

RESEARCH ARTICLE | JULY 10 2023

Improving the success rate of quantum algorithm attacking RSA encryption system

Yumin Dong  ; Hengrui Liu  ; Yanying Fu  ; Xuanxuan Che *J. Appl. Phys.* 134, 024401 (2023)<https://doi.org/10.1063/5.0153709>

Articles You May Be Interested In

Quantum color image encryption based on a novel 3D chaotic system

J. Appl. Phys. (March 2022)

Analysis of RSA and Shor's algorithm for cryptography: A quantum perspective

AIP Conf. Proc. (October 2024)

An efficient quantum circuit implementation of Shor's algorithm for GPU accelerated simulation

AIP Advances (February 2024)

Instruments for Advanced Science

- Knowledge
- Experience
- Expertise

[Click to view our product catalogue](#)

Contact Hiden Analytical for further details:
www.HidenAnalytical.com
info@hiden.co.uk

Gas Analysis

- dynamic measurement of reaction gas streams
- catalysis and thermal analysis
- molecular beam studies
- dissolved species probes
- fermentation, environmental and ecological studies

Surface Science


- UHV TPD
- SIMS
- end point detection in ion beam etch
- elemental imaging - surface mapping

Plasma Diagnostics

- plasma source characterization
- etch and deposition process reaction kinetic studies
- analysis of neutral and radical species

Vacuum Analysis

- partial pressure measurement and control of process gases
- reactive sputter process control
- vacuum diagnostics
- vacuum coating process monitoring



Improving the success rate of quantum algorithm attacking RSA encryption system

Cite as: J. Appl. Phys. 134, 024401 (2023); doi: 10.1063/5.0153709

Submitted: 11 April 2023 · Accepted: 15 June 2023 ·

Published Online: 10 July 2023



Yumin Dong,^{a)} Hengrui Liu,^{b)} Yanying Fu,^{c)} and Xuanxuan Che^{d)}

AFFILIATIONS

College of Computer and Information Science, Chongqing Normal University, Chongqing 401331, China

^{a)}Author to whom correspondence should be addressed: dym@cqnu.edu.cn

^{b)}Electronic mail: 2021210516054@stu.cqnu.edu.cn

^{c)}Electronic mail: 2021210516033@stu.cqnu.edu.cn

^{d)}Electronic mail: 2021210516023@stu.cqnu.edu.cn

ABSTRACT

Shor's factorization algorithm (SFA) aims at finding the non-trivial factor of a given composite number, but the algorithm does not always work. In some cases, it has to call back to the beginning of the algorithm to make recalculation. After the analysis of the principle of SFA and the characteristics of RSA public-key cryptography with a series of data calculations, it can be concluded that the random value selected by the algorithm is closely related to whether the obtained period is effective. Therefore, a new optimized scheme is proposed to tackle with this defect from two perspectives: (a) When the a value is a perfect square, the algorithm can be completed with an odd cycle r . (b) When the period r obtained by the randomly selected a value is a multiple of 3, the algorithm can be completed by modifying the decomposition method to relax the requirements for the period without affecting the complexity of the algorithm. Due to the limitations of hardware in applying the quantum algorithms, the classical algorithm is applied to simulate quantum algorithms to test the success rate of decomposition of some composite numbers. The result indicates the effectiveness of the improved algorithm, which significantly reduces the probability of repeated operations to save the quantum circuit resources.

Published under an exclusive license by AIP Publishing. <https://doi.org/10.1063/5.0153709>

I. INTRODUCTION

An RSA public-key encryption system is the most widely used type of public-key system.¹ RSA is named after the initials of its inventors Rivest, Shamir, and Adelman. The principle is that the product of two large prime numbers is 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 what classical computers can solve. To the surprise of mathematicians, physicists, and computer scientists, in April 1994, Shor³ 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. It randomly selects a number smaller than n and $\gcd(a, n) = 1$ [greatest common factor of (a, n)] and uses the quantum Fourier

transform to calculate the period of $f(x) = a^x \bmod n$ (indicates that a^x when it is divided takes the remainder).

From Euler's theorem in number theory $a^r \equiv 1 \bmod n$, we have $(a^{r/2} - 1)(a^{r/2} + 1) \equiv 0 \bmod n$.

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

In the previous SFA, we often could not get the appropriate cycle directly; therefore, we have to start from scratch. In order to reduce the waste of an 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 dissertation⁴ proposed an efficient and exact quantum algorithm for finding the square-free part of a large integer 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

15 April 2025 08:26:15

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 Ref. 5, and 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. However, it is worth mentioning that when a is squared, the odd cycle of $a^x \bmod n$ 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 $a^r \not\equiv -1 \pmod{n}$ or $a^r \not\equiv 1 \pmod{n}$, the factors of large numbers n cannot be obtained by calculating $\gcd(a^{r/2} + 1, n)$ or $\gcd(a^{r/2} - 1, n)$.

