Introduction to Quantum Cryptography

Rohma Khan

May 13, 2020

1 Introduction

The field of Cryptography has long ago created theoretical algorithms to be carried out on quantum computers. Quantum Computing will make most of our current cryptosystems useless thus leaving our online transactions easy to break into. This paper will delve deeper into what Quantum Cryptography is and the physics behind it. Then it will touch on how we will move forward.

Perhaps because the field was dominated by Cryptographers, Computer Scientists, and Mathematicians, much of Quantum Computing glosses over the *physics* behind the system. Below is a brief attempt to establish a link between Quantum Computing and Quantum Mechanics.

1.1 Determination of Hilbert Space

The first thing to do in any Quantum Mechanical problem is to establish the Hilbert Space, $\mathcal{H}$. One can thinks of Quantum Computing as a Quantum Mechanical problem where the solution is known but it is the System is unknown. The solution is a vector $|\Psi\rangle = a|0\rangle + b|1\rangle$, known as a qubit. It is the simplest bit that can be used for Quantum Computing. It is a superposition of two states that needs to have unitary transformations meaning the state must evolve with time. Thus a qubit is a Two-Level Quantum System. $\mathcal{H}$ is determined by the physical setup used to create the computer.

Here are some examples of qubits with their Hilbert Spaces.

1.1.1 Square Well

We can use a one dimensional square well potential to create a qubit. The Hilbert Space of this problem is well known to us to be $\mathcal{H} \to \mathcal{L}^2(\mathbb{R})$. A two-level system in this space can be written as

$$\Psi(t) = c_a(t)\Psi_a e^{-iE_at/h} + c_b(t)\Psi_b e^{-iE_bt/h} \tag{1}$$

The basis of this space is Ψ_n where $\Psi_n = \sqrt{\frac{2}{L}}\sin(\frac{n\pi}{L}x), n = 0, 1, 2, ...,$ If it were set up in a way where only the two lowest energy levels then the solution can be written as $\Psi(t) = c_a(t)sin(\frac{\pi}{L}x)e^{-iE_at/h} + c_b(t)\sin(\frac{2\pi}{L}x)e^{-iE_bt/h}$. We can write this abstractly as $|\Psi\rangle = \begin{pmatrix} a(x,t) \\ b(x,t) \end{pmatrix}$, which can represent a qubit. By causing perturbations in the hamiltonian, unitary transformations become possible because it causes changes in $c_a(t)$ and $c_b(t)$.

This is not really used but it is an example to show that the abstractness present in Quantum Mechanics translates to a certain level of abstraction in Quantum Computing. Fact harmonic oscillator quantum computers are realizable.

1.1.2 Spin

The spin of an electron can be used to represent individual qubits. The spin of an electron has two eigenstates, spin up or spin down, and the superposition of both causes the system to be an inherent two-state system. The system is written as $|\Psi\rangle = a\,|0\rangle + b\,|1\rangle$ where

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \tag{2}$$

Using hermitian operators as gates allow us to make measurements to the system. The Hilbert Space for an electron's spin is $\mathcal{H} \rightarrow \{\mathcal{X} = \begin{pmatrix} a \\ b \end{pmatrix}; a, b\epsilon(\mathbb{C})\}$.

The generic notation for qubit states are $|0\rangle$ and $|1\rangle$ but that does not necessarily mean spin is what is being represented. The polarization of photons, or the ground and excited states of electrons can equally be described as qubits. Just like current computing is done using binary bits Quantum Computation occurs using qubits. Now that we have lightly explained some of the physical background, let's discuss potential uses of Quantum Computation.

2 Cryptographic Applications

Modern Cryptography uses of mathematical processes to encrypt messages for security and communication. The goal of public key cryptography is to exchange information securely over channels that are not secure. There are two 'keys' in public key cryptography, an encryption key and a decryption key. The encryption key is public but the decryption key is private, and thus makes the a public encrypted message secure. Different cryptosystems have been developed with differing levels of security. One such system developed first by the NSA and then independently by public inventors is RSA.

Quantum Computation gives rise to methods that can easily find the decryption key. Because Quantum Computers are much more efficient in certain areas, previously secure cryptosystems are no longer so. Quantum Computation can speed up calculations of searching, solving the discrete log problem, solutions of NP-complete problems, order finding and more. Below is an example of how Quantum Computing can effect current cryptosystems.

2.1 RSA and Shors Algorithm

RSA is a public key cryptosystem that allows two entities to communicated securely, even with an eavesdropper listening in on their channel. Here is how two people, Alice and Bob would exchange a message.

1. Alice sends a public key (N, e) to Bob, where N and e are chosen. N is picked so it is the composite of two prime numbers p, q and larger than any potential messages. p and q are secret. e is chosen so that $\gcd((p-1)*(q-1), e) = 1$. This is important because RSA is done using modular arithmetic.

2. Bob uses the key to encrypt his message. He does the calculation $c \equiv m^e \pmod{N}$ and sends c to Alice publicly.

