Quantum Polynomial-Time Fixed-Point Attack for RSA

Yahui Wang, Huanguo Zhang*, Houzhen Wang

School of computer, Wuhan University, Wuhan 430072, Hubei Province, China
Key Laboratory of Aerospace Information security and trusted computing Ministry of Education, Wuhan University, Wuhan 430072, China
* The corresponding author, email: liss@whu.edu.cn

Abstract: Security analysis of public-key cryptosystems is of fundamental significance for both theoretical research and applications in cryptography. In particular, the security of widely used public-key cryptosystems merits deep research to protect against new types of attacks. It is therefore highly meaningful to research cryptanalysis in the quantum computing environment. Shor proposed a well-known factoring algorithm by finding the prime factors of a number $n = pq$, which is exponentially faster than the best known classical algorithm. The idea behind Shor's quantum factoring algorithm is a straightforward programming consequence of the following proposition: to factor n, it suffices to find the order r; once such an r is found, one can compute $\gcd\left(a^{r/2} \pm 1, n\right) = p$ or q. For odd values of r it is assumed that the factors of n cannot be found (since $a^{r/2}$ is not generally an integer). That is, the order r must be even. This restriction can be removed, however, by working from another angle. Based on the quantum inverse Fourier transform and phase estimation, this paper presents a new polynomial-time quantum algorithm for breaking RSA, without explicitly factoring the modulus n. The probability of success of the new algorithm is greater than $4\varphi(r)/\pi^2 r$, exceeding that of the existing quantum algorithm for attacking RSA based on factorization. In constrast to the existing quantum algorithm for attacking RSA, the order r of the fixed point C for RSA does not need to be even. It changed the practices that cryptanalysts try to recover the private-key, directly from recovering the plaintext M to start, a ciphertext-only attack attacking RSA is proposed.

Keywords: information security; cryptography; RSA fixed-point; quantum computing

I. INTRODUCTION

Advances in quantum computation present a serious challenge to existing public-key cryptosystems: the RSA public-key cryptosystem can be attacked by Shor's algorithm. Researching cryptanalysis in the quantum computing environment is thus of great significance [1]. It is well known that the security of RSA essentially depends on only the computational intractability of the Integer Factorization Problem (IFP), and in particular, it is only secured if the IFP does not have an efficient algorithm. That is, anyone who can solve the IFP in polynomial-time can break the RSA cryptographic system in polynomial-time. The IFP has been studied since ancient times, and exponential-time algorithms for it have been developed, including Lehman's method, Shanks'SQU are FOrm Factorization method,

Based on the quantum inverse Fourier transform and phase estimation, this paper presents a new polynomial-time quantum algorithm for breaking RSA

Shanks' class group method, the continued fraction method, Pollard's ρ method [2]. With the invention of RSA public-key cryptography in 1978 [3], the problem became important and attracted a great deal of attention.

There are many methods for attacking RSA, such as the integer factorization attacks, the discrete logarithm attacks, the public exponent attacks, the private exponent attacks and side channel attacks [4][5]. The computers we are using at present are called classical computers. The most powerful method for breaking RSA is to use the Number Field Sieve (NFS) [6] to factor n, which runs in subexponential-time $\mathrm{O}\left(\exp\left(c\left(\log n\right)^{1/3}\right)\left(\log\log n\right)^{2/3}\right)$, where $c\approx 1.92$. In fact, all the existing factoring algorithms up to this point, such as the NFS and the ρ methods, are inefficient and cannot run in polynomial-time. The ineffectiveness of factoring makes it useful for constructing unbreakable cryptography. However, a polynomial-time quantum factoring algorithm, proposed by Shor in 1994 [7,8], can solve the IFP in a time proportional to $\mathrm{O}\left(\left(\log n\right)^{2+\varepsilon}\right)$. The idea of Shor's quantum factoring algorithm is a straightforward programming consequence of the following proposition: to factor n, it suffices to find r, which is the smallest integer, satisfying $a^r \equiv 1\left(\mathrm{mod}\, n\right)$, where $\gcd\left(a,n\right)=1$. The factors are then given by $p=\gcd\left(a^{r/2}+1,n\right)$ and $q=\gcd\left(a^{r/2}\text{-}1,n\right)$. It is best to implement the full version of Shor's algorithm to factor n. The most straightforward method for attacking RSA is to factor n. Shor showed that both integer factorization based cryptography and discrete logarithm based cryptography can be totally broken in polynomial-time on practical quantum computers. If a quantum computer with several thousand quantum bits can be built, many existing public-key cryptosystems such as RSA, ElGamal and ECC, will no longer be secure, threatening cyberspace security [1,9].