The literature⁶ 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, the literature⁷ 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 an 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 *et al.*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 two qubits are needed for the second register, thus greatly reducing the number of qubits. Geller and Zhou⁹ used the characteristics of a Fermat number to realize the decomposition of 51 and 85 with 8 qubits and gave the circuit diagram of quantum realization. Cao and Cao⁹ 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 $a^x \bmod n$ 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 realized that the algorithm is still complete when the period $r = 57$ is a multiple of 3 (for example, $a = 20$, $n = 361$, and the period $r = 57$ is obtained).

Through the experiments in Sec. IV A, it can be seen that the efficiency of the new algorithm in cracking RSA is significantly improved, and the success rate of the experimental samples is increased by 0.187 on average. According to the experimental data in 4.2, when a is a perfect square number, it is easy to get an odd number of period r . However, in the traditional SFA, the odd period r should be ignored. This results in that when the perfect square a is selected, the success rate of the algorithm is very low. The optimization scheme in this paper solves this problem.

Section II describes some of the current research results in this field, including the encryption principle of RSA and the calculation principle of SFA. Section III proposes a new optimization scheme, and Sec. IV proves the effect of the optimization scheme by experiments. Finally, Sec. VI concludes.

II. RELATED WORKS

Given two large primes, it is easy to multiply them together. However, given their product, finding their factors is not so easy. This is the key to many modern cryptosystems. In theory, SFA can find a fast method to solve the problem of integer decomposition, and it breaks several important cryptographic systems represented by an RSA public-key encryption algorithm.

A. RSA public-key cryptography based on integer factorization

Rivest, Shamir, and Adelman proposed an RSA public-key cryptosystem in 1977.¹ This cryptosystem is a typical cryptosystem based on an 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 an RSA cryptosystem. An 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.¹⁰

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 an RIPE composite number.

Given an RIPE composite number $n = pq$ and a positive odd number e that satisfies $\gcd(e, \phi(n)) = 1$, for any given random integer $C \in \mathbb{Z}_n^*$ (multiplicative group; this group consists of elements in \mathbb{Z}_n that are relatively prime to n) to find an integer M that satisfies $M^e \equiv C \pmod{n}$, and we call the problem the RSA problem, which is $\{e, n, C \equiv M^e \pmod{n}\} \rightarrow \{M\}$.

An RSA public-key cryptosystem can be defined as

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

where \mathcal{M} is the set of plaintexts, called the plaintext space. \mathcal{C} 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. $n = pq$ is the modulus and are distinct large prime numbers, usually with at least 100 digits. In $\{(e, n), (d, n) \in \mathcal{K}\}$, e is the public encryption exponent (public key), d is the private decryption exponent (key) and satisfies $ed \equiv 1 \pmod{\phi(n)}$, and

15 April 2025 08:26:15

$\phi(n) = (p-1)(q-1)$ is the Euler function. E is the encryption function,

$$E: M \rightarrow C,$$

That is,

$$C \equiv M^e \pmod{n}.$$

D is the decryption function,

$$D: C \rightarrow M,$$

That is,

$$M \equiv C^d \pmod{n}.$$

Obviously, the encryption function $E: M \rightarrow C$ is a one-way trapdoor function¹¹ because it is easy to compute by the fast exponential algorithm. However, its inverse $D: C \rightarrow M$ is intractable because for those who do not 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 .

B. 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,¹² 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.¹³

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 in Fig. 1.

1. Quantum registers and quantum Fourier transforms

Usually, only a single qubit cannot complete the established computing goals. Quantum registers with multiple qubits, such as conventional computers, could be one approach. Generally speaking, a quantum register is a collection of quantum bits,¹⁴ which is a bit string whose length determines the amount of information it

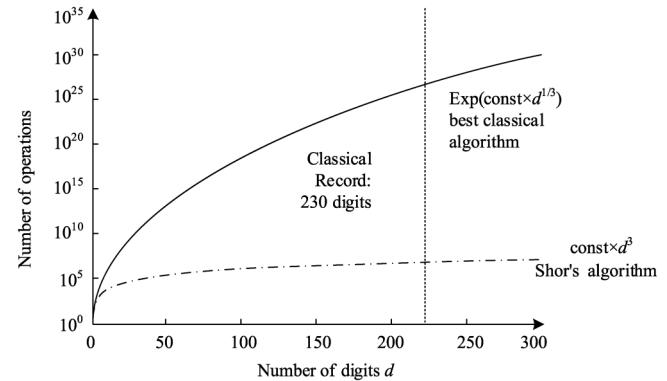


FIG. 1. Shor's algorithm has an exponential acceleration effect compared with the classical algorithm.

can store. A register of length n qubits is the superposition of all 2^n possible strings of length n 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 2^n ;¹⁵ therefore, we can get Eq. (1),

$$|\psi_n\rangle = \sum_{i=0}^{2^n-1} a_i |i\rangle. \quad (1)$$

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 $O(n^2)$.

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

When n is any value,

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

2. Shor's factorization algorithm process

We go back to the function $f(x) = a^x \pmod{n}$ itself. As long as its cycle is found, all the mysteries will be solved, and SFA will be successfully completed.

