Quantum Factoring Algorithm: Resource Estimation and Survey of Experiments

Noboru Kunihiro

Abstract It is known that Shor's algorithm can break many cryptosystems such as RSA encryption, provided that large-scale quantum computers are realized. Thus far, several experiments for the factorization of the small composites such as 15 and 21 have been conducted using small-scale quantum computers. In this study, we investigate the details of quantum circuits used in several factoring experiments. We then indicate that some of the circuits have been constructed under the condition that the order of an element modulo a target composite is known in advance. Because the order must be unknown in the experiments, they are inappropriate for designing the quantum circuit of Shor's factoring algorithm. We also indicate that the circuits used in the other experiments are constructed by relying considerably on the target composite number to be factorized.

Keywords RSA · Quantum computer · Shor's quantum factoring algorithm · Oversimplified Shor's algorithm · Physical experiment

1 Introduction

It is crucial to evaluate the security of cryptosystems in order to securely use cryptographic technology. The security of RSA cryptosystems (Rivest et al. 1977), which are currently used widely, is based on the difficulty of factoring problem, and the evaluating the difficulty of the factoring problem is essential. Based on the security analysis, a 2048-bit composite number is widely used as a standard at present. It is known that prime factorization is possible in quantum polynomial time on the

N. Kunihiro (✉)

Faculty of Engineering, Information and Systems, University of Tsukuba, 1-1-1 Tennodai,
Tsukuba, Ibaraki 305-8573, Japan
e-mail: kunihiro@cs.tsukuba.ac.jp

© The Author(s) 2021

T. Takagi et al. (eds.), *International Symposium on Mathematics, Quantum Theory, and Cryptography*, Mathematics for Industry 33,
https://doi.org/10.1007/978-981-15-5191-8_7

bit length of the composite number using the Shor's algorithm (Shor 1997). Hence, almost all the currently used public-key cryptosystems will be broken if large-scale quantum computers are realized. Therefore, to prepare for the realization of quantum computers, quantum-resistant cryptography is researched actively at present (NIST 2020).

From the theoretical viewpoint, it has been evaluated how much resources are needed for the prime factorization of composite number of the currently used sizes (1024-bit, 2048-bit) (Häner 2017; Kunihiro 2005). However, from the experimental viewpoint, several experiments have been performed for the prime factorization of small composite numbers such as 15 and 21 (Lucero et al. 2012; Martin-Lopez et al. 2012; Monz et al. 2016; Politi 2009; Vandersypen 2001). In addition, commercial services for small-scale quantum computers such as IBM Q (2020) are beginning to be launched, and it is expected that the Noisy Intermediate-Scale Quantum (NISQ) technology might be available in the near future (Preskill 2018).

This paper presents a detailed survey of actual quantum experiments for prime factorization based on Shor's algorithm (Lucero et al. 2012; Martin-Lopez et al. 2012; Monz et al. 2016; Politi 2009; Vandersypen 2001). We give a detailed explanation of the circuits used in the experiments. We also indicate that some of them are problematic because they use a secret information in the circuit construction.

2 Outline of Shor's Quantum Factoring Algorithm (Shor 1997)

2.1 Quantum Computation

This subsection provides the basic facts about quantum gates (Nielsen and Chuang 2000). For the other information about quantum gates and circuits, refer to Nielsen and Chuang (2000).

We first explain a quantum bit, or *qubit*. A qubit has two possible states $|0\rangle$ and $|1\rangle$. We represent a single-qubit state as $\alpha |0\rangle + \beta |1\rangle$, where $\alpha, \beta \in \mathbb{C}$ and $|\alpha|^2 + |\beta|^2 = 1$. The gate that maps this state into $\alpha |1\rangle + \beta |0\rangle$ is called the NOT gate. The following matrix form is convenient for representing the NOT gate. Let a matrix X be

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}.$$

Suppose that the quantum state $\alpha |0\rangle + \beta |1\rangle$ is written in the vector form as

$$\begin{pmatrix} \alpha \\ \beta \end{pmatrix},$$

where the first entry corresponds to the amplitude for $|0\rangle$ and the second entry to the amplitude for $|1\rangle$. The corresponding output from the NOT gate is given by

$$X \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \begin{pmatrix} \beta \\ \alpha \end{pmatrix}.$$

The quantum gates on a single qubit can be described, in general, using 2×2 matrices. Furthermore, the matrix must be unitary. In fact, $X^\dagger X = I$ should hold, where $X^\dagger$ denotes the adjoint of X and I an identity matrix.