The emergence of Shor's algorithm has injected new vitality into research on quantum computation, and has led to an upsurge of quantum computation and quantum computer research over the last twenty years [9-14]. Tremendous efforts have been made to develop practical quantum computers and to improve Shor's algorithm using different techniques. In [15] a compiled version for factoring 15 using quantum entanglement was proposed, while [16] provided an experimental demonstration of a factoring method with a temporal Talbot effect for factoring the number 19403. For a more compiled version of Shor's algorithm, please refer to the references [17-22]. Recent research has sought to reduce the number of quantum bits and to make it easy to run on a quantum computer with fewer quantum bits. For example, [23] constructed simplified quantum circuits and gave an example for factoring the numbers 51 and 85 with 8 qubits, and [24] proposed a quantum computing idea to find the number a such that $a^2 \equiv 1\left(\mathrm{mod}\, n\right)$. For more information, please refer to the references [25,26].

It has nevertheless been known for a long time that there is no need to factor n if the only aim is to attack RSA. In fact, to recover M from C, it is enough to compute the order, r, of the fixed point C. Once the order r has been found, the plaintext M is simply the element $C^{e^{r-1}} \bmod n$. In classical computing, this computation is equivalent to factoring n, which is believed to be hard. In this paper, we present a new polynomial-time quantum algorithm that can be used to attack RSA without factoring the modulus n.

II. Preliminaries

We first present some basic concepts of the RSA problem, the RSA fixed-point and Quantum Fourier Transform(QFT); for more related information, please refer to the references [2,27].

Definition 1 (The RSA Problem) Given the RSA public-key (e,n) and the RSA ciphertext C, find the corresponding RSA plaintext M. That is,

$$\left\{e,n,C\equiv M^e\left(\mathrm{mod}\, n\right)\right\}\xrightarrow{\text{find}}\left\{M\equiv C^d\left(\mathrm{mod}\, n\right)\right\}$$

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

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

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 be the fixed-point of $RSA(e,n)$ with order r, such that

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

Then

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

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

Proof Please refer to the reference [2].

Definition 3 (Quantum Fourier Transform(QFT))The quantum Fourier Transform on an orthonormal basis $|0\rangle,|1\rangle,\cdots,|N-1\rangle$ is defined to be a linear operator with the following action on the basis states,

$$|j\rangle \to \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} e^{2\pi ijk/N} |k\rangle. \tag{4}$$

Equivalently, the action on an arbitrary state may be written as

$$\sum_{j=0}^{N-1} x_j |j\rangle \to \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} \sum_{j=0}^{N-1} x_j e^{2\pi ijk/N} |k\rangle. \tag{5}$$

In addition, for reference, we state the action of the quantum inverse Fourier transform (denoted QFT^{-1}) on the basis states $|0\rangle,|1\rangle,\cdots,|q-1\rangle$:

$$\mathrm{QFT}^{-1}: |k\rangle \to \frac{1}{\sqrt{q}} \sum_{j=0}^{q-1} e^{-2\pi ikj/q} |j\rangle. \tag{6}$$

Accordingly, the phase can be easily estimated using the quantum inverse Fourier transform. This is, of course, based on the basic assumptions of quantum mechanics [27].

III. THE NEW ALGORITHM

Shor discovered a polynomial-time quantum integer factorization algorithm in 1994. For a given number n, he provided an algorithm for finding the order of the element a to get the factors of n, and further, to attack RSA. If the integer factorization problem is solved, RSA can be broken; however, attacking RSA does not require factoring n. Therefore, in this section, using the fixed point property of RSA and based on the quantum inverse Fourier transform and phase estimation, we present a new polynomial-time quantum algorithm for directly recovering the RSA plaintext M from the ciphertext C, without explicitly factoring the modulus n. The specific steps of the algorithm steps are as follows.

3.1 Algorithm design

We present a quantum algorithm for computing the order of the fixed point C, enabling breaking RSA in polynomial-time without factoring.

Algorithm 1 Given the RSA public-key (e,n) and RSA ciphertext C, this algorithm tries to find the order r of the fixed-point C such that $C^{e^r} \equiv C \pmod{n}$, based on the quantum inverse Fourier transform and phase estimation. Once such an r is found, $C^{e^{r-1}} \equiv M \pmod{n}$.