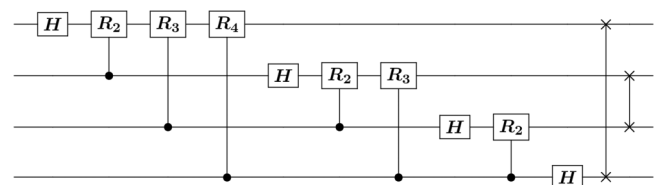


FIG. 2. The quantum circuit diagram when $n = 4$.

15 April 2025 08:26:15

Known: Let n be an odd number, and then n has a true factor if and only if the congruence equation $x^2 \equiv 1 \pmod{n}$ (the square of x is congruent with 1) has a non-trivial solution. ($a \in \mathbb{Z}$) is called a non-trivial solution of $x^2 \equiv 1 \pmod{n}$ if $a^2 \equiv 1 \pmod{n}$ and $a \not\equiv \pm 1 \pmod{n}$.

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

$$f(x) = a^x \bmod n. \quad (2)$$

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,

$$H: |\Psi_0\rangle \rightarrow |\Psi_1\rangle = \frac{1}{\sqrt{q}} \sum_{x=0}^{q-1} |x\rangle|0\rangle. \quad (3)$$

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

$$U_f: |\Psi_1\rangle \rightarrow |\Psi_2\rangle = \frac{1}{\sqrt{q}} \sum_{x=0}^{q-1} |x\rangle|a^x \bmod n\rangle, \quad (4)$$

$$U_f: |x\rangle|y\rangle \rightarrow |x\rangle|y \oplus f(x)\rangle, f(x) = a^x \bmod n.$$

Step 5: Perform a quantum Fourier transform³ on the first register, which maps each state $|x\rangle$ to

$$\frac{1}{\sqrt{q}} \sum_{x=0}^{q-1} e^{2\pi i x c / q} |c\rangle. \quad (5)$$

In other words, 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

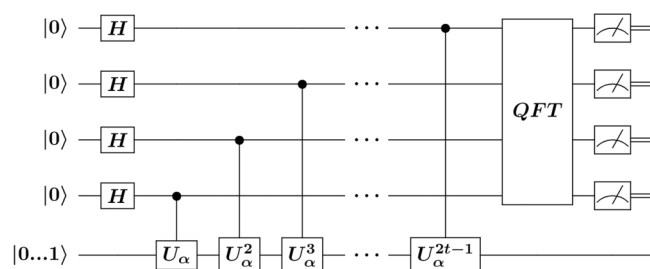


FIG. 3. The quantum circuit diagram of Shor's factorization algorithm.

$|\Psi_3\rangle$; that is,

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

Step 6: Observe the register. Suppose that state $|a^l \bmod n\rangle$ 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 $|c\rangle|a^l \bmod n\rangle$ is

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

rc is $rc \bmod n$ because

$$\begin{aligned} \frac{-r}{2} \leq rc \leq \frac{r}{2} &\Rightarrow \frac{-r}{2} \leq rc - dq \leq \frac{r}{2} \text{ to any } d \\ &\Rightarrow \text{Prob}(c, a^l \bmod n) > \frac{1}{3r^2}. \end{aligned}$$

Therefore, 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 $\gcd(a^{r/2} \pm 1, n) = (p, q)$. If $p, q \neq 1$, then it is a factor of n ; otherwise, reselect a for calculation. Furthermore, it is possible to break RSA. Because

$$ed \equiv 1 \pmod{\phi(n)}, \quad (8)$$

therefore, obtainable

$$d \equiv 1/e \pmod{(p-1)(q-1)}, \quad (9)$$

thereby calculating

$$M \equiv C^d(\text{mod } n), \quad (10)$$

is the plaintext of RSA.

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:¹⁴ Let $n = p_1^{\alpha_1} \dots p_m^{\alpha_m}$ 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, and let r be the order of mod, then there are

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

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 $p \geq 3/4$.

III. OUR SOLUTION

The success rate of traditional SFA is low, and we optimize the classical part of the algorithm from two different angles so that the large number n can be decomposed by the odd number period r in some cases. Thus, the success rate of the algorithm is improved, which saves the circuit resources of quantum computation.

A. Optimization ideas

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

Based on the content of Fig. 4, it can be seen that SFA is not necessarily successful in cracking the RSA encryption system,¹⁶ and 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 $f(x) = a^x \text{ mod } n$ is found to be an odd number, it is necessary to return to the first step and re-select $1 < a < n$.
- (3) If $a^{r/2} + 1 = 0(\text{mod } n)$, you 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).³ 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.

