Quantum Attack On RSA Cipher System

Huajun Zhang student of Kris Gaj

Abstract— Large number of current cryptography systems based on public key method that depends on the high time cost and impractical of factoring large numbers into primes. RSA is one of the most widely used public key cipher system. According to my research, breaking RSA is not an impractical problem, if we can implement Shor's Algorithm in a Quantum computer. The power of Shor's Algorithm gives us an efficient way to factor large number in polynomial time. It's a good example to show Quantum computer is high threat to current Cryptosystems. Although no reliable high Qubits Quantum computer has been available yet, we still need awareness the threat to current cryptography. Since materials and low -temperature physics researches have breakthrough many bottlenecks in recent years, a large commercial Quantum computer will soon be launched in few years. I suggest that we need further research on how to design post quantum public cipher against quantum attack.

I. Introduction

WE all believe that technology continues to change our world past 60 years. It changes the way of how we create new objects, how we work, and how we communicate with each other. Also, the capability of solving complicated mathematical problems is helping us understanding of our world as well as the universe beyond. If we say computer technology is a scientific evolution, the Quantum computers will be next massively revolution.

In general, a Quantum computer can perform any task that classical computer can do. However, It doesn't mean quantum computers guarantee faster than classical computers. In solving some specific problem, Quantum computer is exponentially faster than a classical computer. It's very useful to breakthrough current science researches in Cryptography, Artificial Intelligence, Financial Modeling, Molecular Modeling, Financial Modeling, Weather Forecasting and Particle Physics. I want to investigate how quantum computer affects current Cryptography cipher systems.

Public and private cipher systems are two most well known crypto-systems. In private cipher system such as DES, we can preform a quantum attack using Grover's Algorithm. Grover's Algorithm allows us to process the same tasks with different values simultaneously. The number of parallel tasks depends on the number of Qubit register. With sufficient number of Qubits, we can break a complicate private key in linear time. However, it's easy to defense these attack by adding the key size. On the other hand, public cipher is much hard to against Quantum attack. "Quantum computers can break public-key cryptography that is based on assuming hardness of factoring, discrete logs, and a few other problems." [1] Instead of investigating private cipher systems, my project focuses on analyze why it's hard for RSA to defend Quantum attack.

II. Quantum Computing

Quantum computer are based on some essential features of quantum mechanical such as superposition and entanglement. The process performed on a quantum computers call Quantum computing. The essential features make quantum computer don't like classical computer that based on binary transistor. The classical computing encodes data into binary digits - bit. Each bit can hold either one or zero state. Quantum computers use Quantum binary digit - Qubit as basic unit to store and process data. Similar to classical bit, Qubit can be encoded as zero and one. However, Qubit can hold multi state that means a Qubit can represent both 1 and 0 at the same time. The feature calls superposition. Quantum computer is an ideally machine to perform high volume computing simultaneously because of the power of quantum superposition.

III. Breaking RSA

A. Overview

RSA is one of the first and most well known public-key cryptosystems, which developed by Ron Rivest, Adi Shamir, and Leonard Adleman. Till now, RSA is still very reliable secure system for data transmission. The system based on one way trapdoor functions. It uses two sets of key one for encryption and another for decryption. Encryption key is available for target public and decryption key is only visible by key creator. We use the encryption key to encrypt message. also, decryption function use decryption key as trap door to solve. The time cost for encryption and decryption are practical. However, It's impractical hard to break a RSA cipher without decryption key.

B. Key Generation

The public key (encryption) contains 2 values N and e. N is a product of two large prime numbers p and q. We solve Eulers totient function Phi which $\lambda(N)=(p-1)(q-1)$. e is a value that we random pick number co-prime and smaller than $\lambda(N)$. The private key (decryption) also contains 2 values N and d which $d \equiv e-1 \pmod{\lambda(n)}$. After generating the 2 keys, we need to destroy values of p, q immediately. Also, the value d must be kept in secret. The public key is distributed in public.

C. RSA Breaking

"Anyone can use the public key to encrypt a message, but

with currently published methods, and if the public key is large enough, only someone with knowledge of the prime numbers can decode the message feasibly." [3] So, the hardness of breaking RSA encryption is factoring a large number. Factoring large number is a NP-hard problem. It takes nondeterministic polynomial time to solve. "This asymmetry is based on the practical difficulty of the factorization of the product of two large prime numbers, the "factoring problem"." [2, slide 2.] As an attacker it easy to generate public key and cipher text. However, factoring problem remains an open question. The best-applied factoring algorithm GNFS takes exponential time to factor a large number. As a result, it is still not efficient enough to threat current RSA system.

IV. Shor's Algorithm

A. Overview

Shor's algorithm" is a quantum algorithm that was introduced in 1994 by Peter Shor. With sufficient Qubits operating on a Quantum Computer, Shor's algorithm can theoretically solve factorization problems much more efficient than any classical computers even most powerful super computer. As a result, it will be highly risk for current ciphers system (cryptography system).