3. To decrypt Bob's message Alice finds d her decryption key. $d \equiv e^{-1} \bmod ((p-q)*(q-1))$. She then decrypts the message by carrying out the operation $m \equiv c^d \bmod (N)$

The security of RSA is based on the difficulty of factoring N and finding p & q. There are a few techniques and ways to reduce the risk of someone finding the composition of N. Here is an example of RSA, to help better understand it. Alice sends Bob a public key of $(77, 17)$. Bob want to send the letter "F" to Alice. The message, m, is encoded using ASCII so that $m = 70$, and then encrypted so that $c \equiv 70^{17} \bmod (77)$. He sends Alice $c = 49$. $N, e, \&c$ are all public. In order to decrypt his message Alice takes her secret keys $p = 11\&q = 7$ to find her decryption exponent $d \equiv 17^{-1} \bmod (60)$, so $d = 53$. $p, q, \&d$ are all private. She then takes $m \equiv c^d \bmod (77)$ to find that $m \equiv 49^{53} \bmod (77) = 70$. In 1994, a quantum algorithm was developed to break this cryptosystem called Shor's Algorithm. To use Shor's Algorithm to factor N first chose a random $x \bmod (N)$, find its order r such that $x^r \equiv 1 \bmod (N)$, and find the gcd $(x^{\frac{r}{2}} - 1, N)$. This will be one factor of N and the other will be the gcd $(x^{\frac{r}{2}} + 1, N)$. This is because $(x^{\frac{r}{2}} - 1, N) * (x^{\frac{r}{2}} - 1, N) = x^r - 1 = 0$ mod (N). If r is odd or $x^{r/2} \equiv -1 \bmod (N)$ pick a new x. Finding the order r requires the use of the Quantum Fourier Transform.

2.2 Quantum Fourier Transform

The quantum Fourier transform can be used to find the order of a function. Given an x and and N, it can be used to find $x^r \equiv 1$ mod (N). Below is technical explanation of the process. The QFT is a unitary transformation that does the following

$$|a\rangle \to \frac{1}{\sqrt{q}} \sum_{c=0}^{q-1} e^{2\pi i a c/q} |c\rangle \tag{3}$$

for some a such that $0 \leq a \leq q$. Shor's original algorithm uses the QFT to find r by doing the following. Find $q = 2^h$ where $N^2 \leq q \leq 2N^2$. Next create a machine with with a q-qubit representing numbers $a \pmod{q}$. Our machine is represented by $\frac{1}{\sqrt{q}}\sum_{a=0}^{q-1}|a\rangle$. Then we calculate and apply $|x^a mod(n)\rangle$ giving us $\frac{1}{\sqrt{q}}\sum_{a=0}^{q-1}|a\rangle|x^a(modn)\rangle$. We then use the QFT on the first qubit. This maps $|a\rangle \rightarrow \frac{1}{\sqrt{q}}\sum_{c=0}^{q-1}e^{2\pi iac/q}|c\rangle$. The machine is then in the state

$$\frac{1}{q}\sum_{a=0}^{q-1}\sum_{c=0}^{q-1}e^{2\pi iac/q}|c\rangle|x^a mod(N)\rangle \tag{4}$$

. Now measure the first qubit and observe the value of $|c, x^a mod(N)\rangle$. Our machine is in the state $\frac{1}{q}\sum_{a=0}^{q-1}e^{2\pi iac/q}|c\rangle|x^a mod(N)\rangle$.The probability the machine is in a particular state $|c, x^k mod(N)\rangle$ is the square of its coefficient $\|\frac{1}{q}\sum_{a:x^a\equiv x^k}^{q-1}e^{2\pi iac/q}\|$. The order of x is r so a can be rewritten as $a = br + k$ and our probability becomes $\|\frac{1}{q}\sum_{b=0}^{\lfloor(q-k-1)/r\rfloor}e^{2\pi i(br+k)c/q}\|$. Because k is not in the summation, it can be factored out and then disregarded. rc should be replaced by $\{rc\}_q$, which is the residue of equivalent rc mod(q) values. Then turning the sum into an integral, and substituting $u = \frac{rb}{q}$ the summation becomes

$$\frac{1}{r}\int e^{2\pi i\frac{\{rc\}_q}{r}u}du \tag{5}$$

. With some manipulation of boundaries to get the highest probability there is an ideal range of values for $\{rc\}_q$ found to be $\frac{-r}{2} \leq \{rc\}_q \leq \frac{r}{2}$. Replacing the residue of rc mod (q), $\{rc\}_q$ by $rc - dq$ the inequality is $\frac{-r}{2} \leq rc - dq \leq \frac{r}{2}$. The order r can then be found using the term

$$|\frac{c}{q} - \frac{d}{r}| \leq \frac{1}{2q} \tag{6}$$

because c and q are known there will be a solution to the inequality. By keeping the fraction $\frac{d}{r}$ in lowest terms, the order r can be found. This is a complicated explanation of how to find the order of a number mod (q). However this has been coded in qiskit and is available online, and is theoretically very fast on a quantum computer. All of this to say is that RSA and other private key cryptosystems are in trouble.

3 Conclusion

This paper began with an explanation on what a qubit is in the physical sense. After a brief description, it went on to explain one use of a quantum computer and a specific quantum algorithm. There are many uses for the quantum Fourier transform within the field of cryptography itself. Quantum Computing will completely change online security.

References

[Used in Introduction] DiVincenzo, David P. “The Physical Implementation of Quantum Computation.” Fortschritte Der Physik, vol. 48, no. 9-11, 2000, pp. 771–783.

[Used in Section 1] Griffiths, David J., and Schroeter, Darrell F. Introduction to Quantum Mechanics. Third ed., 2018. chpts 4.4 & 11.1

[Used in Section 1 and 2] Nielsen, Michael A, et al. Quantum Computation and Quantum Information. Cambridge University Press, 2000. chpts2 2 & 7

[Used in Section 2] Shor, Peter W. “Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer.” SIAM Journal on Computing, vol. 26, no. 5, 1997, pp. 1495–1500.

[Used in Section 2] Silverman, J.H, et al. An Introduction to Mathematical Cryptography. Springer New York, 2008.