Improving the Success Rate of Quantum Algorithm Attacking RSA Encryption System

Yumin Dong[1*], Hengrui Liu[1†], Yanying Fu[1†], Xuanxuan Che[1†], Cheng Liu[1†], Lina Sun[1†] and Guangrui Wen[1†]

[1*]College of Computer and Information Science, Chongqing Normal University, Chongqing, 401331, China.

*Corresponding author(s). E-mail(s): dym@cqnu.edu.cn;
Contributing authors: 2021210516054@stu.cqnu.edu.cn; 2021210516033@stu.cqnu.edu.cn; 2021210516023@stu.cqnu.edu.cn; 2021110516016@stu.cqnu.edu.cn; 2021110516020@stu.cqnu.edu.cn; 2021210516014@stu.cqnu.edu.cn;
†These authors contributed equally to this work.

Abstract

The task of Shor's factorization algorithm (SFA) is to find a non-trivial factor of a given composite number $\boldsymbol{N}$. In the algorithm, it is explicitly required that the constructor $\boldsymbol{f(x) = a^x \bmod N}$ cycle is an even number, otherwise it returns to the algorithm header to recalculate. According to the analysis of the principle of SFA, the characteristics of RSA public key cryptosystem and a large number of calculation results, the random selection of a value is related to the obtained period. Aiming at this defect, a new optimization scheme is proposed on the basis of the previous optimization, and the odd cycle obtained when a is a perfect square number randomly becomes effective. When the arbitrarily chosen value of a obtains a period that is a multiple of 3, by modifying the decomposition method, the requirements for the period are relaxed without affecting the algorithm complexity. Using the classic algorithm, we experimentally prove the effectiveness of the improved algorithm, and the new algorithm significantly reduces the probability of repeated operations.

Keywords: Quantum Computation, Shor's factoring algorithm, Quantum algorithms, RSA

1 Introduction

RSA public key encryption system is the most widely used type of public key system[1]. RSA is named after the initials of the inventors Rivest, Shamir and Adelman. The principle is to use the product of two large prime numbers to be difficult to decompose to encrypt the plaintext. Mathematical research has shown that for any computer based on classical physics, the time required to factorize N increases exponentially with the number of digits L of N[2]. This situation is not inherently possible within the scope of classical computers solve. To the surprise of mathematicians, physicists and computer scientists, in April 1994, Shor[3] of Bell Labs in the United States proposed the factorization quantum algorithm SFA, which destroyed the security of RSA cryptography. In order to decompose the large number, the SFA uses two registers, one of which stores the number and the other stores the processed number. Randomly select a number a smaller than N and $gcd(a, N) = 1$, and use quantum Fourier transform to calculate the period of $f(x) = a^x \mod N$.

From Euler's Theorem in Number Theory $a^r \equiv 1 \mod N$, We have $\left(a^{r/2} - 1\right)\left(a^{r/2} + 1\right) \equiv 0 \mod y$.

As long as the period is even, $\left(a^{r/2} - 1\right)$ and $\left(a^{r/2} + 1\right)$ must be factors of or multiples of factors.

In the previous SFA, we often could not get the appropriate cycle directly, so we had to start from scratch. In order to reduce the waste of expensive quantum computer computing cost, many scholars have made some changes to SFA, which improves the efficiency of SFA attacking RSA on the original basis.

A dissertation[4] propose an efficient and exact quantum algorithm for finding the square-free part of a large integer a problem for which no efficient classical algorithm exists. The algorithm relies on properties of Gauss sums and uses the quantum Fourier transform.

If an inappropriate a is chosen, it will cause the SFA to be calculated from scratch. A common strategy is to reduce the probability of finding the useless cycle r, which has a great effect on improving the operating efficiency of the SFA. So far, research in this area has mainly focused on reducing the occurrence of odd cycles, As in[5], it shows that odd cycles can be avoided by choosing a non-square coprime under the modulo algorithm. This method does improve the probability of SFA success, but when N is large enough, the possibility of randomly picking a as a perfect square will decrease sharply. But it is worth mentioning that when a is squared, The odd cycle of $a^x \mod N$ is a very probabilistic event, which we will prove by experiments later.

In the case of obtaining even-numbered cycles, the SFA may still not obtain the desired decomposition result. The reason is that when $a^r \neq -1(\mod N)$ *or* $a^r \neq 1(\mod N)$ the factors of large numbers N cannot be obtained by calculating $\gcd\left(a^{r/2} + 1, N\right)$ *or* $\gcd\left(a^{r/2} - 1, N\right)$.

The literature[6] proposed a method that can still get the results when this situation is encountered. This finding suggests that the above constraints are

only sufficient, but not necessary, conditions for the successful decomposition of SFA.

In addition to solving the problem of integer factorization, literature[7] proposes a quantum algorithm to attack RSA based on equation solving from the point of view of non-factorization and according to the characteristics of RSA public key cryptosystem. It has polynomial time complexity, and the success rate is higher than that of SFA attacking RSA, and it does not have to meet the limit of period r.