Input: C,e,n

Output: M

Step 1. Find a number $q = 2^k$, where $k = \lceil \log n \rceil$.

Step 2. Give two k dimensional quantum registers whose initial state are $|\Psi_0\rangle = |0^k\rangle|C^k\rangle$.

Step 3. Perform a Hadamard transform on the first register, yielding

$$\mathrm{H}: |\psi_0\rangle \to |\psi_1\rangle = \frac{1}{\sqrt{q}} \sum_{x=0}^{q-1} |x\rangle|C\rangle. \tag{7}$$

Step 4. Perform the unitary transform U_C^x on the second register, giving

$$\mathrm{U}_C^x: |\psi_1\rangle \to |\psi_2\rangle = \frac{1}{\sqrt{q}} \sum_{x=0}^{q-1} |x\rangle \mathrm{U}_C^x |C\rangle \tag{8}$$

$$= \frac{1}{\sqrt{q}} \sum_{x=0}^{q-1} |x\rangle \left| C^{e^x} \pmod{n} \right\rangle, \tag{9}$$

$$= \frac{1}{\sqrt{qr}} \sum_{x=0}^{q-1} \sum_{s=0}^{r-1} |x\rangle e^{\frac{2\pi isx}{r}} |\Phi_s\rangle, \tag{10}$$

where r is the order of the fixed point C.

Step 5. Perform QFT^{-1} on the first register, giving

$$\mathrm{QFT}^{-1}: |\Psi_2\rangle \to |\Psi_3\rangle$$

$$= \frac{1}{q} \sum_{x=0,c=0}^{q-1} \sum_{s=0}^{r-1} \frac{1}{\sqrt{r}} e^{\frac{-2\pi icx}{q}} e^{\frac{2\pi isx}{r}} |c\rangle|\Phi_s\rangle$$

$$= \frac{1}{q\sqrt{r}} \sum_{s=0}^{r-1} \sum_{x=0,c=0}^{q-1} e^{2\pi i\left(\frac{s}{r} - \frac{c}{q}\right)x} |c\rangle|\Phi_s\rangle. \tag{11}$$

Suppose $\frac{s}{r}=\phi_s$, then

$$\left|\Psi_3\right\rangle=\frac{1}{q\sqrt{r}}\sum_{s=0}^{r-1}\sum_{x=0,c=0}^{q-1}e^{2\pi i\left(\phi_s-\frac{c}{q}\right)x}\left|c\right\rangle\left|\Phi_s\right\rangle. \quad (12)$$

Step 6. Measure the first register; suppose we observe the state $\left|c_1\right\rangle$, using the continued fractions algorithm to get r_1 satisfying $\left|c_1/q-s_1/r_1\right|\le 1/\sqrt{q/2}$.

Step 7. Repeat Steps 1-5; suppose we observe the state $\left|c_2\right\rangle$ using the continued fractions algorithm to get r_2 satisfying $\left|c_2/q-s_2/r_2\right|\le 1/\sqrt{q/2}$.

Step 8. Compute $r=\mathrm{LCM}\left(r_1,r_2\right)$, where $\mathrm{LCM}\left(r_1,r_2\right)$ is the least common multiple of r_1 and r_2. Compute $C^{e^r}\left(\mathrm{mod}\ n\right)$.

Step 9. If $C^{e^r}\equiv C\left(\mathrm{mod}\ n\right)$, we get the order r of the fixed-point C, and then output r.

Step 10. Compute $M\equiv C^{e^{r-1}}\left(\mathrm{mod}\ n\right),$. The required plaintext M is thus obtained; that is, RSA is attacked.

As we can see, Algorithm 1 breaks RSA without factoring n and without using any knowledge of the trap-door information $\left\{d,p,q,\varphi(n)\right\}$, which only recovers the plaintext M from the ciphertext C. Moreover, obtaining the ciphertext C is also the most easily satisfied condition in the real break, so the attack for Algorithm 1 belongs to the category of ciphertext-only attacks. It changes the practices by which cryptanalysts try to recover the private-key, directly from recovering the plaintext M to start, Algorithm 1 gives a ciphertext-only attacks of RSA.

Algorithm 1 breaks RSA from the angle of non-factorization. That is, attacking RSA in Algorithm 1 does not pass through factorization from the traditional mathematical method of RSA itself, whereas traditional methods for breaking RSA all pass through factorization.

