Wuhan University
Journal of Natural Sciences

2021, Vol.26 No.6, 489-494

Article ID 1007-1202(2021)06-0489-06
DOI https://doi.org/10.1051/wujns/2021266489

Quantum Algorithm for Attacking RSA Based on Fourier Transform and Fixed-Point

☐ **WANG Yahui[1], ZHANG Huanguo[2†]**

1. School of Computer and Information Technology, Xinyang Normal University, Xinyang 464000, Hubei, China;

2. School of Cyber Science and Engineering, Wuhan University, Wuhan 430072, Hubei, China

Abstract: Shor in 1994 proposed a quantum polynomial-time algorithm for finding the order r of an element a in the multiplicative group Z_n^*, which can be used to factor the integer n by computing $\gcd(a^{r/2}\pm 1, n)$, and hence break the famous RSA cryptosystem. However, the order r must be even. This restriction can be removed. So in this paper, we propose a quantum polynomial-time fixed-point attack for directly recovering the RSA plaintext M from the ciphertext C, without explicitly factoring the modulus n. Compared to Shor's algorithm, the order r of the fixed-point C for RSA(e, n) satisfying $C^{e^r} \equiv C \pmod{n}$ does not need to be even. Moreover, the success probability of the new algorithm is at least $4\varphi(r)/\pi^2 r$ and higher than that of Shor's algorithm, though the time complexity for both algorithms is about the same.

Key words: information security; cryptology; RSA fixed-point; quantum computing

CLC number: TP 391

Received date: 2021-09-28

Foundation item: Supported by Nanhu Scholars Program for Young Scholars of Xinyang Normal University

Biography: WANG Yahui, female, Ph. D., research direction: quantum computing and cryptography. E-mail: wangyh_ecc@whu.edu.cn

† To whom correspondence should be addressed. E-mail: liss@whu.edu.cn

0 Introduction

Since the discovery of quantum mechanics, people have paid much attention to quantum computers and quantum computing[1,2], which can perform some tasks, such as integer factorization problems, phase estimating problems, hidden subgroup problems, that are not feasible on a classical computer by using quantum parallelism and interference effect. In these quantum algorithms mentioned above, the Quantum Fourier Transform (QFT), which is a linear unitary transform, plays a significant role and lies in the core of the algorithms. Moreover, the QFT is one of the most important computational problems and many real-world applications require that the transform should be performed as quickly as possible.

It is well known that factoring an integer n can be reduced to finding the order of an integer a with respect to the module n. The order, r, of an element a in the multiplicative group Z_n^*, denoted by *order*(a, n), plays a significant role in the period of certain pseudo-random number generators, and particularly in Shor's quantum integer factorization algorithm and other cryptographic applications. So far as is known, there is not a polynomial time algorithm running on classical computers which can compute *order*(a, n) in polynomial time. The main idea of Shor's algorithm is simple: to factor n, one first computes the order r. If the computed r is even, then one further computes, with high probability, $\gcd(a^{r/2}\pm 1, n) = \{p, q\}$, with $1 < p, q < n$.

The world was astonished when Shor announced in

1994[3] that he found an efficient quantum integer factorization algorithm which can solve IFP (Integer Factrorization Problem) in time proportion to $O((\log n)^{2+\varepsilon})$. Security analysis of public key cryptosystems is of great significance in theory and practical application, especially the security of widely used public key cryptosystems such as RSA, ElGamal and ECC, which is worthy of further research[1]. Current research on quantum factoring is concentrated on various improved and compiled versions of Shor's original algorithm. Smolin *et al*[4] claimed that if one can find *a* such that *order* (*a*, *n*)=2, then Shor's quantum factoring algorithm can be implemented easily using two quantum bits. Peng *et al*[5] found an approach to implement the prime factorization of 21=3×7 based on the adiabatic theory. More recently, it extends to 143 on a Dipolar-Coupling Nuclear Magnetic Resonance System[6]. Wang *et al*[7] analyzed the RSA deciphering method based on D-Wave quantum annealing principle, which is a new attack algorithm for quantum computing.

