the same key for both encryption and decryption. Since a single key is used for both processes, it is also known as symmetric encryption. A real-world analogy to private key cryptography is a typical mechanical lock. Each member will have the same key and can use it to lock or open the lock. Because anyone who knows the encrypting key can easily decipher the message, the two communicating parties must employ a very reliable and secure channel to establish the key.

A symmetric cryptosystem is a five-tuple $(\mathcal{P}, \mathcal{C}, \mathcal{K}, \mathcal{E}, \mathcal{D})$ satisfying the following conditions:
$\mathcal{P}$ is a finite set of possible plaintexts.
$\mathcal{C}$ is a finite set of possible ciphertexts.
$\mathcal{K}$ is a finite set of possible keys.

For each $k \in \mathcal{K}$, there is an encryption rule $e_k \in \mathcal{E}$ and a corresponding decryption rule $d_k \in \mathcal{D}$, where $e_k : \mathcal{P} \longrightarrow \mathcal{C}$ and $d_k : \mathcal{C} \longrightarrow \mathcal{P}$ are functions such that for all $m \in \mathcal{P}$, we have $d_k(e_k(m)) = m$. [1]

There are two types of symmetric encryption. One method is block cipher which involves converting one block of plaintext at a time. Some examples of block ciphers include Data Encryption Standard (DES) and Advanced Encryption Standard (AES). The other method, stream cipher involves converting one byte of plaintext at a time. The most widely used stream cipher is RC4.

The benefits of private key cryptography is that it is simpler, faster, and more efficient since the same key is used for both encryption and decryption. However, if the key is intercepted by another person, that person can easily decrypt the message. At the same time, it can be very difficult to keep track of all the keys as the number of users increases.

2.1.2 Asymmetric Encryption - Public Key

In order to solve the problem of key distribution in symmetric encryption, asymmetric encryption method was developed. In asymmetric cryptography, each user has a public key and a private key. [2] The public key can be published, but the private key must be kept secret. Different from symmetric encryption, asymmetric encryption is a padlock that is locked by one key and can only be unlocked by another key. One of the most widely used public key cryptosystem is RSA developed by Ron Rivest, Adi Shamir, and Leonard Adleman.

The RSA cryptosystem relies on the fact that that it is extremely difficult to factor a large integer. We will take a look at a small example. In order to generate the public and private keys, Bob, the receiver, needs to do the following:

First, he chooses any two prime numbers, for example, $p = 3$ and $q = 5$ and computes $m = pq = 15$.
Then he computes $\phi(m) = (p-1)(q-1) = 2 \cdot 4 = 8$.
Bob can publish the number m, but he must keep the values of p and q private.
Next, he chooses a number $k = 7 < m$ such that $\gcd(k, \phi(m)) = 1$.
Then he computes $k^* = 7$ for which $kk^* \equiv 1 \pmod{\phi(m)}$.
Since only Bob knows $\phi(m)$, only Bob can compute k^* and keep it private.
The $(k, m) = (7, 15)$ will be the public key of Bob, and he can send it to Alice, the sender.

If Alice wants to send a plaintext, say $M = 8$ to Bob, with $M < m$, she needs to look up Bob's
public key and computes $C = M^k \pmod{m} = 8^7 \pmod{15} = 2$.
After that, Alice can send the ciphertext, $C = 2$, to Bob.
When Bob receives C from Alice, he can recover the original plaintext by computing
$M \equiv C^{k^*} (\text{mod } m) = 2^7 (\text{mod } 15) = 8$. [3]

In this example, we use very small numbers, however, in the real world, the typical RSA key sizes are 1024 bits or 2048 bits. Indeed, this method offer more security as there is no need for a private key exchange. Even if someone, say Mallory, has obtain the public key, she is unable to decipher the message without the private key. However, in this case, Mallory can replace Bob's public key with her own and send it to Alice. Alice, believing this public key comes from Bob, encrypts her message with Mallory's key and sends the enciphered message. Mallory can intercept again and decipher Alice's message using her private key. She can also re-encipher the message using the Bob's public key and send it back to Bob. When Bob receives the newly enciphered message, he will not know that it has been altered. Therefore, it is very important to obtain a public key certificate to verify the authenticity of the public key. At the same time, while it may seem impossible to factor a large number with a classical computer, Shor's algorithm shows that it takes only polynomial time on a quantum computer.

2.2 Quantum Cryptography

Although asymmetric encryption addresses the key distribution problem in symmetric encryption, both methods do not detect eavesdropping from a third party. As a result, quantum cryptography was developed to overcome the shortcomings of these schemes. BB84 is the first quantum cryptography protocol developed by Charles Bennett and Gilles Brassard.

First, Alice randomly chooses a sequence of photons with polarization and sends them to Bob.
For each photon Bob receives, he randomly chooses one of the two polarization bases
and measures the polarization of each photon Alice sends.
After that, Alice tells Bob via public channel which basis she used for each photon.

They only keep the data where they used the same polarization bases for measurement.
Then they openly evaluate and compute the error rate of a randomly selected subset of
their polarization data.
If the evaluation shows proof of eavesdropping, Alice and Bob abandon all their data and start over.
If the check passes, Alice and Bob can use this information to create some shared private keys. [4]

3 Conclusion

Cryptography plays an essential role in our everyday lives. We take advantage of cryptography every time we are sending an e-mail, purchasing an online product, or withdrawing cash from an ATM machine. With the advancement of new technologies, the need for faster and more secure encryption algorithms is constantly growing. In classical cryptography, there are many loopholes, and an eavesdropper can easily monitor the channel without being noticed. On the other hand, in quantum cryptography, we can ensure security and detect the presence of an eavesdropper. Quantum cryptography also has the potential to encrypt data for longer periods of time than classical cryptography. However, communication range of quantum cryptography is only 10 miles maximum, while communication range of classical cryptography is millions of miles. In addition, though quantum cryptography may sound perfect in theory, it is not known whether these security protocols still hold in real space under various types of attack. It is also not known whether we have the capability to build large scale quantum computers to implement such quantum key distribution systems.

References

[1] G. Bruss D.and Erdelyi et al. “Quantum cryptography: A survey.” In: (2007).

[2] Aakash Goyal, Sapna Aggarwal, and Aanchal Jain. *Quantum Cryptography & its Comparison with Classical Cryptography: A Review Paper.* 2011.

[3] Joseph H. Silverman. *A friendly introduction to number theory.* Pearson, 2012. ISBN: 9780321816191.

[4] I.V. Volovich and Ya.I. Volovich. “On Classical and Quantum Cryptography.” In: (2001).