3.2 Algorithm analysis

First, we give a complexity analysis of Algorithm 1 as follows.

The dominant computation of the algorithm is that of the inverse quantum Fourier transform, which takes time proportional to $\mathrm{O}\left(c\left(\log n\right)^2\log\log n\log\log\log n\right)$, whereas the gcd computation takes time $\mathrm{O}\left(\left(\log n\right)^2\right)$. Therefore, in total, the computation time of the algorithm is proportional to

$$\mathrm{O}\left(c\left(\log n\right)^2\log\log n\log\log\log n\right)=\mathrm{O}\left(\left(\log n\right)^{2+\varepsilon}\right)$$

That is, Algorithm 1 recovers M from C in quantum polynomial-time $\mathrm{O}\left(\left(\log n\right)^{2+\varepsilon}\right)$.

We next analyze the correctness of Algorithm 1.

From the literature [27], we know that the Hadamard transform we use in Step 3 of Algorithm 1 is a unitary transform. The transform U_C^x used in Step 4 is constructed as follows:

For a given positive integer C, which is prime to n, there exists a unitary transform $\mathrm{U}_C\left|y\right\rangle=\left|y^e\mathrm{mod}\ n\right\rangle$, and the unitary transform U_C can be performed efficiently. Thus, $\mathrm{U}_C^x\left|y\right\rangle=\left|y^{e^x}\mathrm{mod}\ n\right\rangle$.

Consider the following state:

$$\left|\Phi_s\right\rangle=\frac{1}{\sqrt{r}}\sum_{t=0}^{r-1}e^{\frac{-2\pi ist}{r}}\left|C^{e^t}\left(\mathrm{mod}\ n\right)\right\rangle, \quad (13)$$

where s is a positive integer satisfying $0\le s\le r-1$.

Although the state $\left|\Phi_s\right\rangle$ may be difficult to prepare, the state $\left|C\right\rangle$ is easy to prepare.

Consider

$$\frac{1}{\sqrt{r}}\sum_{s=0}^{r-1}\left|\Phi_s\right\rangle=\frac{1}{\sqrt{r}}\sum_{s=0}^{r-1}\frac{1}{\sqrt{r}}\sum_{t=0}^{r-1}e^{\frac{-2\pi ist}{r}}\left|C^{e^t}\left(\mathrm{mod}\ n\right)\right\rangle. \quad (14)$$

Notice that because r is the order of the fixed point C, $C^{e^r}\equiv C\ \mathrm{mod}\ n.$ The amplitude of the state $\left|C\right\rangle$ in the above state is then the sum over the terms for which $t=0$. That is,

$$\frac{1}{\sqrt{r}}\frac{1}{\sqrt{r}}\sum_{s=0}^{r-1}e^{\frac{-2\pi is_0}{r}}=\frac{1}{r}\sum_{s=0}^{r-1}1=1. \quad (15)$$

Thus the amplitude of the state $\left|C\right\rangle$ is 1, and consequently the amplitude of all other basis states must be 0. Therefore, we have

$$\frac{1}{\sqrt{r}}\sum_{s=0}^{r-1}\left|\Phi_s\right\rangle=\left|C\right\rangle. \quad (16)$$

Performing the unitary transform U_C on the quantum state $\left|\Phi_s\right\rangle$, we get

$$\mathrm{U}_C\left|\Phi_s\right\rangle=\frac{1}{\sqrt{r}}\sum_{t=0}^{r-1}e^{\frac{-2\pi ist}{r}}\mathrm{U}_C\left|C^{e^t}\pmod n\right\rangle$$
$$=\frac{1}{\sqrt{r}}\sum_{t=0}^{r-1}e^{\frac{-2\pi ist}{r}}\left|C^{e^{t+1}}\pmod n\right\rangle$$
$$=e^{\frac{2\pi is}{r}}\frac{1}{\sqrt{r}}\sum_{t=0}^{r-1}e^{\frac{-2\pi is(t+1)}{r}}\left|C^{e^{t+1}}\pmod n\right\rangle$$
$$=e^{\frac{2\pi is}{r}}\left|\Phi_s\right\rangle. \tag{17}$$

Thus, $\left|\Phi_s\right\rangle$ is an eigenstate of U_C with eigenvalue $e^{\frac{2\pi is}{r}}$.