Smolin[8] proposed a new idea of quantum integer decomposition, and realized the decomposition of a large number N by finding a random number a of order 2.Since the order of the elements is 2, only 2 qubits are needed for the second register, thus greatly reducing the number of qubits. Geller[9] used the characteristics of Fermat number to realize the decomposition of 51 and 85 with 8 qubits, and gave the circuit diagram of quantum realization. Dattani[10] proposed an improved algorithm of SFA, which uses multiple quantum registers to achieve integer decomposition. This method requires more qubits. Cao[11] proposed an improved algorithm of SFA, using multiple quantum registers to achieve integer factorization.

Through further research and analysis on the scheme of cracking the RSA public key encryption system, it is found that most of the breakthroughs in the optimization of the predecessors are concentrated in the following two points: After SFA requires random number a, the period r of $a^x \mod N$ must be an even number. And The factor of the output must be a non-trivial factor of N.

This paper optimizes the algorithm from a different angle from the previous:

(1) In a few cases, inputting a perfect square number a does not necessarily get an odd period r (For example, $a = 4, N = 35$, get a period $r = 6$). If you remove all perfect squares a, this may waste a suitable desirable random number a. After research, we found that a perfectly square random number a can still complete the algorithm when an odd cycle r is obtained. In this way, we can avoid the situation of removing the appropriate random number a in the previous algorithm.
(2) If a is not a perfect square, the algorithm has to be restarted when an odd period r is obtained. After improvements, we implemented the algorithm to still complete when the period r is a multiple of 3. (For example, $a = 20, N = 361$, and the period $r = 57$ is obtained)

The following chapter will describe some of the current research results in this field in the second chapter after the introduction chapter, Including the encryption principle of RSA and the calculation principle of SFA, The third chapter proposes a new optimization scheme, and the fourth chapter proves the effect of the optimization scheme by experiments. Finally, the fifth chapter concludes.

2 Related works

2.1 RSA public key cryptography based on integer factorization

Rivest, Shamir and Adelman proposed RSA public key cryptosystem in 1977[12], This cryptosystem is a typical cryptosystem based on integer factorization problem. Rivest, Shamir and Adelman were awarded the Turing Award in 2000 by the Academy of Computing Machinery (ACM) for their contributions to the theory and practical application of public key cryptosystems, especially the invention of RSA cryptosystem. RSA public key cryptosystem is based on the idea that finding two large prime numbers is not difficult, but factoring a large composite number into its prime factorized form is very difficult[13].

Let p and q be two large prime numbers with the same binary length. The binary length satisfying $n = pq$ is not less than 1024 bits, and both $p-1$ and $q-1$ have large prime factors, and is called a RIPE composite number.

Given a RIPE composite number$n = pq$and positive odd number e that satisfies $\gcd(e, \phi(n)) = 1$, for any given random integer $C \in \mathbb{Z}_n^*$ Find an integer M that satisfies $M^e \equiv C(\mathrm{mod} n)$, Call the problem the RSA problem which is, $\{e, n, C \equiv M^e(\bmod n)\} \to \{M\}$.

RSA public key cryptosystem can be defined as :

$$\mathrm{RSA} = \{\mathcal{M}, \mathcal{C}, \mathcal{K}, M, C, e, d, n, E, D\}$$

where $\mathcal{M}$ is the set of plaintexts, called the plaintext space. is the set of ciphertexts, called the ciphertext space. $\mathcal{K}$. is the set of keys, called the key space. $M \in \mathcal{M}$ is a special plaintext. $C \in \mathcal{C}$ is a special ciphertext. $=$ is the modulus, and are distinct large prime numbers, usually with at least 100 digits. $\{(e, n), (d, n) \in \mathcal{K}\}$, is the public encryption exponent (public key), is the private decryption exponent (key) and satisfy $ed \equiv 1(\mathrm{mod}\phi(n))$, $\phi(n) = (p-1)(q-1)$ is the Euler function. E is the encryption function.

$$E : M \to C$$

That is

$$C \equiv M^e(\mathrm{mod} n)$$

D is the decryption function

$$D : C \to M$$

That is

$$M \equiv C^d(\mathrm{mod} n)$$

Obviously, the encryption function $E : M \to C$ is a one-way trapdoor function[14] because it is easy to compute by the fast exponential algorithm. And its inverse $D : C \to M$ is intractable, because for those who don't know

the decryption key (trapdoor information) d, in order to find d they will have to factorize and compute $\phi(n)$. However, for those who know d, the calculation of D is as simple as the calculation of E.

2.2 Shor's factorization algorithm

Shor first proposed the quantum integer factorization algorithm in 1994, which immediately attracted the research of many researchers, and also set off the climax of quantum computer research. The integer factorization problem takes exponential time to complete on the classical computer, thus ensuring the security of the RSA public key cryptosystem. However, SFA fully demonstrates the advantages of quantum algorithms in solving some classical problems. For example, under the condition of quantum computing, integer factorization problems can be solved in polynomial time. This means that once the quantum computer is applied, it will bring a devastating blow to the cryptosystem[15], and this alone is enough to cause people to pay great attention to the SFA quantum algorithm. The core of the SFA quantum algorithm is to use a special structure hidden in the factorization problem, which allows the integer factorization problem to be reduced to the problem of finding the period of a specific function. Using some theorems in number theory, the integer factorization problem is transformed into finding the period of a certain periodic function. The attack on RSA can be realized by using SFA[16].