The core idea of Shor's Algorithm is to reduce Factoring problem to Period-finding problem. If we can solve Period-finding problem in polynomial time, we can solve Factoring problem in polynomial time. However, the period finding step in Shor's Algorithm base on Quantum Fourier Transformation that must to be performed on a Quantum computer.

We recalled that breaking RSA could be recognized as a large number factoring problem. The prime task of Shor's Algorithm is finding a non-trivial factor for large number N. In RSA, key N is product of 2 large prime p and q. Shor's Algorithm will help us to find the exactly values of p and q. Then we can use p and q to recover the key generation process to compute the private key (N, d).

B. Five Step of Shor's Algorithm

Assume that we want to factor a large integer N. According Peter's "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer", I rewrite Shor's Algorithm as a five steps simple pseudo code in Figure 1.

Figure 1.

Step 1: random pick m from (0,N)
If **gcd(N,m) = 1** continue to step 2,
Else **gcd(N,m)** is a non-trivial factor of N, we done

Step 2: Quantum period finding r of
f(x) = m ^ x mod N
m^1 mod N, m^2 mod N, m^3 mod N...

Step 3: If P is **even** continue to Step 4
Else go back to step 1

Step 4: If $\boldsymbol{m^{r/2} \neq 0 \; mod \; N}$ continue to step 5
Else go back to step 1

Step 5: we compute $\mathbf{d = gcd\left(m^{\frac{r}{2}} - 1, N\right)}$
d is a non-trivial prime factor is of **N**

The Step 2 in Figure 1 is the only Quantum step. In Step 2, we perform quantum period finding that based on Quantum Fourier Transformation. Step 3 and Step 4 is simply condition check base on Euclid Algorithm. Since Step 3, 4 and 5 involved modular arithmetic we can simple apply it by using Euler's Algorithm. Time complexity of Euler's Algorithm is around O(log n).

C. Period in Shor's Algorithm

Briefly, we call the sequence of number is periodic if a small sequence of numbers repeats in the large sequence of numbers again and again. In mathematics, a periodic function is a function that repeats its values in regular intervals or periods. [4]

In Shor's Algorithm we are looking for the period of function of f(x) = m ^ x mod N. Let's perform a general period finding below.

Example 1.

Let m=2, N=15, x=1,2,3,4,5 …
m^x = 2^1, 2^2, 2^3, 2^4, 2^5, 2^6, 2^7, 2^8, 2^9, 2^{10} …
= 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024 ...
f(x) = m^x mod N
= 2, 4, 8, 1, 2, 4, 8, 1, 2, 4, ...

In Example 1, we can see that number 2,4,8,1 periodic appear and f(x) = f(x+4). So, the period of Example1 is 4.

V. Quantum Period Finding

A. Overview

If we have a periodic function f(x) = f(x+r), r is the period of function f(x). Classical computer don't have any algorithm can efficiently find period of a function. We need repeat computing function f(x) with different values then finding the period through the values of f(x). It computes f(x) at least x times. "A Quantum Computer can do very much better than this It can compute f(x) for all values of x in a single parallel computation."[7, slide 4.] It's a great advantage to find period in a quantum computer.

B. Quantum Steps

The idea of the quantum period finding is reference from Dr Iain Styles's Lecture slide "Shor's Factorization Algorithm". In order to factor a large number N, we need two 2^n Qubits register to hold the value N. The quantum period finding has 3 major steps construct and initialize superposition state, evaluating f(x) in parallel, making the measurement.

The detail process follows Figure 2.

Figure2. [7, slide 5-7.]

1. Initialize 2 register in state $|0\rangle|0\rangle$.
2. Construct a composite system that has state $|\psi\rangle = |0\rangle|0\rangle$.
3. Apply the QFT to the first register. Recall that the QFT is:
$$y_k = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} x_j e^{2\pi i j k/N}$$
Then the state after QFT is:
Where $w = 2^n$. $|\psi\rangle = \frac{1}{\sqrt{\omega}} \sum_{j=0}^{\omega-1} |x\rangle\,|0\rangle$
4. Put f(x) value in second register $|x\rangle|f(x)\rangle$
Then the total state:
$$|\psi\rangle = \frac{1}{\sqrt{\omega}} \sum_{x=0}^{\omega-1} |x\rangle\,|f(x)\rangle$$
5. Then we group the states by the value f(x)=u
$$|\psi\rangle = \frac{1}{\sqrt{m}} \sum_{k=0}^{m-1} |x_0 + kr\rangle\,|u\rangle$$
6. Perform QFT and find the spectrum
$$\begin{aligned} |\psi\rangle &= \frac{1}{\sqrt{m}} \sum_{k=0}^{m-1} \frac{1}{2^{n/2}} \sum_{y=0}^{2^n-1} e^{2\pi i (x_0+kr) y/2^n} |y\rangle \\ &= \frac{1}{\sqrt{2^n m}} \sum_{y=0}^{2^n-1} e^{2\pi i x_0 y/2^n} \sum_{k=0}^{m-1} e^{2\pi i k r y/2^n} |y\rangle \end{aligned}$$
7. The probability distribution graph of state x should follow:
$$p(y) = \frac{1}{\sqrt{2^n m}} \left| \sum_{k=0}^{m-1} e^{2\pi i k r y/2^n} \right|^2$$
The Maxima should appear at $y \approx j2^n/r$
Where j is an integer
Then we can say Measurement likely to give $j2^n/r$