Performing the unitary transform U_C^x on the quantum state $\left|\Phi_s\right\rangle$, we get

$$\mathrm{U}_C^x\left|\Phi_s\right\rangle=e^{\frac{2\pi isx}{r}}\left|\Phi_s\right\rangle. \tag{18}$$

Then substituting (16) and (18) into $\left|\psi_2\right\rangle=\frac{1}{\sqrt{q}}\sum_{x=0}^{q-1}\left|x\right\rangle\mathrm{U}_C^x\left|C\right\rangle$ of (8), we get

$$\left|\Psi_2\right\rangle=\frac{1}{\sqrt{qr}}\sum_{x=0}^{q-1}\sum_{s=0}^{r-1}\left|x\right\rangle e^{\frac{2\pi isx}{r}}\left|\Phi_s\right\rangle.$$

That is (10).

From [27], we known that the quantum inverse Fourier transform is a unitary transform. The transforms we use in Algorithm 1 satisfy the reversible conditions required by the quantum computing algorithm. Thus in terms of the transforms used in Algorithm 1, the algorithm is correct. Figure 1 presents a circuit for implementing the Algorithm 1.

Finally, we analyze the success probability of Algorithm 1 as follows.

By (12) in Step 5, if $r\,|\,q$,

$$\sum_{x=0}^{q-1}e^{2\pi i\left(\phi_s-\frac{c}{q}\right)x}:=\begin{cases}q\text{: }\phi_s=c/q,\\0\text{: }\phi_s\neq c/q.\end{cases} \tag{19}$$

Accordingly, the amplitude of the quantum state $\left|c\right\rangle$ is zero, and does not satisfy $\phi_s=c/q$; that is, after Step 5, the quantum states $\left|c\right\rangle$ leaving in the first register satisfy $\phi_s=c/q$. By Step 5, the probability $\mathrm{Prob}(c)$ that the machine reaches the state $\left|c\right\rangle$ is

$$\mathrm{Prob}(c)=\frac{1}{rq^2}\left|\sum_{x=0}^{q-1}e^{2\pi i\left(\phi_s-\frac{c}{q}\right)x}\right|^2, \tag{20}$$

where $\phi_s=c/q$, so at this time each state of the quantum superposition that we require is observed with the probability

$$\mathrm{Prob}(c)=\frac{1}{rq^2}\left|\sum_{x=0}^{q-1}e^{2\pi i\left(\phi_s-\frac{c}{q}\right)x}\right|^2:=\begin{cases}1/r\text{: }\phi_s=c/q,\\0\text{: }\phi_s\neq c/q.\end{cases} \tag{21}$$

The observed state $\left|c\right\rangle$ has r possible values in Step 6, but the state $\left|c\right\rangle$ satisfying $\gcd(c,r)=1$ only has $\varphi(r)$ possible values. Thus the probability of outputting the correct state $\left|c\right\rangle$ in Step 6 is $P=\mathrm{Prob}(c)\times\varphi(r)=\frac{\varphi(r)}{r}$. That is, when $r\,|\,q$, the probability of Algorithm 1 is $\frac{\varphi(r)}{r}$, where φ is the Euler function.

If $r\nmid q$,we can see from Figure 2 that c satisfying

$$\left|\frac{s}{r}-\frac{c}{q}\right|\leq\frac{1}{2q} \tag{22}$$

must exist in $\{0,1,2,\cdots,q-1\}$.

When satisfying (22), it is easy to learn that the phase is concentrated in the upper or lower part of the complex plane, while at the same time, the summation of x can lead to the superposition of phases. If it does not satisfy (22) at this point, the summation of x can lead to the offset of phases by each other, with its size being almost negligible. Therefore, the probability of observing the state $\left|c\right\rangle$ is

$$\mathrm{Prob}(c)=\frac{1}{rq^2}\left|\sum_{x=0}^{q-1}e^{2\pi i\left(\phi_s-\frac{c}{q}\right)x}\right|^2 \tag{23}$$
$$\approx\begin{cases}1/r\text{: }|s/r-c/q|\leq 1/2q,\\0\text{: }\quad\text{otherwise.}\end{cases} \tag{24}$$