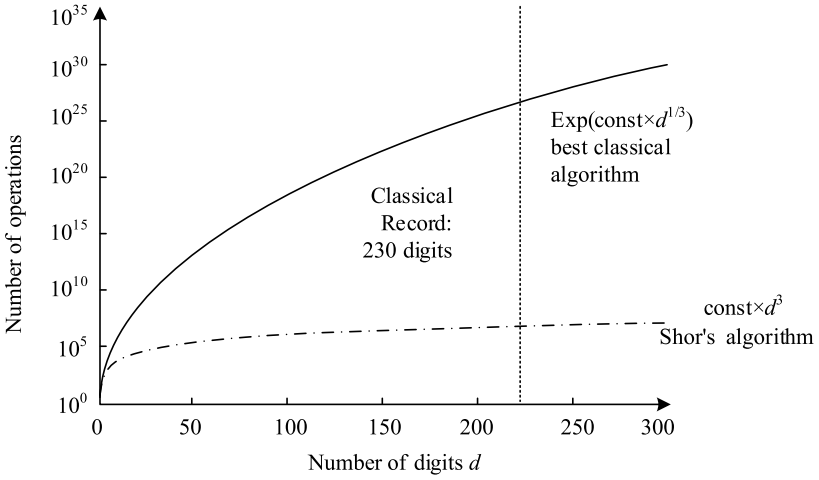

Fig. 1 Shor's algorithm has exponential acceleration effect compared with classical algorithm

As the number of large numbers on the abscissa continues to increase, the ordinary prime factorization algorithm obviously increases quickly, but Shor's algorithm performs exceptionally well here.

2.2.1 Quantum Registers and Quantum Fourier Transforms

Usually, only a single qubit can not complete the established computing goals, quantum registers with multiple qubits like conventional computers could be one approach. Generally speaking, a quantum register is a collection of quantum bits[17], which is a bit string whose length determines the amount of information it can store. A register of length n qubits is the superposition of all 2n possible strings of length qubits represented by n bits,In other words, the state space of a quantum register of length n is a linear combination of n-bit basis vectors, each of length 2n[18], So we can get Eqs. 1:

$$|\psi_n\rangle = \sum_{i=0}^{2^n-1} a_i \, |i\rangle \tag{1}$$

The discrete Fourier transform uses the position of the peaks to guess the length of one cycle of the original sequence. We can construct a new structure in a quantum computer to correspond to the discrete Fourier transform with a time complexity of $O\left(n^2\right)$.

For example, when $n = 4$, the quantum circuit diagram as Fig. 2:

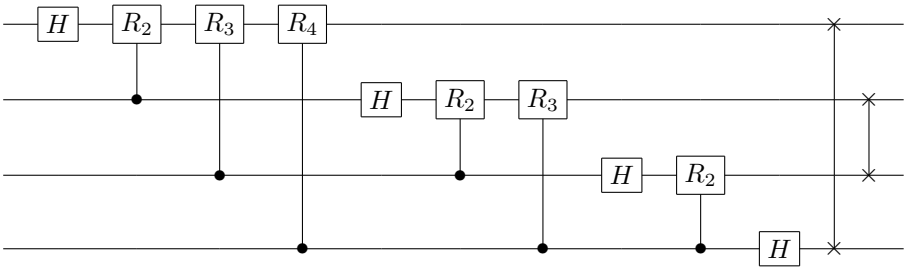

Fig. 2 The quantum circuit diagram when n = 4

When n is any value:

$$R_k = \begin{bmatrix} 1 & 0 \\ 0 & e^{2\pi i/2^k} \end{bmatrix}$$

2.2.2 Shor's factorization algorithm process

We go back to the function $f(x) = a^x \bmod N$ itself, as long as its cycle is found, all the mysteries will be solved, and SFA will be successfully completed. Known: Let be an odd number, then n has a true factor if and only if the congruence equation $x^2 \equiv 1(\bmod n)$has a non-trivial solution. $(a \in Z)$ is called a non-trivial solution of $x^2 \equiv 1(\bmod n)$ if $a^2 \equiv 1(\bmod n)$ and $a \neq \pm 1(\bmod n)$).

The quantum circuit diagram of SFA is as follows Fig. 3:

We need to initialize two quantum registers. The first quantum register is to store the t qubits we input, which is the exponent of a in the function, that

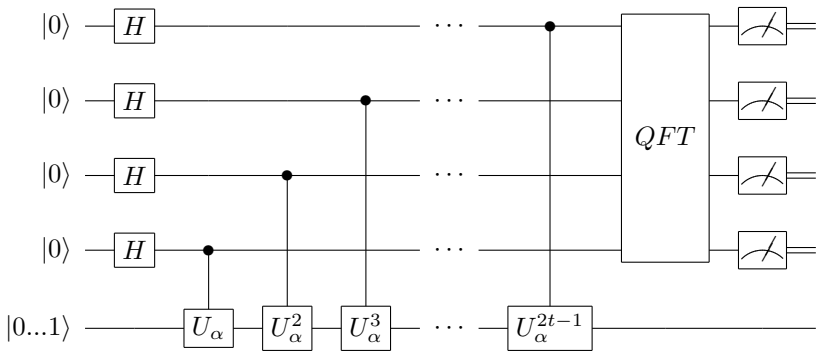

Fig. 3 The quantum circuit diagram of Shor's factorization algorithm

is, the input value. The second quantum register is used to store the quantum state of the result of the function

$$f(x) = a^x \mod N \tag{2}$$

The process of Shor's algorithm to crack RSA:

Input: n

Output: r

Step 1: Choose a such that $\gcd(a, n) = 1$ and choose q, where $a \in \mathbb{Z}_n^* and\, n^2 \leq q = 2^t < 2n^2$

Step 2: Given two quantum registers, initialized to zero state $|\Psi_0\rangle = |0\rangle\,|0\rangle$.