Therefore, we propose a new optimized scheme to address this limitation from two perspectives: (a) When the value of a is a perfect square, the algorithm can be completed with an odd cycle r . (b) When the period r obtained by the randomly selected a value is

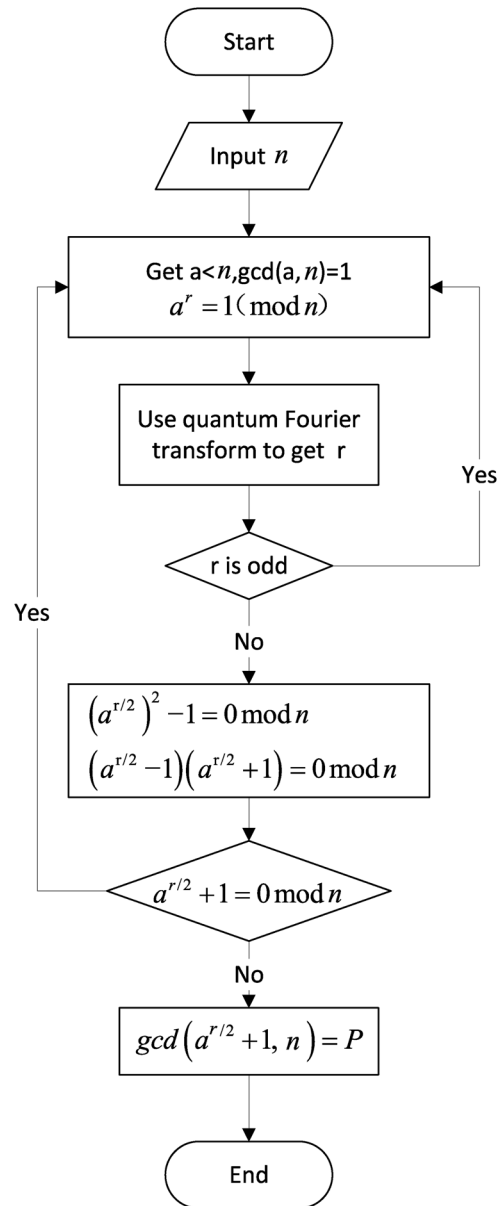


FIG. 4. The traditional Shor's factorization algorithm flow chart.

a multiple of 3, the algorithm can be completed by modifying the decomposition method. This modification relaxes the requirements for the period while maintaining the algorithm's complexity.

In other words, our optimization scheme aims to leverage the characteristics of these two specific cases. By adapting and improving the algorithm accordingly, we can achieve a higher success rate in attacking RSA encryption systems when r is a perfect square or a multiple of 3. These optimizations are designed to enhance the algorithm's performance without compromising its complexity.

B. 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,

$$\begin{aligned} (a^{r/3})^3 - 1 &= 0 \pmod{n}, \\ (a^{r/3} - 1)(a^{2r/3} + a^{r/3} + 1) &= 0 \pmod{n}. \end{aligned} \quad (11)$$

If $\alpha^{\frac{2r}{3}} + \alpha^{\frac{r}{3}} + 1 \not\equiv 0 \pmod{n}$ and $\alpha^{\frac{r}{3}} - 1 \not\equiv 0 \pmod{n}$, get $\gcd(\alpha^{\frac{r}{3}} - 1, n)$ or $\gcd(\alpha^{\frac{2r}{3}} + \alpha^{\frac{r}{3}} + 1, n)$. The decomposition factor P can be obtained, and the algorithm ends.

The optimized SFA flow chart in Fig. 5.

For example, decomposition $n = 35$, get random number $a = 21$, satisfying $1 < a < 35$, $\gcd(21, 35) = 1$. Constructor $11^x = 1 \pmod{35}$, through the quantum Fourier transform; the period obtained in the polynomial complexity is 3. Then, there are