We then show the other important single-qubit gates, namely, the Z and H gates, in addition to the NOT gate. The matrix forms for the Z and H gates are given as follows.

$$Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, \quad H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

The H gate is usually referred to as the Hadamard gate. The Hadamard gate turns the state $|0\rangle$ into $(|0\rangle + |1\rangle)/\sqrt{2}$ and the state $|1\rangle$ into $(|0\rangle - |1\rangle)/\sqrt{2}$ because

$$H \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 1/\sqrt{2} \\ 1/\sqrt{2} \end{pmatrix} \quad \text{and} \quad H \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 1/\sqrt{2} \\ -1/\sqrt{2} \end{pmatrix}.$$

Furthermore, employing the Hadamard gate, we can construct the flat superposition from the state $|0\rangle$.

We now discuss multiple-qubit gates. The first gate is the Controlled-NOT (C-NOT) gate, which has two input qubits. The action of the C-NOT gate can be described as

$$|0\rangle|0\rangle \rightarrow |0\rangle|0\rangle, |0\rangle|1\rangle \rightarrow |0\rangle|1\rangle, |1\rangle|0\rangle \rightarrow |1\rangle|1\rangle, \text{ and } |1\rangle|1\rangle \rightarrow |1\rangle|0\rangle.$$

Equivalently, we can describe the action as

$$|a\rangle|b\rangle \rightarrow |a\rangle|b \oplus a\rangle,$$

where $\oplus$ denotes the exclusive OR.

The second one is the Toffoli gate, which has three input qubits. The action of the Toffoli gate can be described as

$$|a\rangle|b\rangle|c\rangle \rightarrow |a\rangle|b\rangle|c \oplus (a \wedge b)\rangle,$$

where $\wedge$ denotes the logical operator AND. The first two qubits are the *control* qubits and the third one is the *target* qubit.

We can consider the *generalized* version of the Toffoli gate as follows.

$$|c_1\rangle|c_2\rangle \cdots |c_n\rangle|t\rangle \rightarrow |c_1\rangle|c_2\rangle \cdots |c_n\rangle|t \oplus (c_1 \wedge c_2 \wedge \cdots \wedge c_n)\rangle.$$

In this case, the first n qubits are the *control* qubits, and the last qubit is the target qubit. It is well known that the generalized Toffoli gate can be decomposed into several Toffoli gates (Nielsen and Chuang 2000).

We then explain the controlled circuit. We denote a unitary operation by U . The action of the control- U circuit (C- U circuit) is described as

$$|0\rangle |x\rangle \rightarrow |0\rangle |x\rangle, \quad |1\rangle |x\rangle \rightarrow |1\rangle U|x\rangle.$$

Or, equivalently, the action can be described as

$$|c\rangle |x\rangle \rightarrow |c\rangle U^c |x\rangle.$$

We explain the Quantum Fourier Transformation (QFT). The QFT on a basis $|0\rangle, |1\rangle, \dots, |N-1\rangle$ is defined to be a linear operation with the following action on the states:

$$|j\rangle \rightarrow \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} \exp\left(\frac{2\pi i j k}{N}\right) |k\rangle.$$

The circuit for the QFT is constructed with the Hadamard gates and the controlled rotation gates. For the details, see the Sect.5 in Nielsen and Chuang (2000). The inverse QFT is defined to be the *inverse* operation of QFT.