Step 3: Perform the Hadamard transform on the first quantum register

$$H : |\Psi_0\rangle \to |\Psi_1\rangle = \frac{1}{\sqrt{q}} \sum_{x=0}^{q-1} |x\rangle\,|0\rangle \tag{3}$$

Step 4: Do modular exponentiation U_f on the second quantum register to get

$$\begin{aligned} U_f : |\Psi_1\rangle \to |\Psi_2\rangle &= \frac{1}{\sqrt{q}} \sum_{x=0}^{q-1} |x\rangle\,|a^x (\mathrm{mod} n)\rangle \\ U_f : |x\rangle\,|y\rangle \to |x\rangle\,|y \oplus f(x)\rangle\,, f(x) &= a^x \bmod n \end{aligned} \tag{4}$$

Step 5: Perform a quantum Fourier transform [10] on the first register, which maps each state $|x\rangle$ to:

$$\frac{1}{\sqrt{q}} \sum_{x=0}^{q-1} e^{2\pi i x c/q}\,|c\rangle \tag{5}$$

That is, act on the unitary matrix so that the element at position (x, c) is $\frac{1}{\sqrt{q}} \sum_{x=0}^{q-1} e^{2\pi i x c/q}$. This leaves the register in state $|\Psi_3\rangle$,That is

$$\begin{aligned} |\Psi_3\rangle &= \mathrm{QFT}\left(|\Psi_2\rangle\right) \\ &= \mathrm{QFT}\left(\frac{1}{\sqrt{q}} \sum_{x=0}^{q-1} |x\rangle \left|a^x (\mathrm{mod} n)\right\rangle\right) \\ &= \frac{1}{q} \sum_{x=0}^{q-1} \sum_{c=0}^{q-1} e^{\frac{2\pi i x c}{q}} |c\rangle \left|a^x (\mathrm{mod} n)\right\rangle \end{aligned} \tag{6}$$

Step 6: Observe the register. Suppose that state $\left|a^l \bmod n\right\rangle$ is observed, where $0 \leq l < r$. At this time, the state in the first register will also collapse to all x satisfying, which $x = l \bmod r$, set $x = br + l$.

Step 7: Get the desired value of r, the probability amplitude of the obtained state $|c\rangle \left|a^l \bmod n\right\rangle$ is:

$$\begin{aligned} \mathrm{Prob}\left(c, a^l (\bmod n)\right) &= \left| \frac{1}{q} \sum_{x=0}^{q-1} e^{2\pi i x c/q} \right|^2 \\ &= \left| \frac{1}{q} \sum_{b=0}^{(q-l-1)/r} e^{2\pi i (br+l) c/q} \right|^2 \\ &= \left| \frac{1}{q} \sum_{b=0}^{(q-l-1)/r} e^{2\pi i (br+l) c/q} \right|^2 \end{aligned} \tag{7}$$

$\{rc\}$ is $rc (\mathrm{mod}\, n)$ because

$$\begin{aligned} \frac{-r}{2} \leq \{rc\} \leq \frac{r}{2} \Rightarrow \frac{-r}{2} \leq rc - dq \leq \frac{r}{2}, to\, any\, \mathbf{d} \\ \Rightarrow \mathrm{Prob}\left(c, a^l (\mathrm{mod}\, n)\right) > \frac{1}{3r^2} \end{aligned}$$

So we have:

$$\left| \frac{c}{q} - \frac{d}{r} \right| \leq \frac{1}{2q}$$

because $\frac{c}{q}$ is known, Use the continuous fraction algorithm to calculate, and judge whether r is the order of a. If yes, continue, otherwise go back to step 1 until the correct order r is found.

Step 8: If r is odd, reselect a. If r is even, then calculate $\gcd\left(a^{r/2} \pm 1, n\right) = (p, q)$.If $p, q \neq 1$, then it is a factor of n, otherwise re-select a for calculation. Further, it is possible to break RSA. because

$$ed \equiv 1 \bmod (\phi(n)) \tag{8}$$

therefore obtainable

$$d \equiv 1/e \bmod ((p-1)(q-1)) \tag{9}$$

thereby calculating

$$M \equiv C^d (\bmod n) \tag{10}$$

is the plaintext of RSA.

2.2.3 Analysis of the algorithm results:

The total required qubits for the SFA quantum decomposition algorithm is $3[\log n]$ SFA proves that the probability of success of running the algorithm once is $4\phi(r)/\pi^2 r$, where $4\phi(r)/\pi^2 r$ is the Euler function and is the order of a modulo $n, a \in \mathbb{Z}_n^*$. It can be seen that the success probability of SFA depends on the order r of a modulo n.