There are three important research directions of quantum computing public-key cryptographic attacks:

1) Improve, modify and simply Shor's algorithm or even invent new quantum factoring algorithms to be run on quantum computers with fewer quantum bits[5, 8-12].

2) Quantum attack algorithms based on adiabatic quantum computing[6,7].

3) Quantum attack algorithms based on quantum annealing principle[13,14].

It has been known for a long time that there is no need to factor *n* if one just wish to break RSA.

In fact, to recover *M* from *C*, one could just compute the sequence of numbers (assume *C* is known):

$$\overline{C, C^{e}, C^{e^2}, C^{e^3}, \cdots, C^{e^{r-1}}},$$

$$\overline{C^{e^r}, C^{e^{r+1}}, C^{e^{r+2}}, \cdots, C^{e^{2r-1}}},$$

$$C^{e^{2r}}, C^{e^{2r+1}}, \cdots$$

where the overline symbol indicates the periodic elements. Once the first occurrence of $C^{e^{r-1}} \bmod n = C$ is found, the plaintext *M* is just the element $C^{e^{r-1}} \bmod n$ immediately preceding $C^{e^r} \bmod n$.

In classical computing, this process of computation is equivalent to the factorization of *n*, which is believed to be a hard problem. However, it can be done efficiently on a quantum computer, and it is even more convenient than Shor's original algorithm. In this paper, we shall propose a new quantum algorithm for directly recovering the RSA plaintext *M* from the ciphertext *C* by computing the order *r* of the fixed-point *C*, without explicitly factoring the modulus *n*, with higher success probability. Before discussing the algorithm, we present some basic concepts and results that will be used throughout the paper.

Definition 1[15] The RSA problem may be defined as follows. Given the RSA public-key (e,n) and the RSA ciphertext C, find the corresponding RSA plaintext M. That is,

$$\left\{ e, n, C \equiv M^e \pmod n \right\} \xrightarrow{\text{find}} \left\{ M \equiv C^d \pmod n \right\}$$

Definition 2 Let $0 \leqslant x < n$. If

$$x^{e^x} \equiv x \pmod n, r \in \mathbf{Z}^+, \tag{1}$$

then *x* is called a fixed-point of RSA (*e*, *n*) and the smallest *r* satisfying (1) is the order of the fixed-point.

Theorem 1 Let C to be the fixed-point of RSA (*e*, *n*) with order *r* :

$$C^{e^r} \equiv C \pmod n, r \in \mathbf{Z}^+ \tag{2}$$

then

$$C^{e^{r-1}} \equiv M \pmod n, r \in \mathbf{Z}^+ \tag{3}$$

where *M* is the plaintext, *C* is ciphertext, and *e* is the encryption key.

Proof See Ref. [15].

1 The New Algorithm

In this section, we shall present a polynomial-time quantum algorithm for computing the order *r* of the fixed-point *C* for RSA (*e*, *n*), such that $C^{e^r} \equiv C \pmod n$.

Algorithm 1 Quantum algorithm for attacking RSA based on Fourier transform and fixed-point

Input: C, e, n

Output: M

Step 1 Find a number *q*, a power of 2, say 2^t, such that $n^2 \leqslant q = 2^t < 2n^2$.

Step 2 Initialize the two quantum registers, Reg1 and Reg2, $|\Psi_0\rangle = |0\rangle|C\rangle$, where Reg1 requires $2\lceil \log n \rceil$ qubits, Reg2 requires $\lceil \log n \rceil$ qubits (whose number depends on the space requirement).