2.2 Shor's Quantum Factoring Algorithm

Let N denote a target composite to be factored, and n denote a bit length of N . To simplify the discussion, hereafter, we assume that p and q are distinct prime integers and that N is the product of p and q . Let a denote a positive integer coprime to N . The final goal of Shor's algorithm is to find the prime factors p and q . However, before doing so, the algorithm will find a positive integer r such that $a^r \bmod N = 1$ as a subgoal. This positive integer r is called an order. If we know the order r , we can easily find the prime factors p and q of N with high probability.

We will now explain Shor's factoring algorithm in detail. Letting $m = 2n$, we first prepare the initialized state as follows:

$$\underbrace{|0\rangle}_{m\text{-qubit}} \underbrace{|1\rangle}_{n\text{-qubit}},$$

where the first register (referred to as the control register in Martin-Lopez et al. 2012 or the period register in Monz et al. 2016) is of m qubits and the second register (referred to as the work register in Martin-Lopez et al. 2012 or the computational register in Monz et al. 2016) is of n qubits. We may use ancilla in the calculation if required. Applying the Hadamard gate to the first register, we obtain the flat super-

Fig. 1 Shor's quantum factoring algorithm for the case of $m = 4$

position as follows:

$$\frac{1}{2^{m/2}} \sum_{x=0}^{2^m-1} \underbrace{|x\rangle}_{m\text{-qubit}} \underbrace{|1\rangle}_{n\text{-qubit}}.$$

Subsequently, we apply the modular exponentiation to this superposition to obtain the following state:

$$\frac{1}{2^{m/2}} \sum_{x=0}^{2^m-1} \underbrace{|x\rangle}_{m\text{-qubit}} \underbrace{|a^x \bmod N\rangle}_{n\text{-qubit}}.$$

We then apply the inverse of the Quantum Fourier Transformation to this state. At the last step, we obtain some value by measuring the first register. Using the measured value, we calculate the order r with the help of the continued fraction algorithm and then we find the prime factors of N by classical computers.

Here, the modular exponentiation is operated by sequentially applying $C-U_a$, $C-U_{a^2}$, $C-U_{a^4}$, $C-U_{a^{2^j}}$, and $C-U_{a^{2^{m-1}}}$ circuits, as shown in Fig. 1. Note that the action of the U_b operator is described as $|x\rangle \rightarrow |bx \bmod N\rangle$.

Suppose that we can find the order r of a modulo N . For simplicity, let us assume r to be even. By computing $\gcd(a^{r/2} - 1 \bmod N, N)$, we can find the prime factors of N with high probability.

Hereafter, we do not discuss the part of the Hadamard transformation and the part of the inverse of Quantum Fourier Transformation because the circuit complexity of both these parts can be ignored compared with that of the modular exponentiation part. Hereafter, we focus on the discussion of the resources necessary for modular exponentiation.

Table 1 Number of qubits and elementary gates (Kunihiro 2005)

	The number of qubits	The number of gates
R-ADD	$3n + 2$	$270n^3 + O(n^2)$
GT-ADD	$2n + 4 \rightarrow 2n + 2$ (Häner 2017)	$\frac{16}{3}n^5 + O(n^4)$
Q-ADD	$2n + 3 \rightarrow 2n + 2$ (Takahashi and Kunihiro 2006)	$97n^4 + O(n^4)$

2.3 Circuit Construction and Resource Estimation for Shor's Quantum Factoring Algorithm

The modular exponentiation can be executed by performing $O(n^3)$ gate operations for the standard construction of circuit. Kunihiro gave three construction types for modular exponentiation (Kunihiro 2005). These constructions adopt different types of addition circuits. In Kunihiro (2005), the number of qubits and the number of gates for Shor's factoring circuit were evaluated precisely. It was also shown that $3n + 2$ qubits and $270n^3 + O(n^2)$ Toffoli gates are required for modular exponentiation if the addition circuit similar to the classical addition is adapted. This result implies that we require 6146 qubits and 3.04×10^{12} Toffoli gates for factoring a 2048-bit composite. Table 1 presents the resource estimation of n -bit composite for quantum factoring. Table 2 shows those of 768-bit composite and 2048-bit composite. Note that the current world record for factoring is 768-bit composite (Kleinjung 2010) and the current recommendation of RSA composite is with 2048-bit.