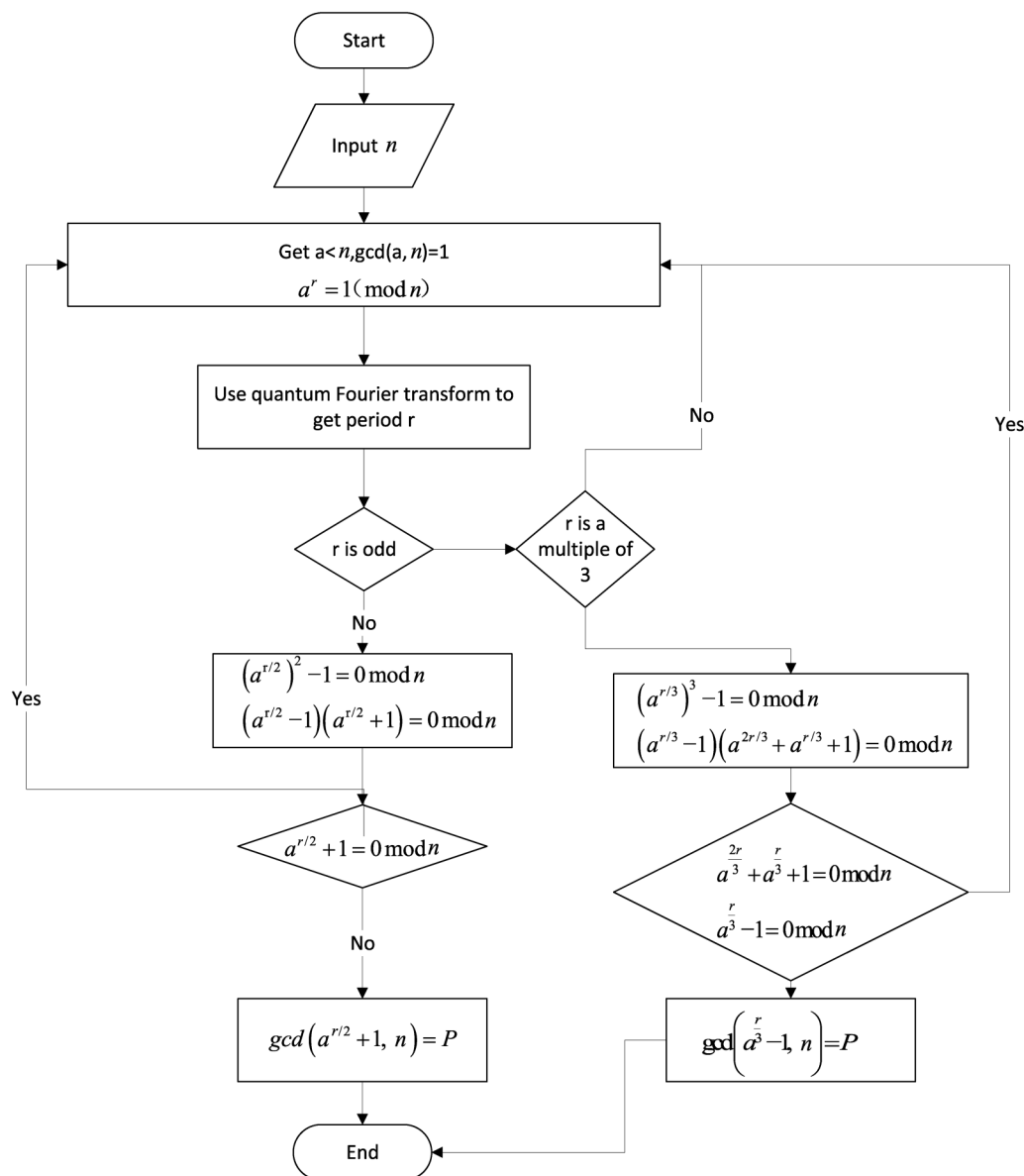


FIG. 5. The optimized SFA flow chart.

15 April 2025 08:26:15

$$\begin{aligned} (11^{3/3})^3 - 1 &= 0 \pmod{n}, \\ (11^{3/3} - 1)(11^{2 \cdot 3/3} + 11^{3/3} + 1) &= 0 \pmod{n}. \end{aligned} \quad (12)$$

Obviously,

$$11^{2 \cdot 3/3} + 11^{3/3} + 1 \neq 0 \pmod{n}, \quad (13)$$

$$11^{3/3} - 1 \neq 0 \pmod{n}. \quad (14)$$

Therefore, we can get

$$P_1 = \gcd(11^{3/3} - 1, 35) = \gcd(10, 35) = 5, \quad (15)$$

$$P_2 = \gcd(11^{2 \cdot 3/3} + 11^{3/3} + 1, 35) = \gcd(133, 35) = 7, \quad (16)$$

and the 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.

C. When a 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, and

$$\begin{aligned} (\sqrt{a^r})^2 - 1 &= 0 \pmod{n}, \\ (\sqrt{a^r} - 1)(\sqrt{a^r} + 1) &= 0 \pmod{n}. \end{aligned} \quad (17)$$

At this time, although r is an odd number, but if $(\sqrt{a^r} - 1) \neq 0 \pmod{n}$ and $(\sqrt{a^r} + 1) \neq 0 \pmod{n}$, get $\gcd(\sqrt{a^r} + 1, n)$ or $\gcd(\sqrt{a^r} - 1, n)$ to get the decomposition factor P , and the algorithm ends.

For example, when n equals 35, take $a = 16$ (the perfect square of 4^2), the constructor is

$$16^x = 1 \pmod{35}.$$

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

$$\begin{aligned} (16^{3/2})^2 - 1 &= 0 \pmod{n}, \\ (16^{3/2} + 1)(16^{3/2} - 1) &= 0 \pmod{n}. \end{aligned} \quad (18)$$

Calculate

$$(\sqrt{16^3} - 1) \neq 0 \pmod{n}, \quad (19)$$

$$(\sqrt{16^3} + 1) \neq 0 \pmod{n}. \quad (20)$$

Therefore, we get

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

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

and the algorithm ends.

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

IV. EXPERIMENT

This section designs two algorithms to verify the optimization effect.

A. 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 $a^x = 1 \pmod{n}$ 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 Table I.

For composite numbers of the form $n = pq$, when p, q take all optional primes, calculate the probability of parameters 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 the optimization degree of this optimization scheme for different N . Selecting representative data, it can be seen that for any sample, the value of P_1 will not exceed P_3 , and the success rate of cracking RSA is significantly improved.

B. 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 $a^x = 1 \pmod{n}$.

Let $n = 961, 589, 361$ as followed from the example in Table II.

According to the experimental data, when a is a perfect square, it is easy to get an odd number of periods r , but in the traditional SFA, the odd period r should be ignored. This results in that when the complete square a is selected, the success rate of the algorithm is very low. After optimization in Sec. C of this paper, no matter what period r is obtained, the algorithm can successfully decompose the large number n .