Step 3 Perform a Hadamard transform on Reg1, we get

$$H : |\Psi_0\rangle \to |\Psi_1\rangle = \frac{1}{\sqrt{q}} \sum_{x=0}^{q-1} |x\rangle |C\rangle \tag{4}$$

Step 4 Perform the unitary transform U_C^x on Reg2, we get

$$U_C^x : |\Psi_1\rangle \to |\Psi_2\rangle$$

$$= \frac{1}{\sqrt{q}} \sum_{x=0}^{q-1} |x\rangle U_C^x |C\rangle$$

$$= \frac{1}{\sqrt{q}} \sum_{x=0}^{q-1} |x\rangle \left| C^{e^x} \pmod{n} \right\rangle$$

where $U_C |y\rangle = \left| y^e \bmod n \right\rangle$, thus, $U_C^x |y\rangle = \left| y^{e^x} \bmod n \right\rangle$.

Step 5 Measure Reg2. Suppose we observe $m \equiv C^{e^l} \bmod n$, and at the same time, the state in Reg1 is collapsed into a superposition over all x such that $C^{e^x} \equiv m \bmod n$. That leaves Reg1 in state

$$|\Psi_3\rangle = \frac{1}{\sqrt{n_t + 1}} (|l\rangle + |l + r\rangle + \cdots + |l + n_l r\rangle) \tag{5}$$

where n_l is the largest positive integer satisfying $l + n_l r \leqslant 2^t - 1$.

Step 6 Perform QFT on Reg1.

$$\begin{aligned}
&\mathrm{QFT}(|\Psi_3\rangle) \\
&= \mathrm{QFT}\left(\frac{1}{\sqrt{n_l + 1}} (|l\rangle + |l + r\rangle + \cdots + |l + n_l r\rangle) \right) \\
&= \mathrm{QFT}\left(\frac{1}{\sqrt{n_l + 1}} \sum_{j=0}^{n_l} |l + jr\rangle \right) \\
&= \frac{1}{\sqrt{n_l + 1}} \mathrm{QFT}\left(\sum_{j=0}^{n_l} |l + jr\rangle \right) \\
&= \frac{1}{\sqrt{n_l + 1}} \frac{1}{\sqrt{q}} \sum_{c=0}^{q-1} \left(\sum_{j=0}^{n_l} e^{\frac{2\pi i (l + jr) c}{q}} \right) |c\rangle \\
&= \frac{1}{\sqrt{q(n_l + 1)}} \sum_{c=0}^{q-1} w_q^{lc} \left(\sum_{j=0}^{n_l} w_q^{jrc} \right) |c\rangle,
\end{aligned}$$

where $w_q = e^{\frac{2\pi i}{q}}$.

That is,

$$\mathrm{QFT}(|\Psi_3\rangle) = \sum_c f(c) |c\rangle \tag{6}$$

where $f(c) = \frac{1}{\sqrt{q(n_l + 1)}} w_q^{lc} \sum_{j=0}^{n_l} w_q^{jrc}$.

Step 7 Observe Reg1. This yields the state $|c\rangle$ with probability

$$\begin{aligned}
\mathrm{Prob}(c) &= |f(c)|^2 \\
&= \left| \frac{1}{\sqrt{q(n_l + 1)}} w_q^{lc} \sum_{j=0}^{n_l} w_q^{jrc} \right|^2 \\
&= \left| \frac{1}{\sqrt{q(n_l + 1)}} \sum_{j=0}^{n_l} w_q^{jrc} \right|^2 \\
&= \left| \frac{1}{\sqrt{q(n_l + 1)}} \sum_{j=0}^{n_l} w_q^{j(rc \bmod q)} \right|^2
\end{aligned}$$

Then we can use continued fraction method to find the closest to c/q among all the convergent of the continued fractions with their denominators less than n, thus its denominator is the required order r, similar to Shor's method[3] for obtaining r from the observation value c.

Step 8 Compute $M \equiv C^{e^{r-1}} (\bmod n)$, hence, the required plaintext M is obtained, that is, RSA is broken.

An example illustrating each of computational steps is given as follows.