In addition to the classical addition-based circuits (referred to as R-ADD in Table 1), (Kunihiro 2005) also gave a resource estimation, which was derived from both the circuits based on the Generalized Toffoli gate and circuits based on the Quantum Addition (referred to as GT-ADD and Q-ADD in Table 1, respectively). The circuits based on the Generalized Toffoli gate require $2n + 4$ qubits and $\frac{16}{3}n^5$ Toffoli gate and those based on the Quantum Addition requires $2n + 3$ qubits and $20n^4$ C-NOT gates and $37n^4$ single-qubit gates. Takahashi and Kunihiro proposed the circuit construction that works even for $2n + 2$ qubits for the necessary qubits (Takahashi and Kunihiro 2006). Häner et al. also presented a similar result (Häner 2017).

The resource estimation for solving the elliptic curve discrete logarithm problem was presented in Roetteler et al. (2017), and further improvement is provided in Kurama and Kunihiro (2019).

2.4 Survey of Quantum Experiments for Factoring

In 2001, a research group of IBM performed an experiment for factoring 15 by implementing Shor's algorithm by using Nuclear Magnetic Resonance (NMR) (Van-

Table 2 Number of qubits and elementary gates for 768 and 2048 bits (Kunihiro 2005)

	World record ($n = 768$)		Recommended ($n = 2048$)	
	Qubits	# of gates	Qubits	# of gates
R-ADD	2306	1.22×10^{11}	6146	3.04×10^{12}
GT-ADD	1540		4100	
Q-ADD	1539	8.68×10^{11}	4099	1.22×10^{13}

Table 3 Summary of quantum experiments for factoring

Device	Research group	Year	Target	# of qubits	Embedding of order information
NMR	IBM (Vandersypen 2001)	2001	15	6	✓
Photonic chip	U. of Bristol (Politi 2009)	2009	15	4	✗(used)
Superconductivity	UCSB (Lucero et al. 2012)	2012	15	3	✓
Ion trap	U. Innsbruck (Monz et al. 2016)	2016	15	6	✓
Photon	U. of Bristol (Martin-Lopez et al. 2012)	2012	21	$1 + \log_2 3$	✗(used)

dersypen 2001). Since the group's pioneering work, several experiments based on Shor's algorithm have been conducted. Table 3 summarizes five of these experiments, of which four experiments dealt with the factorization of 15, and the fifth one with the factorization of 21.

Because the bit length of composite 15 is 4, it requires at least 14 qubits with standard construction based on the usual addition (R-ADD) and 10 qubits with the construction based on Takahashi and Kunihiro (2006) to factorize 15. As can be seen, all of the experiments employed fewer qubits than those in the above-mentioned construction for general composites. We can say that the circuits for factoring are customized to factor the target composites such as 15 and 21, and are not based on the general construction. In Sect. 3, we describe the detailed circuits without using the order information based on Lucero et al. (2012), Monz et al. (2016), and Vandersypen (2001). Though their circuits do not use any secret information, they are applicable to specific composite such as $2^n - 1$ for an even integer n , which are never used for RSA composite. In Sect. 4, we describe the detailed circuits by using the order information based on Martin-Lopez et al. (2012) and Politi (2009). These circuit constructions are inappropriate since the order information must be secret.

Fig. 2 Shor's factoring algorithm for $N = 15$

3 Quantum Circuits Without Using the Order Information

Before describing the details of each quantum circuits for factoring 15, we explain a common strategy for factoring 15. The positive integers relatively coprime to 15 are given by 2, 4, 7, 8, 11, 13, and 14. Their order modulo 15 are given by 4, 2, 4, 4, 2, 4, and 2, respectively. Clearly, the elements with order 4 are 2, 7, 8, and 13. In many cases, we consider using them as a . Note that $a^2 \bmod 15 = 4$ for $a = 2, 7, 8$, and 13.