15 April 2025 08:26:15

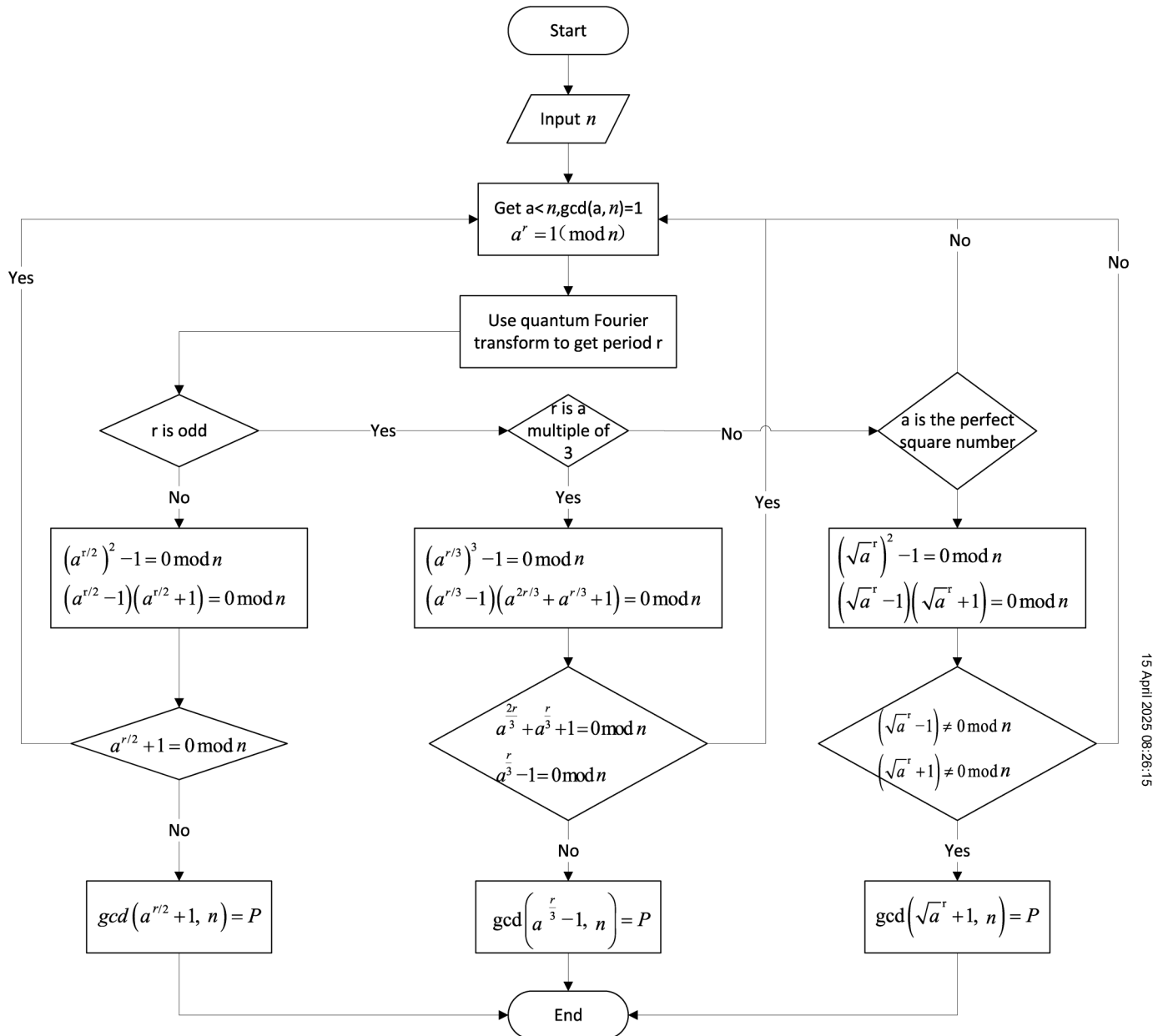


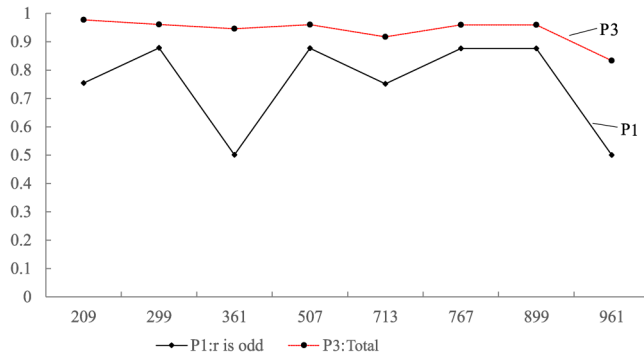
FIG. 6. The final optimization scheme.

V. DISCUSSION

Many scholars have tried to modify the classic part of SFA to improve its efficiency. The idea is always focused on reducing the probability of finding a useless cycle r . The most successful of these is Ref. 17, which shows that odd orders can be avoided by picking a coprime that is non-square under a modular arithmetic. This article reverses the logic of the predecessors, making the

odd-numbered cycles that need to be avoided as much as possible to be effective, and improved SFA success rates, covering their optimization range. Only under the optimization scheme of Sec. C is the success rate significantly improved, and the following is the mathematical proof:

Proof: We know that $n = pq$, and p and q are both prime numbers. $a \in Z_n^*$, where $Z_n^* = \{a \in Z \mid a < n, \gcd(a, n) = 1\}$.