Example 1 Let n=35, e=5, C=10.

Step 1 Find a number q such that $35^2 < q = 2^{11} = 2048 < 2 \times 35^2$

Step 2 Initialize the two quantum registers $|\Psi_0\rangle = |0\rangle |C\rangle$

Step 3 Perform a Hadamard transform on Reg1, we get

$$H : |\Psi_0\rangle \to |\Psi_1\rangle = \frac{1}{\sqrt{2048}} \sum_{x=0}^{2047} |x\rangle |10\rangle \tag{7}$$

Step 4 Perform the unitary transform U_{10}^x on Reg2, we get

$$\begin{aligned}
U_{10}^x : |\Psi_1\rangle &\to |\Psi_2\rangle \\
&= \frac{1}{\sqrt{q}} \sum_{x=0}^{q-1} |x\rangle U_{10}^x |C\rangle \\
&= \frac{1}{\sqrt{2048}} \sum_{x=0}^{2047} |x\rangle \left| 10^{5^x} (\bmod 35) \right\rangle
\end{aligned}$$

where $U_{10} |C\rangle = \left| 10^5 \bmod 35 \right\rangle$, thus, $U_{10}^x |10\rangle = \left| 10^{5^x} \bmod 35 \right\rangle$.

The computation of the detailed exponentiations 10^5 mod 35 is as follows

$$\mathrm{Reg2} = [10, 5, 10, 5, 10, 5, 10, 5, \cdots, 10, 5] \tag{8}$$

Step 5 Measure Reg2. Suppose we observe $10^{5^l} \equiv 5 \bmod 35$, this means that the state in Reg1 is collapsed into a superposition over all x such that $10^{5^x} \equiv 5 \bmod 35$. That leaves Reg1 in state

$$|\Psi_3\rangle = \frac{1}{\sqrt{1024}} (|1\rangle + |3\rangle + |5\rangle + \cdots + |2045\rangle + |2047\rangle) \tag{9}$$

Step 6 Perform QFT on Reg1.

$$\begin{aligned}
&\mathrm{QFT}(|\Psi_3\rangle) \\
&= \mathrm{QFT}\left(\frac{1}{\sqrt{1024}} (|1\rangle + |3\rangle + \cdots + |2045\rangle + |2047\rangle) \right) \\
&= \frac{1}{\sqrt{1024}} \mathrm{QFT}(|1\rangle + |3\rangle + \cdots + |2045\rangle + |2047\rangle) \\
&= \frac{1}{\sqrt{1024}} \frac{1}{\sqrt{2048}} \left(\sum_{c=0}^{2047} e^{\frac{2\pi i c}{2048}} |c\rangle + \sum_{c=0}^{2047} e^{\frac{2\pi i 3c}{2048}} |c\rangle + \cdots \right.
\end{aligned}$$

$$+\left(\sum_{c=0}^{2047} e^{\frac{2\pi i 2045c}{2048}}|c\rangle+\sum_{c=0}^{2047} e^{\frac{2\pi i 2047c}{2048}}|c\rangle\right)$$

$$=\frac{1}{2}|c\rangle-\frac{1}{2}|1024\rangle.$$

Step 7 Measure Reg1. Suppose that c=1024 is observed with a higher probability 1/2, and in fact, all other states are observed with the probability 0. Then use the continued fraction expansion

$$\frac{c}{q}=\frac{1024}{2048}=\frac{1}{2} \tag{10}$$

r=2 can be deduced.

Step 8 Compute $M\equiv 10^{5^{2-1}}(\mathrm{mod}\ 35)\equiv 5$, hence, the required plaintext is obtained, that is, RSA is broken. Table 1 summarizes the main processes and differences between Shor's algorithm and Algorithm 1 for breaking RSA.