Thus, the probability of the observed state

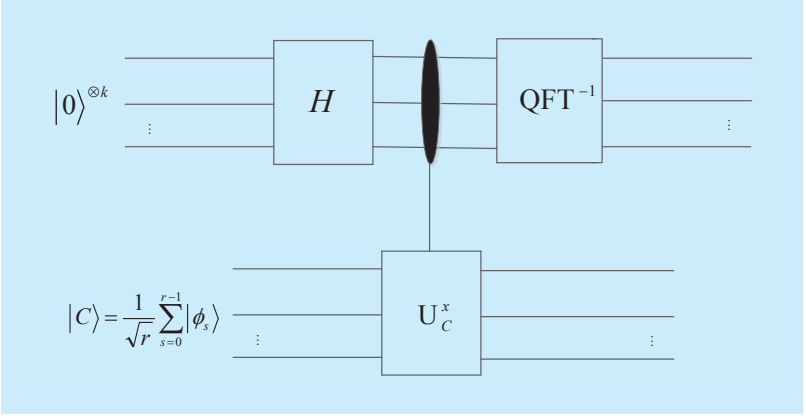

Fig. 1. *Circuit for implementing the quantum part of Algorithm 1.*

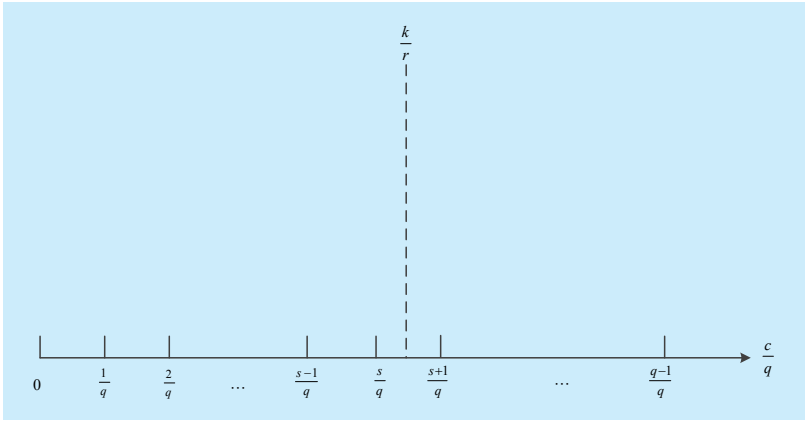

Fig. 2. *The search for r in the case $r \nmid q$.*

Table I. *Comparison of resource consumption.*

Algorithm for attacking RSA	Success probability	Time complexity	Qubits	Theoretical basis	Type of attack
[8,9]	P_{Shor}	$O\left((\log n)^3\right)$	$3\lceil \log n \rceil$	Factorization	Integer factorization attack
[17]	$\varphi(r)/3r$	$O\left((\log n)^3\right)$	$4\lceil \log n \rceil$	Factorization	Integer factorization attack
[28]	P_{Shor}	$O\left((\log n)^3\right)$	$3\lceil \log n \rceil$	Factorization	Integer factorization attack
Algorithm 1	$\approx \frac{\varphi(r)}{r}$	$O\left((\log n)^3\right)$	$3\lceil \log n \rceil$	Non-factorization	RSA fixed-point attack

where $3\varphi(r)/\pi^2 r \le P_{Shor} < 4\varphi(r)/\pi^2 r$.

$|c\rangle$ is approximated to $\frac{1}{r}$ in case of $r \nmid q$. Similarly, the state $|c\rangle$ satisfying $\gcd(c,r)=1$ only has $\varphi(r)$ possible values. Thus the probability of outputting the correct state $|c\rangle$ in Step 6 is

$$P \approx \text{Prob}(c) \times \varphi(r) = \frac{\varphi(r)}{r}, \tag{25}$$

that is, when $r \nmid q$, the probability of success of Algorithm 1 is $\approx \frac{\varphi(r)}{r}$.

Accordingly, the probability of success of Algorithm 1 depends on the order of the fixed point C. In particular, when r is a prime number, the probability of Algorithm 1 is similar to $(r-1)/r$; that is, as r increases, the probability of success approaches 1. Further, we show that the probability of success of Algorithm 1 is higher than that of Shor's algorithm. In fact, the probability of success of Shor's algorithm for breaking RSA is $3\varphi(r)/\pi^2 r \le P_{\text{Shor}} < 4\varphi(r)/\pi^2 r$. From the above analysis, we know that for Algorithm 1, when $r \mid q$, the probability of success of Algorithm 1 is $\frac{\varphi(r)}{r}$, and when $r \nmid q$, the probability of success of Algorithm 1 is $\approx \frac{\varphi(r)}{r}$. Because $\frac{4\varphi(r)}{\pi^2 r} < \frac{\varphi(r)}{r}$, the probability of success of Algorithm 1 is higher than that of Shor's algorithm.