Known theorem[19]: Let $n = p_1^{\alpha_1} ... \cdots p_m^{\alpha_m}$ be the prime factorization of a positive odd composite number, let x be an integer chosen uniformly at random within $1 \leq x \leq n-1$, and x and n are relatively prime, let r be the order of mod, then there are:

$$p\left(r\, is\, even,\, and\, x^{r/2} \neq -1 (\bmod n)\right) \geq 1 - \frac{1}{2^m}.$$

inference: Let $n = pq$, $x$is an integer coprime to n randomly selected from [0, n], The order of $x$modulo n is r, Then the probability of integer factorization of n using x is $p \geq 3/4$.

3 Our Solution

3.1 Optimization ideas

First, let's take a look at the traditional SFA flow chart in Fig. 4:

Based on the above content, it can be seen that SFA is not necessarily successful in cracking the RSA encryption system[20],usually a constant number of attempts are required, making the success rate of the algorithm infinitely close to 1. Its shortcomings are reflected in the following aspects:

1) If $gcd(a, N) \neq 1$, you need to go back to the first step and re-select $1 < a < N$.
2) If the period r of the constructor $f(x) = a^x \bmod N$ is found to be an odd number, it is necessary to return to the first step and re-select $1 < a < N$.
3) If $a^{\frac{r}{2}} + 1 = 0 (\bmod N)$, also need to go back to the first step.

Obviously, if $gcd(a, N) \neq 1$, then the factor P can be directly obtained using Euclid's algorithm (rolling division method) [10]. Whether a suitable period r can be obtained or not largely determines the success of cracking RSA. If we can expand the available range of the second step cycle r, the execution efficiency of SFA can be effectively increased.

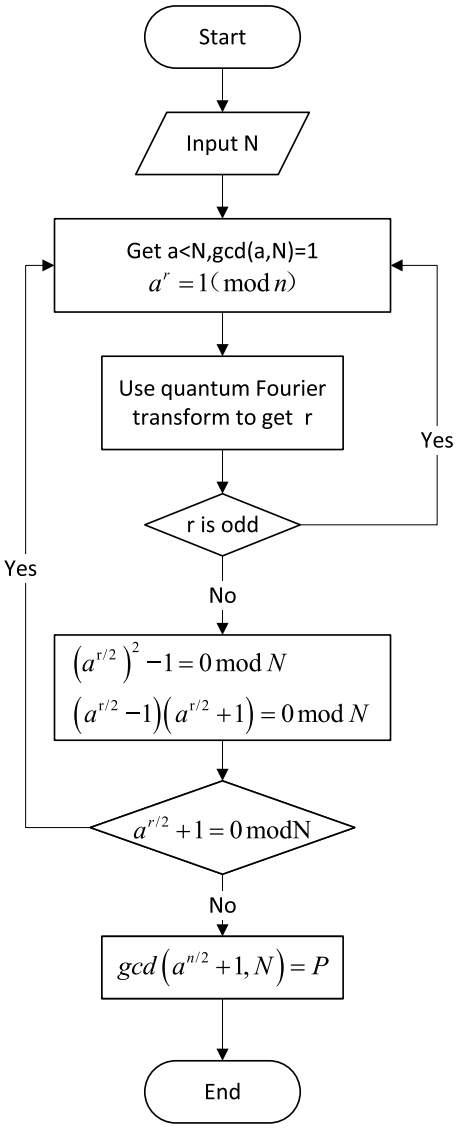

Fig. 4 The traditional Shor's factorization algorithm flow chart

3.2 Relax the restriction on the period of the function when r is a multiple of 3

When the value of r is odd, changes are made to the subsequent algorithm. If r is a multiple of 3, then:

$$\begin{aligned} \left(a^{r/3}\right)^3 - 1 = 0 \mod N \\ \left(a^{r/3} - 1\right)\left(a^{2r/3} + a^{r/3} + 1\right) = 0 \mod N \end{aligned} \tag{11}$$

If $\alpha^{\frac{2r}{3}} + \alpha^{\frac{r}{3}} + 1 \neq 0 \bmod N$ and $\alpha^{\frac{r}{3}} - 1 \neq 0 \bmod N$, get $\gcd\left(\alpha^{\frac{r}{3}} - 1, N\right)$ or $\gcd\left(\alpha^{\frac{2r}{3}} + \alpha^{\frac{r}{3}} + 1, N\right)$, The decomposition factor P can be obtained, and the algorithm ends.

The optimized SFA flow chart in Fig. 5.