Table 1 Comparison of Shor's algorithm and Algorithm 1 for breaking RSA

Shor's algorithm	Algorithm 1
1. choose a, compute the order r of a modulo n 2. if r is not even, go back to step 1 3. compute $\gcd(a^{r/2}\pm 1,n)=(a',b')$ 4. if $(a',b')\neq(p,q)$, go back to step 1 5. compute $d\equiv 1/e(\mathrm{mod}(p-1)(q-1))$ 6. compute $M\equiv C^d(\mathrm{mod}\,n)$	1. compute the period r of $C^{e^x}\bmod n$ 2. compute $C^{e^{r-1}}$, that is M 3. output M

In what follows, we give a performance analysis of the algorithm. The number of gates needed are $O(\log n)$ for the initial Hadamard in Step 3. The computation for the multiple modular $C^{e^x}\bmod n$ (which is a number between 1 and n 1), in Step 4, which takes time proportion to $O((\log n)^{2+\varepsilon})$. The QFT in Step 6 requires $O((\log n)^2)$ gates[16]. The classical continued fraction algorithm in Step 7 needs $O((\log n)^{2+\varepsilon})$ (classical) gates. Thus the quantum circuit of Algorithm 1 requires only $O((\log n)^2)$ elementary quantum gates. That is, Algorithm 1 breaks RSA in quantum polynomial-time $O((\log n)^{2+\varepsilon})$.

Now we estimate the size of the probability Prob(c). In Step 7, the probability Prob(c) that the machine reaches the state $|c\rangle$ ($0\leqslant c\leqslant q-1$) is

$$\mathrm{Prob}(c)=\frac{1}{q(n_l+1)}\left|\sum_{j=0}^{n_l} w_q^{j(rc\bmod q)}\right|^2 \tag{11}$$

Definition 3 If the state $|c\rangle$ was observed and r can be found correctly by Algorithm 1, then c is a good value.

Theorem 2 If there exists a positive integer d which is less than r and is prime to r, such that

$$\left|\frac{c}{q}-\frac{d}{r}\right|\leqslant\frac{1}{2q} \tag{12}$$

then c is a good value.

Proof We first introduce a lemma which will be used in the proof of the theorem.

Lemma 1 Suppose s/r is a rational number such that

$$\left|\frac{s}{r}-\varphi\right|\leqslant\frac{1}{2r^2}$$

Then s/r is a convergent of the continued fraction for φ.

Because $q\quad n^2, n\quad \varphi\ n\quad r$, thus

$$\left|\frac{c}{q}-\frac{d}{r}\right|\leqslant\frac{1}{2q}\leqslant\frac{1}{2n^2}\leqslant\frac{1}{2r^2}$$

Therefore, by Lemma 1, d/r must be a convergent of the continued fraction for c/q.

Suppose p_s/q_s is the closest to p/q among all the convergent of the continued fractions with their denominators less than n. Then we prove that $\frac{p_s}{q_s}=\frac{d}{r}$.

Because d/r is a convergent of the continued fraction for c/q and $r<n$, thus

$$\left|\frac{c}{q}-\frac{p_s}{q_s}\right|\leqslant\left|\frac{c}{q}-\frac{d}{r}\right|$$

and because

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

therefore

$$\left|\frac{p_s}{q_s}-\frac{d}{r}\right|\leqslant\left|\frac{p_s}{q_s}-\frac{c}{q}+\frac{c}{q}-\frac{d}{r}\right|\leqslant\frac{1}{2q}+\frac{1}{2q}=\frac{1}{q}$$

On the other hand,

$$\left|\frac{p_s}{q_s}-\frac{d}{r}\right|=\left|\frac{p_s r-q_s d}{q_s r}\right|>\frac{|p_s r-q_s d|}{n^2}\qquad\frac{|p_s r-q_s d|}{q}$$

Accordingly, $|p_s r - q_s d| = 0$, that is, $p_s / q_s = d / r$. Finally, because $\gcd(p_s, q_s) = 1 = \gcd(d, r)$, thus $q_s = r$. This shows that r is found correctly by Algorithm 1, so c is a good value.