For the element a with the order 4, $a^{2^k} \bmod 15$ is always 1 for integers $k \geq 2$. Hence, $U_{a^{2^k}}$ for $k \geq 2$ becomes an identity operation and they can be ignored in the calculation. On the basis of the above-mentioned observation, it is sufficient to implement $C-U_a$ and $C-U_{a^2 \bmod 15}$ circuits for the modular exponentiation. Here, $a^2 \bmod 15 = 4$ and the necessary operation can be simplified into $C-U_a$ and $C-U_4$. Hence, while constructing the quantum circuits, it is sufficient to consider a multiplication circuit by employing a as $a = 2, 4, 7, 8$, and 13. From the above-mentioned discussion, the general form for factoring $N = 15$ is given by Fig. 2 under the condition that the element of order 4 element is used.

3.1 Quantum Factoring Experiment Shown in Vandersypen (2001)

The literature (Vandersypen 2001) shows an experiment of factoring $N = 15$ using NMR. The experiment uses $a = 7$ as a chosen element. The order of 7 modulo 15 is given by 4.

As described previously, it is sufficient to construct multiplication circuits with 7 and 4. The multiplication circuit with 4 will be constructed by using the following strategy. Here, we denote a 4-bit nonnegative integer by $(y_3 y_2 y_1 y_0)_2$. By multiplying it with 4, we have $(y_3 y_2 y_1 y_0 00)_2$. By calculating the residue by 15, we have $(y_1 y_0 y_3 y_2)_2$. In summary, the multiplication of $(y_3 y_2 y_1 y_0)_2$ by 4 modulo 15 is given by $(y_1 y_0 y_3 y_2)_2$. It is sufficient to construct a circuit transferring $|y_3 y_2 y_1 y_0\rangle$ into $|y_1 y_0 y_3 y_2\rangle$ instead of directly implementing the multiplication circuit. From the above-mentioned discussion, it is sufficient to swap the first and the third qubits and swap the second and the fourth qubits for multiplication with 4 and taking modulo

Fig. 3 Quantum Circuit for Factoring 15 in Vandersypen (2001)

Fig. 4 Experiment for $a = 4$ and $N = 15$ in Lucero et al. (2012)

15. The swap operation can be executed without using ancilla qubits. Furthermore, the controlled-SWAP can be divided into one Toffoli gate and two C-NOT gates.

Subsequently, we explain the multiplication circuit with 7. Their shown circuit does not directly implement the multiplication with 7. We can easily verify that it is sufficient that $|0\rangle|1\rangle$ is mapped to $|0\rangle|1\rangle$ and $|1\rangle|1\rangle$ is mapped to $|1\rangle|7\rangle$ for multiplication with 7 in this situation. This operation can be executed via controlled-addition with 6. In this experiment, the controlled-addition with 6 is implemented by using two controlled-NOT gates.

On the basis of the above-mentioned idea, the authors of Vandersypen (2001) implemented the circuit as depicted in Fig. 3. Note that no ancilla qubit was used in applying U_a and U_4 , and consequently only six qubits were involved in the implementation.

3.2 Quantum Factoring Experiment Shown in Lucero et al. (2012)

This experiment involves the factorization of 15 and uses $a = 4$ as the chosen element. Note that the order of 4 is 2. Hence, it is sufficient to implement U_4 for the experiment. In the circuit shown in Lucero et al. (2012), the circuit for multiplication with 4 is not implemented directly. It is sufficient to implement the circuit that transforms $|0\rangle|1\rangle \rightarrow |0\rangle|1\rangle$ and $|1\rangle|1\rangle \rightarrow |1\rangle|4\rangle$. This operation can be executed via controlled-addition with 3. In this experiment, the controlled-addition with 3 is

implemented by using two C-NOT gates. Summing up the above discussion, the authors in Lucero et al. (2012) presented the circuit depicted in Fig. 4.

Note that no ancilla qubit was used in applying U_4 and consequently only three qubits were involved in the implementation.