We now more clearly distinguish the features of Algorithm 1 from those of its predecessor in attacking RSA. Table 1 compares time complexity, the probability of success, required quantum qubits, theoretical basis, and type of attack for both algorithms .

In Table 1, we can see that Algorithm 1 has the following properties: 1) it recovers the RSA plaintext M from the ciphertext C without factoring n; 2) it does not require the order of the element to be even; 3) it has higher probability of success; 4) it is the first quantization algorithm for an RSA fixed-point attack based on the current literature.

IV. Conclusion

Because the RSA cryptosystem is widely used in industry and government, quickly cracking RSA has become an important research direction for modern cryptanalysis. The essential trick to attacking the RSA public-key cryptosystem is a method for factoring modulus n efficiently. If the only goal, however, is to break RSA, we can compute the order r of the fixed-point C directly without factoring. This paper thus presents a new quantum algorithm for finding the order r of the fixed-point C of the given RSA public-key $(e, n=pq)$, such that $C^{e^r} \equiv C \pmod n$, based on the QFT^{-1} and phase estimation. Because once r is found, the RSA plaintext M can be immediately obtained by computing $M \equiv C^{e^{r-1}} \pmod n$, RSA can be attacked without factoring the modulus n. The probability of success of Algorithm 1 is higher than that

of Shor's algorithm, and the algorithm runs in polynomial time. Moreover, our algorithm for breaking RSA does not require randomly choosing an integer x such that the order of x modulo n is even; it only needs to find the order of the fixed-point C, regardless of its parity.

Until now, only Shor's algorithm has been suitable for quickly solving some periodic problems, such as the integer factorization problem, discrete logarithm problem, elliptic curve discrete logarithm problem, and Pell equation. However, the question of whether Shor's algorithm can also solve non-periodic problems is a meaningful research direction, which remains a problem for further study.

Acknowledgement

This work is partially supported by he State Key Program of National Natural Science of China No. 61332019, Major State Basic Research Development Program of China (973 Program) No. 2014CB340601, the National Science Foundation of China No. 61202386, 61402339, the National Cryptography Development Fund No. MMJJ201701304.

References

[1] H.G Zhang, W.B Han, X.J Lai, et al., "Survey on cyberspace security", *SCIENCE CHINA: Information Science*, vol. 58, no. 11, 2015, pp. 1-43.

[2] S.Y Yan, "Quantum computational number theory", *Berlin: Springer*, 2015.

[3] R.L Rivest, A. Shamir, L. Adleman, "A method for obtaining digital signatures and public key cryptosystems", *Communications of the ACM*, vol. 21, no. 2, 1978, pp. 120-126.

[4] F. Jia, D. Xi, "A unified method based on SPA and timing attacks on the improved RSA", *China Communications*, vol. 13, no. 4, 2016, pp. 89-96.

[5] P. Zhou, T. Wang, G. Li, F. Zhang, X.J Zhao, "Analysis on the parameter selection method for FLUSH+RELOAD based cache timing attack on RSA", *China Communications*, vol. 12, no. 6, 2015, pp. 33-45.

[6] A.K Lenstra, H.W Lenstra, M.S Manasse, J.M Pollard, "The number field sieve", *Lecture Notes in Mathematics, Berlin: Springer*, vol. 1554, 1993, pp. 11-42.

[7] P.W Shor, "Algorithms for quantum computation: discrete logarithms and factoring", *Proc. 35th Annual Symposium on Foundations of Computer Science*, 1994, pp. 124-134.

[8] P.W Shor, "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer", *SIAM Journal on Computing*, vol. 26, no. 5, 1997, pp. 1484-1509.

[9] W.Q Wu, H.G Zhang, H.Z Wang, et al., "A public key cryptosystem based on data complexity under quantum environment", *SCIENCE CHINA: Information Science*, vol. 58, no. 11, 2015, pp. 1-11.

[10] H.F Wang, "Quantum algorithm for obtaining the eigenstates of a physical system", *Physical Review A*, vol. 93, 2016, pp. 052334.

[11] S.W Mao, H.G Zhang, W.Q Wu, et al. "Key exchange protocol based on tensor decomposition problem", *China Communications*, vol. 13, no. 3, 2016, pp. 174-183.