Lemma 2 If c is a good value, then $|rc \bmod q| \leqslant \frac{r}{2}$.

Proof If c is a good value, by Definition 3, there exists a positive integer d which is less than r and is prime to r, such that

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

that is,

$$|rc - dq| \leqslant \frac{r}{2} < \frac{q}{2} \tag{13}$$

We denote

$$\frac{rc}{q} + \frac{1}{2} = \left\lceil \frac{rc}{q} + \frac{1}{2} \right\rceil + t, \quad 0 \leqslant t < 1 \tag{14}$$

thus, we can denote $rc \bmod q = \left(t - \frac{1}{2}\right) q$.

Of course,

$$-\frac{q}{2} \leqslant rc \bmod q < \frac{q}{2} \tag{15}$$

Using (13), (14) and (15), we can get

$$rc \bmod q = rc - dq \tag{16}$$

So

$$|rc \bmod q| \leqslant \frac{r}{2} \tag{17}$$

Since $\left|e^{ix} - 1\right|^2 = |\cos x + i \sin x - 1|^2 = 4\sin^2 \frac{x}{2}$, so the probability Prob (c) is

$$\begin{aligned} \mathrm{Prob}(c) &= \frac{1}{q(n_l + 1)} \left| \sum_{j=0}^{n_l} w_q^{j(rc \bmod q)} \right|^2 \\ &= \frac{1}{q(n_l + 1)} \frac{\sin^2 \frac{\pi(rc - dq)(n_l + 1)}{q}}{\sin^2 \frac{\pi(rc - dq)}{q}} \end{aligned}$$

Using the inequalities $4x^2 / \pi^2 \leqslant \sin^2 x \leqslant x^2$ (where the lower bound holds for $|x| \leqslant \pi/2$), we find

$$\mathrm{Prob}(c) \quad \frac{4}{\pi^2 r} \tag{18}$$

Thus, the probability of observing a good value $|c\rangle$ is at least $4/(\pi^2 r)$.

Then we wish to extract the information of the value of r, given a value of c satisfying

$$|rc \bmod q| \leqslant \frac{r}{2} \tag{19}$$

To do this we note that (19) is equivalent to

$$|rc - dq| \leqslant \frac{r}{2} \tag{20}$$

for some $0 \leqslant d \leqslant r - 1$.

Dividing by rq and rearranging the terms gives

$$\left|\frac{c}{q} - \frac{d}{r}\right| \leqslant \frac{1}{2q} \tag{21}$$

Because $q \quad n^2$, there is exactly one fraction d / r with $r < n$ that satisfies the above inequality. This fraction can be found efficiently using a continued fraction expansion of c/q. Hence, if $\gcd(d, r) = 1$, we get the value of r. In fact, there are $\varphi(r)$ such co-prime values of d, so we get

$$\mathrm{Prob}(\ d \text{ is prime to } r)_c \quad \frac{4\varphi(r)}{\pi^2 r} \tag{22}$$

So the success probability of Algorithm 1 is at least $4\varphi(r)/(\pi^2 r)$.

According to Theorem 5.3 in Ref. [16], we can conclude that

$$\mathrm{Prob}\left(r \text{ is even and } x^{r/2} \neq -1 (\bmod n)\right) \quad 1 \quad \frac{1}{2^m} \tag{23}$$

where m denotes the number of the factors of n.

So we can conclude that suppose $n=pq$, let x be an integer chosen uniformly at random from Z_n^* and r be the order of x modulo n. Then the probability of factoring integer n is greater than or equal to 3/4. And as the success probability of performing Shor's order finding algorithm is $4\varphi(r)/(\pi^2 r)$. Therefore, the success probability of Shor's algorithm for breaking RSA is

$$3\varphi(r) / (\pi^2 r) \leqslant \mathrm{Prob(Shor's)} < 4\varphi(r) / (\pi^2 r) \tag{24}$$

However, the success probability of Algorithm 1 is at least $4\varphi(r)/(\pi^2 r)$. Hence the success probability of Algorithm 1 is higher than that of Shor's algorithm for breaking RSA.