FIG. 7. The probability of different cases of period r .

Let $p = 2^x + 1$, $q = 2^y + 1$, and $x \neq y$, then the probability of successfully cracking RSA is

$$2^{-(x+y)} \left(1 + \sum_{j=0}^{\min\{x,y\}-1} 4^j \right) \leq \frac{1}{2}.$$

Similarly, suppose there are numbers $f \in Z_p^*$, $g = f^s$, where s is odd, then the order of g satisfies $\text{ord}_p(g)s = 0 \pmod{p-1}$, which is that the order of $g \pmod{p}$ must be even.

If a has even order in Z_p^* or Z_q^* , then there are even orders in a in Z_n^* . It can be deduced that at least half of $a \in Z_n^*$ satisfies the following two conditions:

- (1) $\exists k$ satisfy $\text{ord}_n(a) = 2k$;
- (2) $a^k \not\equiv -1 \pmod{n}$.

Let $d = g^u$, v is the order of d , this is equivalent to $uv = 0 \pmod{p-1}$, and v is the smallest.

It has been deduced that the order of a belonging to Z_n^* is even, calculating $a \in Z_n^*$ does not satisfy condition (2), but satisfy the element of $(a/n) = -1$, equally computes $a^k \equiv -1 \pmod{p}$ and $a^k \equiv -1 \pmod{q}$. This means that neither u nor v can be divided. $(a/n) = -1$ means $(a/p) = -1$ and $(a/q) = 1$, or $(a/p) = 1$ and $(a/q) = -1$.

Assume $m \in Z_p^*$, $a = m^t$ (t is odd), need to calculate all even numbers t_1 that satisfy the condition. The order of $a = m^{t_1}$ is 2^i s,

TABLE I. The probability distributions of r for different periods of n probability.

n	P1: r is odd	P2: r is odd and multiple of 3	P3: 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

TABLE II. 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

where s is odd. At this time, all odd numbers between 1 and $p-1$ can be solved for t_1 .

Discuss elements related to q . It has been assumed that $q = 2^y + 1$, let $r \in Z_q^*$, $a = r^{t_2}$, and compute all even numbers t_2 that satisfy the condition. When $x = y$, t_2 has no even solution; when $x > y$, the order of r^{t_2} will divide r^{t_2} but has no such value. Therefore, we are only left with $x < y$.

When $x \neq y$, just consider case $x < y$, satisfying condition (1) and (2), all values satisfy

$$\begin{aligned} A(n) &= \frac{p-1}{2} 2^{(x-1)} \\ &= \frac{p-1}{2} \frac{2^y}{2^{(y-x+1)}} \frac{1}{4} (p-1)(q-1) \frac{1}{2^{(y-x)}} \\ &= \frac{1}{4} \varphi(n) \frac{1}{2^{(y-x)}}. \end{aligned}$$

Therefore, the probability of successfully cracking RSA is

$$p(n) = 1 - \frac{A(n)}{\varphi(n)} = 1 - \frac{1}{2^{(y-x+2)}} \geq \frac{3}{4}, x < y.$$

In Ref. 18, it is also proved that the lower limit of the success probability of an SFA algorithm is $1/2$. Xu *et al.* improved the success rate of SFA by at least $\frac{2}{\sqrt{N}}$.¹⁹ From the proof process, it can be seen that the SFA's success rate in cracking RSA is improved. It shows that the optimization idea is feasible and effective.

Still, there are some points in the optimization of SFA for further exploration. We can find a more appropriate way to choose a , working through the coprimes in order, $a = 2, 3, 4, \dots$, until the factors are found certainly not efficient. After a deeper understanding, we can keep improving SFA. In future experiments, we will replace the quantum part with a classical algorithm and reduce the number of calculations. This is of great significance for saving quantum computing resources.

VI. CONCLUSIONS

This paper presents a novel optimization scheme to address the drawback of repetitive callback computations in the Shor algorithm. We approach this issue from two perspectives:

When the value of a is a perfect square, the algorithm can be completed using an odd cycle r . Unlike traditional Shor's algorithm that requires multiple iterations to obtain an odd cycle, our

optimization scheme directly utilizes the properties of perfect squares to obtain the desired odd cycle r , eliminating the need for unnecessary computations.

When the period r obtained from randomly selected a values is a multiple of three, we modify the factoring method without impacting the algorithm's complexity. This modification relaxes the requirement on the period while preserving the algorithm's overall complexity. By allowing the period r to be a multiple of three, we increase the flexibility of the algorithm, reducing computational overhead under specific conditions.

Experimental results demonstrate the effectiveness of our optimization scheme. We observe a significant reduction in repetitive computations, resulting in an improved success rate for quantum algorithm attacks on RSA encryption systems. Our approach's innovation lies in the specific strategies we propose for perfect squares and periods divisible by three, which enhance the algorithm's efficiency and expand its applicability.