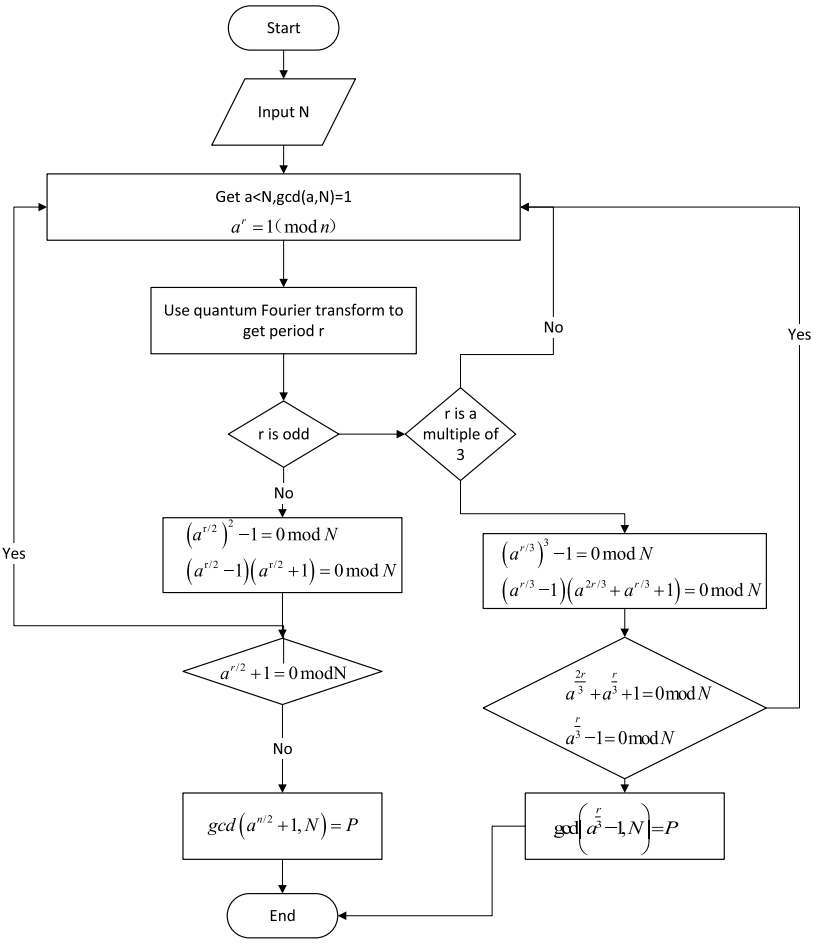

Fig. 5 The optimized SFA flow chart

For example: decomposition $N = 35$ get random number $a = 21$, Satisfy $1 < a < 35$, $gcd(21, 35) = 1$ Constructor $11^x = 1 (\bmod 35)$, Through the quantum Fourier transform, the period obtained in the polynomial complexity is 3. Then there are

$$\begin{gathered}\left(11^{3/3}\right)^3 - 1 = 0 \mod N \\ \left(11^{3/3} - 1\right)\left(11^{2*3/3} + a^{3/3} + 1\right) = 0 \mod N\end{gathered} \tag{12}$$

Obviously,

$$11^{\frac{2*3}{3}} + 11^{\frac{3}{3}} + 1 \neq 0 \mod N \tag{13}$$

$$11^{\frac{3}{3}} - 1 \neq 0 \mod N \tag{14}$$

So we can get

$$P_1 = \gcd\left(11^{\frac{3}{3}} - 1, 35\right) = \gcd(10, 35) = 5 \tag{15}$$

$$P_2 = \gcd\left(11^{\frac{2*3}{3}} + 11^{\frac{3}{3}} + 1, 35\right) = \gcd(133, 35) = 7 \tag{16}$$

Algorithm ends.

It can be seen from the above derivation and examples that by relaxing the range of the applicable period r, the success rate of using SFA to crack RSA has been improved, indicating that the optimization idea is effective.

3.3 When is a perfect square, odd cycles can still complete the algorithm

For the decomposition of large integers, if a is a perfect square, the result can still be obtained when the period r is odd after making certain changes to the SFA.

a is a perfect square,then a is an integer.

So,

$$\begin{gathered}\left(\sqrt{a}^{\mathrm{r}}\right)^2 - 1 = 0 \mod N \\ \left(\sqrt{a}^{\mathrm{r}} - 1\right)\left(\sqrt{a}^{\mathrm{r}} + 1\right) = 0 \mod N\end{gathered} \tag{17}$$

At this time, although r is an odd number, But if$\left(\sqrt{a}^{\mathrm{r}} - 1\right) \neq 0 \mod N$ and $\left(\sqrt{a}^{\mathrm{r}} + 1\right) \neq 0 \mod N$, get $\gcd\left(\sqrt{a}^{\mathrm{r}} + 1, N\right)$ or $\gcd\left(\sqrt{a}^{\mathrm{r}} - 1, N\right)$ to get the decomposition factor P.

Algorithm ends.

For example: when N equals 35

Take $a = 16$ (the perfect square of $4 * 4$), the constructor:

$$16^x = 1(\bmod 35)$$

Through the quantum Fourier transform, the period r of x is obtained in polynomial complexity to be 3. Then there are

$$\begin{gathered}\left(16^{3/2}\right)^2 - 1 = 0 \mod n \\ \left(16^{3/2} + 1\right)\left(16^{3/2} - 1\right) = 0 \mod n\end{gathered} \tag{18}$$

Calculate

$$\left(\sqrt{16}^{3}-1\right) \neq 0 \mod N \tag{19}$$

$$\left(\sqrt{16}^{3}+1\right) \neq 0 \mod N \tag{20}$$

So we get

$$P_1 = \gcd\left(\sqrt{16}^{3}-1, 35\right) = \gcd(63, 35) = 7 \tag{21}$$

$$P_2 = \gcd\left(\sqrt{16}^{3}+1, 35\right) = \gcd(65, 35) = 5 \tag{22}$$

Algorithm ends.