3.3 *Quantum Factoring Experiment Shown in Monz et al. (2016)*

The authors presented the circuits not only for $a = 7$ but also for several other a 's in the experiments. Concretely, the authors showed the circuit for $a = 2, 7, 8, 11$, and 13, and $a^2 \bmod 15 = 4$ for these a 's. Hence, it is sufficient to construct the U_a circuit and U_4 circuits. As shown in Sect. 3.1, the U_4 circuit can be constructed using SWAP. In Monz et al. (2016), the authors showed that the multiplication circuit U_a can also be constructed using SWAP and NOT gate.

We first present the multiplication circuit for $a = 2$. We denote the binary representation of a by $(a_3a_2a_1a_0)_2$ as previously. The double of a modulo 15 is given by $(a_2a_1a_0a_3)_2$ in the binary representation. The state $|a_2a_1a_0a_3\rangle$ can be obtained from $|a_3a_2a_1a_0\rangle$ using the following three sequential SWAP operations: SWAP between the first and second qubits, SWAP between the second and third qubits, and then SWAP between the third and fourth qubits. We can verify its correctness by following transition: $|a_3a_2a_1a_0\rangle \rightarrow |a_2a_3a_1a_0\rangle \rightarrow |a_2a_0a_3a_1\rangle \rightarrow |a_2a_0a_1a_3\rangle$.

We then consider the multiplication circuit for $a = 8$. The multiplication of a with 8 is given by $(a_0a_3a_2a_1)_2$ in the binary representation. The state $|a_0a_3a_2a_1\rangle$ can be obtained from $|a_3a_2a_1a_0\rangle$ using the following three sequential SWAP operations: SWAP between the third and fourth qubits, SWAP between the second and third qubits, and then SWAP between the first and second qubits.

We, thus, know that we can implement the multiplication with 2, 4, and 8 by using only the SWAP circuit.

We then implement the multiplication with $a = 7, 11$, and 13; the values of $15 - a$ for them are given by $a = 8, 4$, and 2, respectively. To construct the multiplication circuits with 7, 11, and 13, we will use the above-mentioned property. For the multiplication with $a = 13$, we first apply the multiplication with 2, and we then apply the NOT gate for all of the four qubits. Figure 5 depicts the concrete multiplication circuit with them. We can also obtain the multiplication circuits for $a = 7, 11$ in a similar manner.

Fig. 5 Unitary operations U_2 , U_{13} and the circuit for C-SWAP

4 Quantum Circuits with Explicitly Using the Order information

This section presents two experiments that explicitly use the order information. We want to emphasize that these experiments are inappropriate for employing in factoring algorithms because the purpose of Shor's algorithm is to find the order of a given element.

4.1 Quantum Factoring Experiment of $N = 15$ Shown in Politi (2009)

The authors of Politi (2009) conducted an experiment that factorized 15 with an element $a = 7$. The order of $a = 7$ is given by 4. Because the order is 4, the only four values, namely, 1, 7, 4, and 13 can appear in the second register, and the authors utilized this property. The authors represented these four values by using two bits. Concretely speaking, they adopted the following encoding: $1 \rightarrow 0(= 00)_2$, $7 \rightarrow 1(= 01)_2$, $4 \rightarrow 2(= 10)_2$, $13 \rightarrow 3(= 11)_2$.

As described previously, it is sufficient to implement the multiplication circuits with 7 and 4. The multiplication with 7 corresponds to the addition with +1 under the encoding and the multiplication with 4 corresponds to addition with +2. These operations can be implemented using only one C-NOT gate. Summing up the above-mentioned discussion, the entire circuit is depicted in Fig. 6.

Fig. 6 Quantum circuit for $N = 15$ in Politi (2009)

Fig. 7 Unitary operations U_+ and U_-

4.2 Quantum Factoring Experiment of $N = 21$ Shown in Martin-Lopez et al. (2012)

The target of this experiment is 21. In this experiment, a is set to $a = 4$. Because $a^3 \bmod 21 = 1$, the order of a modulo 21 is given by 3. Note that the purpose of Shor's algorithm is to obtain the order 3. The only three elements, namely, 1, 4, and 16 can appear in the second register.