[12] W.Q Wu, H.G Zhang, H.Z Wang, S.W Mao, "Polynomial-time quantum algorithms for finding the linear structures of boolean function", *Quantum Information Process*, vol. 14, no. 4, 2015, pp. 1215-1226.

[13] H.G Zhang, S.W Mao, W.Q Wu, et al., "Overview of quantum computation complexity theory", *Chinese Journal of Computers*, vol. 39, no. 12, 2016, pp. 2403-2428.

[14] S. Wang, X.H Song, X.M Niu, "Quantum cosine transform based watermarking scheme for quantum images", *Chinese Journal of Electronics*, vol. 24, no. 2, 2015, pp. 321-325.

[15] B.P Lanyon, T.J Weinhold, N.K Langford, "Experimental demonstration of a compiled version of Shor's algorithm with quantum entanglement", *Physical Review Letters*, vol. 99, no. 25, 2007, pp. 250505.

[16] M.R Geller, Z.Y Zhou, "Factoring 51 and 85 with 8 qubits", *Scientific Reports*, vol. 3, no. 3023, 2013, pp. 1-5.

[17] Z.J Cao, Z.F Cao, "On Shor's factoring algorithm with more registers and the problem to certify quantum computers", *arXiv:1409.7352v1 [cs.DS] 10 Sep 2014*.

[18] C. Lu, D. Browne, T. Yang, et al., "Demonstration of a compiled version of Shor's quantum algorithm using photonic qubits", *Physical Review Letters*, vol. 99, no. 25, 2007, pp. 250504.

[19] E. Lucero, R. Barends, Y. Chen, et al., "Computing prime factors with a Josephson phase qubit quantum processor", *Nature Physics*, vol. 8, no. 10, 2012, pp. 719-723.

[20] X.H Peng, Z.Y Liao, N.Y Xu, et al., "Quantum adiabatic algorithm for factorization and its experimental implementation", *Physical Review Letters*, vol. 101, no. 22, 2008, pp. 220405.

[21] N.Y Xu, J. Zhu, D.W Lu, et al., "Quantum factorization of 143 on a dipolar-coupling nuclear magnetic resonance system", *Physical Review Letters,* vol.108, no.13, 2012, pp. 130501.

[22] T. Lawson, "Odd orders in Shor's factoring algorithm", *Quantum Information Process,* vol. 14,

no. 3, 2015, pp. 831-838.
[23] D. Bigourd, B. Chatel, W.P Schleich, et al., "Factorization of numbers with the Temporal Talbot effect: optical implementation by a sequence of shaped Ultrashort pulse", *Physical Review Letters*, vol. 100, no. 3, 2008, pp. 030202.
[24] J.A Smolin, G. Smith, A. Vargo, "Oversimplifying quantum factoring", *Nature*, vol. 499, 2013, pp. 163-165.
[25] N.S Dattani, N. Bryans, "Quantum factorization of 56153 with only 4 qubits", *http://arxiv.org/pdf/1411.6758, 27 Nov 2014,* 6 pages, 2014.
[26] E. Martin-Lopez, A. Laing, T. Lawson, et al., "Experimental realization of Shor's quantum factoring algorithm using qubit recycling", *Nature Photonics,* vol. 6, no. 11, 2012, pp. 773-776.
[27] M.A Nielsen and I.L Chuang, "Quantum computation and quantum information", *10th anniversary edition, Cambridge: Cambridge University Press*, 2010.
[28] L.H Liu and Z.J Cao, "On computing $ord_n(2)$ and its application", *Information and Computation*, vol. 204, no. 7, 2006, pp. 1173-1178.

Biographies

Yahui Wang, is a Ph.D. candidate at school of computer, Wuhan University, China. She received the M.S.degree in school of mathematics and statistics from Wuhan University. Her research interests include quantum computing and cryptography. Email: wangyh_ecc@whu.edu.cn.

Huanguo Zhang, is a professor at school of computer, Wuhan University, China. He received the Ph.D.degree in Xiandian University. His research interests include information security, cryptography, trusted computing, cloud computing, fault tolerance, and computer application. *The corresponding author, email: liss@whu.edu.cn

Houzhen Wang, is a lecture at school of computer, Wuhan University, China. He received the Ph.D.degree in Wuhan University. His research interests include information security, cryptography.