According to the above analysis, we get the final optimization scheme in Fig. 6.

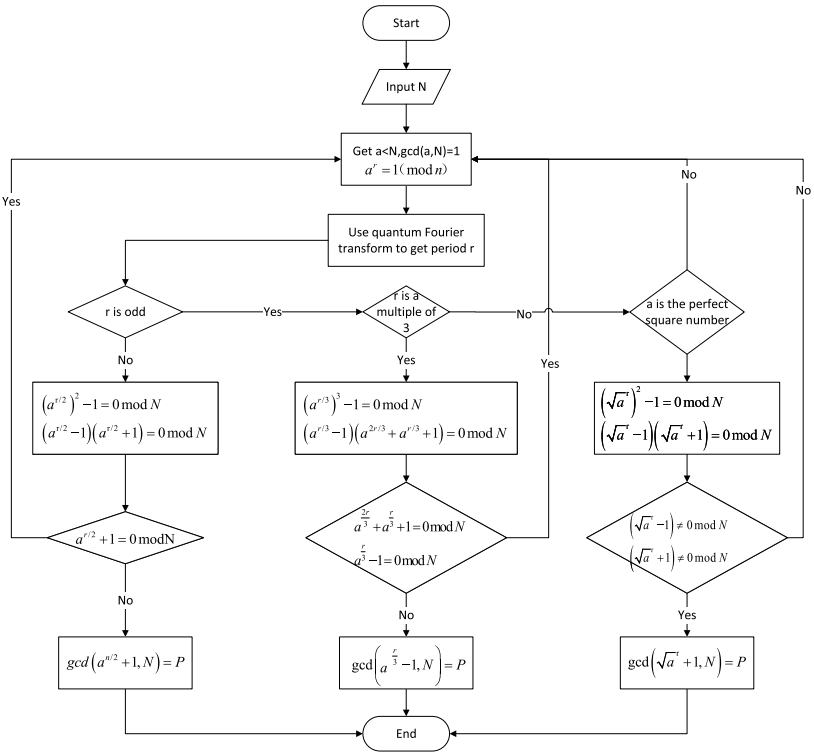

Fig. 6 The final optimization scheme

4 Experiment

This chapter designs two algorithms to verify the optimization effect.

4.1 Prove the optimization effect of relaxing the restriction on the period of the function $f(x)$ when r is a multiple of 3

In order to facilitate the calculation of the cycle r for different N when the value of a is different, the following program is written:

Input: Product N of two large prime numbers

Output: All values a that satisfy $gcd(a, N) = 0, 1 < a < N$, The frequency of the case where the period r of $a^x = 1(\bmod n)$ is an even number, and the frequency of the case where r is an odd number but a multiple of 3.

Take different a, the probability of different cases of function period r in Fig. 7 and Tab. 1.

Table 1 The probability distributions of r for different periods of N

	Probability		
N	P1: r is even	P2: r is odd and multiple of 3	Total
209	0.754	0.223	0.977
299	0.878	0.083	0.961
361	0.501	0.445	0.946
507	0.877	0.083	0.96
713	0.751	0.166	0.917
767	0.876	0.083	0.959
899	0.876	0.083	0.959
961	0.5	0.333	0.833

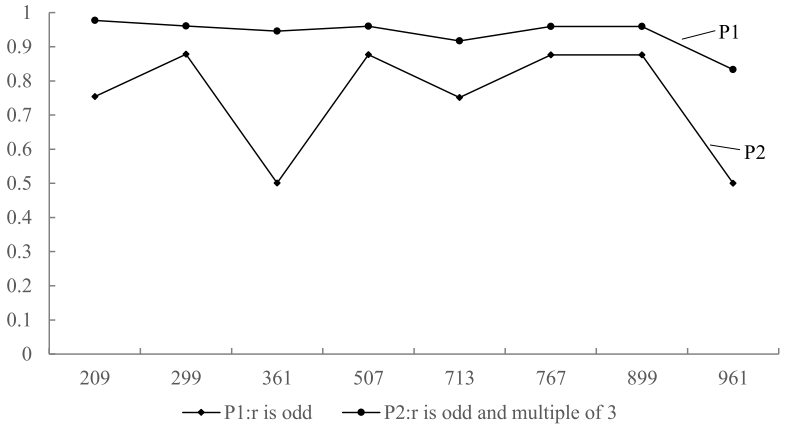

Fig. 7 The probability of different cases of period r

For composite numbers of the form $N = pq$, when p, q take all optional primes, calculate the probability of parameter a suitable for decomposition. The calculation results are shown above. Due to the limitations of classical computers, we randomly selected a combination of numbers less than 1000, and found that the optimization degree of this optimization scheme for different N. Selecting representative data, it can be seen that for any sample, the value of $P1$ will not exceed $P2$, and the success rate of cracking RSA is significantly improved.