In conclusion, our paper introduces a new optimization scheme to address the issue of repetitive callback computations in the Shor algorithm. By considering perfect squares and periods divisible by three, we have significantly improved the success rate of quantum algorithm attacks on RSA encryption systems. The key innovation of our work lies in the specific methods we propose, which effectively alleviate the computational burden and enhance the algorithm's performance. These findings offer valuable insights and contribute to the advancement of quantum algorithm applications in the field of encryption.

ACKNOWLEDGMENTS

This work was funded by the National Natural Science Foundation of China (Nos. 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), and the Chongqing Technology Innovation and Application Development Special General Project (No. cstc2020jscx-lyjsAX0002).

AUTHOR DECLARATIONS

Conflict of Interest

The authors have no conflicts to disclose.

Author Contributions

Yumin Dong: Conceptualization (equal); Data curation (equal); Formal analysis (equal); Funding acquisition (equal); Writing – original draft (equal). **Hengrui Liu:** Validation (equal); Visualization (equal). **Yanying Fu:** Validation (equal); Visualization (equal). **Xuanxuan Che:** Validation (equal); Visualization (equal).

DATA AVAILABILITY

The datasets generated during and/or analyzed during the current study are available from the corresponding author upon reasonable request.

REFERENCES

- ¹R. Cramer and V. Shoup, "Signature schemes based on the strong RSA assumption," *ACM Trans. Inf. Syst. Secur.* **3**, 161–185 (2000).
- ²M. J. Wiener, "Cryptanalysis of short RSA secret exponents," *IEEE Trans. Inf. Theory* **36**, 553–558 (1990).
- ³P. W. Shor, "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer," *SIAM Rev.* **41**, 303–332 (1999).
- ⁴J. Li, X. Peng, J. Du, and D. Suter, "An efficient exact quantum algorithm for the integer square-free decomposition problem," *Sci. Rep.* **2**, 260 (2012).
- ⁵G. Leander, "Improving the success probability for Shor's factoring algorithm," *arXiv:quant-ph/0208183* (2002).
- ⁶M. R. Geller and Z. Zhou, "Factoring 51 and 85 with 8 qubits," *Sci. Rep.* **3**, 3023 (2013).
- ⁷Y. Wang, "Research on quantum attack methods of RSA cryptography," Ph.D. thesis (WuHan University, 2017).
- ⁸J. A. Smolin, G. Smith, and A. Vargo, "Oversimplifying quantum factoring," *Nature* **499**, 163–165 (2013).
- ⁹Z. Cao and Z. Cao, "On Shor's factoring algorithm with more registers and the problem to certify quantum computers," *arXiv:1409.7352* (2014).
- ¹⁰D. Boneh and M. Franklin, "Efficient generation of shared RSA keys," in *Advances in Cryptology—CRYPTO'97: 17th Annual International Cryptology Conference—Proceedings, Santa Barbara, CA, 17–21 August 1997* (Springer, 1997), pp. 425–439.
- ¹¹X. Zhou and X. Tang, "Research and implementation of RSA algorithm for encryption and decryption," in *Proceedings of 2011 6th International Forum on Strategic Technology* (IEEE, 2011), Vol. 2, pp. 1118–1121.
- ¹²A. Yimsiriwattana and S. J. Lomonaco, Jr., "Distributed quantum computing: A distributed Shor algorithm," *Proc. SPIE* **5436**, 360–372 (2004).
- ¹³E. Gerjuoy, "Shor's factoring algorithm and modern cryptography. An illustration of the capabilities inherent in quantum computers," *Am. J. Phys.* **73**, 521–540 (2005).
- ¹⁴Y. S. Weinstein, M. Pravia, E. Fortunato, S. Lloyd, and D. G. Cory, "Implementation of the quantum Fourier transform," *Phys. Rev. Lett.* **86**, 1889 (2001).
- ¹⁵D. E. Browne, "Efficient classical simulation of the quantum Fourier transform," *New J. Phys.* **9**, 146 (2007).
- ¹⁶V. Bhatia and K. Ramkumar, "An efficient quantum computing technique for cracking RSA using Shor's algorithm," in *2020 IEEE 5th International Conference on Computing Communication and Automation (ICCCA)* (IEEE, 2020), pp. 89–94.
- ¹⁷L. M. Vandersypen, M. Steffen, G. Breyta, C. S. Yannoni, M. H. Sherwood, and I. L. Chuang, "Experimental realization of Shor's quantum factoring algorithm using nuclear magnetic resonance," *Nature* **414**, 883–887 (2001).
- ¹⁸A. Ekert and R. Jozsa, "Quantum computation and Shor's factoring algorithm," *Rev. Mod. Phys.* **68**, 733 (1996).
- ¹⁹G. Xu, D. Qiu, X. Zou, and J. Gruska, "Improving the success probability for Shor's factorization algorithm," in *Reversibility and Universality: Essays Presented to Kenichi Morita on the Occasion of His 70th Birthday* (Springer, 2018), pp. 447–462.

15 April 2025 08:26:15