2 Conclusion and Future Work

In this paper, a quantum algorithm for computing the order r of the fixed-point C (the RSA ciphertext) of the given RSA public-key (e, $n=pq$) such that $C^{e^r} \equiv C (\bmod n)$ is presented. Since once r is obtained, the RSA plaintext M can be immediately computed by $M \equiv C^{e^{r-1}} (\bmod n)$, and hence, break the RSA completely. Compared to Shor's original order finding algorithm, the order in the new algorithm does not need to be even and the algorithm is easy to be implemented on a quantum computer. Of course, for the algorithm to be

practical, more research still needs to be done. One of our current research directions, along with this line, is to reduce the quantum bits used in the algorithm, so that it may be run on a smaller quantum computer that may be relatively easy to construct and build.

References

[1] Zhang H G, Han W B, Lai X J, *et al*. Survey on cyberspace security [J]. *Science China Information Sciences*, 2015, **58**(11): 1-43.

[2] Wang Y L, Xu Q L. Principle and research progress of quantum computation and quantum cryptography [J]. *Journal of Computer Research and Development*, 2020, **57**(10): 2015-2026.

[3] Shor P W. Algorithms for quantum computation: Discrete logarithms and factoring [C] //*Proceedings of* 35*th Annual Symposium on Foundations of Computer Science.* Washington D C: IEEE Computer Society Press, 1994: 124-134.

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

[5] Peng X H, Liao Z Y, Xu N Y, *et al*. Quantum adiabatic algorithm for factorization and its experimental implementation [J]. *Physical Review Letters*, 2008, **101**(22): 220405.

[6] Xu N Y, Zhu J, Lu D W, *et al.* Quantum factorization of 143 on a dipolar-coupling nuclear magnetic resonance system [J]. *Physical Review Letters*, 2012, **108**(13): 130501.

[7] Wang C, Yao H N, *et al*. Progress in quantum computing cryptography attacks [J]. *Chinese Journal of Computers*, 2020, **43**(9): 1691-1707(Ch).

[8] Geller M R, Zhou Z Y. Factoring 51 and 85 with 8 qubits [J]. *Scientific Reports*, 2013, **3**(3023): 1-5.

[9] Wang Y H, Zhang H G, Wu W Q, *et al*. Quantum algorithms for breaking RSA based on phase estimation and equation solving [J]. *Chinese Journal of Computers*, 2017, **40**(12): 2687- 2699(Ch).

[10] Wang Y H, Zhang H G, Wang H Z. Quantum polynomial-time fixed-point attack for RSA [J]. *China Communications*, 2018, **15**(2): 25-32.

[11] Wang Y H, Yan S Y, Zhang H G. A new quantum algorithm for computing RSA ciphertext period [J]. *Wuhan University Journal of Natural Sciences*, 2017, **22**(1): 68-72.

[12] Lawson T. Odd orders in Shor's factoring algorithm [J]. *Quantum Information Process*, 2015, **14**(3): 831-838.

[13] Dattani N S, Bryans N. Quantum factorization of 56153 with only 4 qubits. [EB/OL]. [2021-05-10]. *http://arxiv.org/pdf /*1411.6758, 27, 2014.

[14] Peng W C, Wang B N, Hu F, *et al*. Factoring larger integers with fewer qubits via quantum annealing with optimized parameters [J]. *Science China*: *Physics*, *Mechanics & Astronomy*, 2019, **62**(6): 5-12(Ch).

[15] Yan S Y. *Quantum Computational Number Theory* [M]. Berlin: Springer-Verlag, 2015.

[16] Nielson M A, Chuang I L. *Quantum Computation and Quantum Information* [M]. Cambridge: Cambridge University Press, 2000.

□