Most of time Shor's Algorithm works. However, there are two situations we need to notify. Firstly, when you measure the first register at the end, there is a finite probability of getting a bad result. [7, slide 14] Secondly, j and r can need to be co-prime. In order to avoid these situations we can make measurement with all f(x) value then we can calculate the average valid measurement.

VI. Time Analysis

A. Classical GNFS

In classical computing, the most efficient factoring algorithm is General Naive Field Sieve algorithm. If we want to factor a large number N, the time complexity of GNFS is $\boldsymbol{O(exp\ (lg\ N\ )^{1/3}(lg\ lg\ N\ )^{2/3})}$ in a exponential time. [6, page2]

B. Shor's Algorithm

Briefly, Shor algorithm can factor large number in polynomial time. Shor's Algorithm step 1 we Euclid algorithm to find greatest common divisor that take polynomial time. We may repeat step 1 $phi(N)$ times in the worse case. The Quantum period finding running time is roughly $O(lg\ N)$. However, we may need repeat measurement $O(lg\ lg\ N)$ times. Also, modular arithmetic and GCD operations in step 3,4,5 can be done in polynomial time. As a result, the time complexity of Shor's Algorithm is $\boldsymbol{O((lg\ \ N)^{\ 2}(lg\ lg\ N)\ (lg\ lg\ lg\ N))}$.

VII. Conclusions

In the mission of factoring a large number, Shor's Algorithm runs significantly faster GNFS. Instead of searching a large interval of the potential prime factors, Shor's Algorithm just search in a small interval the potential periods. After Quantum period finding, the step 3 is check if the period is r. The probability that P is odd, and you have to return to step one is $\boldsymbol{(1/2)^k}$ where k is the number of distinct prime factors of N. [8]

So even if we increase the size of key, there will nearly no affect to the performance of Shor's Algorithm. RSA will be very hard to keep security in the era of Quantum Computer. "There is a 1 in 7 chance that some fundamental public-key crypto will be broken by quantum by 2026, and a 1 in 2 chance of the same by 2031." Dr. Michele Mosca Deputy Director of the Institute for Quantum Computing, University of Waterloo April 2015. [9, slide 49] Not only RSA, most of public crypto-systems could be victims of Quantum Attack.

In Order to prepare the upcoming Quantum Attack, researchers should investing a lot time and resources on Post-Quantum Cryptography research in these years. The basic idea of PQC is "Capable of being implemented using any traditional methods, including software and hardware" and Running efficiently on any modern computing platforms: PCs, tablets, smartphones, servers with FPGA accelerators, etc".[9, slide 50]

REFERENCES

[1] R. Wolf, Quantum Algorithms. pqcrypto.org, 2017.

[2] S. Nigel, “Dr Clifford Cocks CB,” *Bristol University*, Aug. 2011.

[3] Rivest, R. Shamir, A. Adleman, L. "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems”. 120–126. 2008

[4] “Periodic function,” *Wikipedia*, 13-Nov-2018. [Online]. Available: https://en.wikipedia.org/wiki/Periodic_function. [Accessed: 10-Dec-2018].

[5] Peter Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. SIAM journal on computing, 26(5):1484–1509, 1997.

[6] Lomonaco and S. J., “Shor's Quantum Factoring Algorithm,” *[astro-ph/0005112] A Determination of the Hubble Constant from Cepheid Distances and a Model of the Local Peculiar Velocity Field*, 09-Oct-2000. [Online]. Available: https://arxiv.org/abs/quant-ph/0010034v1.

[7] I. Styles, "Lecture 7: Shor’s Factorisation Algorithm", *Cs.bham.ac.uk*, slides 4-10.2018. [Online]. http://www.cs.bham.ac.uk/internal/courses/intro-mqc/current/lecture07_handout.pdf. [Accessed: 10- Dec- 2018].

[8] A. Marchenkova, “Break RSA encryption with this one weird trick – Quantum Bits – Medium,” medium.com, 13-Aug-2015. [Online]. Available: https://medium.com/quantum-bits/break-rsa-encryption-with-this-one-weird-trick-d955e3394870. [Accessed: 10-Dec-2018].

[9] K. Gaj, “Public Key Cryptography: Algorithms, Key Sizes, & Standards,” 30-Nov-2018.