4.1.1 Prove when a is a perfect square, it is easy to get an odd period r

Next, for different N, the period r when a takes a perfect square is counted The algorithm is as follows:

Input: Large number N and coprime number a less than N

Output: Period r of constructor $a^x = 1(\bmod N)$

Let $N = 961, 589, 361$ as follows example in table:

Table 2 The value of period r when a is a perfect square

N	a	r	N	a	r	N	a	r
961	4	155	589	4	45	361	4	55
961	9	465	589	9	45	361	9	165
961	16	155	589	16	45	361	16	55
961	25	93	589	25	9	361	25	33
961	36	93	589	36	9	361	36	33
961	49	465	589	49	15	361	49	165
961	64	155	589	64	15	361	64	33

5 Conclusion

Quantum factorization algorithms represent a breakthrough in complexity theory and modern cryptography. In 2001, one of the pioneers in the field of quantum computing, Chuang, devised an experiment based on molecular NMR (nuclear magnetic resonance) to factorize integers 15[21]. The experimental results were published in the journal Nature. This is the first time that the main proof-of-principle of SFA has been achieved experimentally. However, so far, experimentally, the largest integer that can be decomposed using SFA is only 21. In other words, the scalability of this experiment is very challenging, which is a major problem currently facing the development of quantum computers. In this paper, the quantum decomposition algorithm has been analyzed many times, and the original algorithm has been modified from multiple angles to improve the speed and efficiency of the algorithm. Finally, the classical algorithm is used to simulate the decomposition of large numbers, and the experimental data shows the optimization range. In addition,

in the experiment, we found that when the large number N is determined, the cycle r obtained by different random numbers a always repeats a lot. This phenomenon needs to be further studied, and perhaps a more efficient optimization scheme can be proposed.

acknowledgements

The National Natural Science Foundation of China (No.61772295, 61572270, and 61173056). the PHD foundation of Chongqing Normal University(No.19XLB003). the Science and Technology Research Program of Chongqing Municipal Education Commission(Grant No.KJZD-M202000501). Chongqing Technology Innovation and application development special general project(cstc2020jscx-lyjsAX0002).

References

[1] Cramer, R., Shoup, V .: Signature schemes based on the strong RSA assumption. ACM Trans. Inf. Syst. Secur. 3(3), 46-51 (1999)

[2] Wiener, Michael J. "Cryptanalysis of short RSA secret exponents." IEEE Transactions on Information theory 36.3 (1990): 553-558.

[3] SFA, P .W.: Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. SIAM J. Comput. 26(5), 1484-1509 (1997).

[4] Li, Jun, et al. "An efficient exact quantum algorithm for the integer square-free decomposition problem." Scientific reports 2.1 (2012): 1-5.

[5] Leander, G.: Leander2002 (2002). arXiv:quant-ph/0208183

[6] Geller M R, Zhou Z. Factoring 51 and 85 with 8 qubits[J]. Scientific reports, 2013, 3(1): 1-5.

[7] WANG YaHui. Research on quantum attack methods of RSA cryptography. Diss. WuHan University, 2017.

[8] Smolin J A, Smith G, Vargo A. Oversimplifying quantum factoring[J]. Nature, 2013, 499(7457): 163-165.

[9] Geller M R, Zhou Z. Factoring 51 and 85 with 8 qubits[J]. Scientific reports, 2013, 3(1): 1-5.10. SFA, P .W.: Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. SIAM J. Comput. 26(5), 1484-1509 (1997).

[10] SFA, P .W.: Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. SIAM J. Comput. 26(5),

1484-1509 (1997).

[11] Cao Z, Cao Z. On SFA's factoring algorithm with more registers and the problem to certify quantum computers[J]. arXiv preprint arXiv:1409.7352, 2014.

[12] Cramer, R., Shoup, V .: Signature schemes based on the strong RSA assumption. ACM Trans.Inf. Syst. Secur. 3(3), 46-51 (1999).

[13] Boneh, Dan, and Matthew Franklin. "Efficient generation of shared RSA keys." Annual international cryptology conference. Springer, Berlin, Heidelberg, 1997.

[14] Zhou, Xin, and Xiaofei Tang. "Research and implementation of RSA algorithm for encryption and decryption." Proceedings of 2011 6th international forum on strategic technology. Vol. 2. IEEE, 2011.

[15] Yimsiriwattana, Anocha, and Samuel J. Lomonaco Jr. "Distributed quantum computing: A distributed Shor algorithm." Quantum Information and Computation II. Vol. 5436. SPIE, 2004.

[16] Gerjuoy, Edward. "Shor's factoring algorithm and modern cryptography. An illustration of the capabilities inherent in quantum computers." American journal of physics 73.6 (2005): 521-540. 17. Weinstein, Yaakov S., et al. "Implementation of the quantum Fourier transform." Physical review letters 86.9 (2001): 1889.

[17] Weinstein, Yaakov S., et al. "Implementation of the quantum Fourier transform." Physical review letters 86.9 (2001): 1889.

[18] Browne, Daniel E. "Efficient classical simulation of the quantum Fourier transform." New Journal of Physics 9.5 (2007): 146.

[19] Weinstein, Yaakov S., et al. "Implementation of the quantum Fourier transform." Physical review letters 86.9 (2001): 1889.

[20] Bhatia, Vaishali, and K. R. Ramkumar. "An efficient quantum computing technique for cracking RSA using Shor's algorithm." 2020 IEEE 5th international conference on computing communication and automation (ICCCA). IEEE, 2020.

[21] Vandersypen, Lieven MK, et al. "Experimental realization of Shor's quantum factoring algorithm using nuclear magnetic resonance." Nature 414.6866 (2001): 883-887.