It is sufficient to construct the quantum circuits $U_{4^{2^k \bmod 21}}$ for $k = 0, 1, 2, \dots$ for the modular exponentiation. Note that $4^{2^k} \bmod 21 = 4$ for even k and $4^{2^k} \bmod 21 = 16$ for odd k . Then, it is sufficient to apply the unitary operation U_4 for even k and U_{16} for odd k .

In the experiment of Martin-Lopez et al. (2012), the following encoding is adapted as in the case of $N = 15$.

$$1 \rightarrow 0, \quad 4 \rightarrow 1, \quad 16 \rightarrow 2$$

We consider the multiplication with 4 and 16 under the aforementioned encoding. The multiplication with 4 is mapped into addition with +1, and the multiplication with 16 is mapped into addition with +2 or, equivalently, -1.

The experiment in Martin-Lopez et al. (2012) utilized a *qutrit*, which takes three quantum states instead of qubits, as the second register. We denote the unitary operations by

$$U_+ : |x\rangle \mapsto |x + 1 \bmod 3\rangle, \quad U_- : |x\rangle \mapsto |x - 1 \bmod 3\rangle.$$

The operations U_+ and U_- act on the quantum states as depicted in Fig. 7.

Fig. 8 Quantum circuit for $N = 21$ in Martin-Lopez et al. (2012)

Using the above-mentioned notation, Fig. 8 depicts the quantum circuit for factoring $N = 21$ described in Martin-Lopez et al. (2012). Here, in the circuit construction, the so-called qubit-recycling technique is employed to reduce the number of qubits. For the details of the qubit-recycling technique, refer to Martin-Lopez et al. (2012).

4.3 Oversimplified Shor's Algorithm (Smolin et al. 2013)

As described previously, the purpose of Shor's algorithm is to find the order of a given element. Hence, the circuit that explicitly utilizes the order information is inappropriate for (even the simplified version of) Shor's factoring algorithm. If we can use the order information, we can, in principle, factorize any large composite. We will explain the details of this fact by following the description provided in Smolin et al. (2013).

The modular exponentiation part in Shor's algorithm constructs the quantum superposition as follows:

$$\frac{1}{2^{m/2}} \sum_{x=0}^{2^m-1} |x\rangle |a^x \bmod N\rangle$$

from the flat superposition $\frac{1}{2^{m/2}} \sum_{x=0}^{2^m-1} |x\rangle |1\rangle$.

However, the circuits described in this section constructs the quantum superposition as follows:

$$\frac{1}{2^{m/2}} \sum_{x=0}^{2^m-1} |x\rangle |x \bmod r\rangle$$

from the flat superposition $\frac{1}{2^{m/2}} \sum_{x=0}^{2^m-1} |x\rangle |0\rangle$.

In this discussion, the following encoding is employed:

$$a^x \bmod N \mapsto x \bmod r.$$

This encoding includes the encodings described in Sects. 4.1 ($r = 4$) and 4.2 ($r = 3$) as a special case. This discussion is mathematically correct, but, it is inappropriate from the computational viewpoint because finding the order r is strongly believed to be infeasible in the classical polynomial time.

Fig. 9 Oversimplified factoring algorithm

This circuit is constructed on the basis of the knowledge of the order r . Under this encoding, the operation $U_{a^{2^j}}$ is transformed into the addition operation with $2^j \bmod r$. Assume that $r = 4$. The unitary operation $U_{a^{2^j}}$ for $j = 0$ corresponds to the addition with 1; that for $j = 1$ corresponds to the addition with 2; that for $j \geq 2$ corresponds to an identity operation. Next, we assume that $r = 3$. The unitary operation $U_{a^{2^j}}$ for even j corresponds to the addition with 1; that for odd j corresponds to the addition with 2 or, equivalently, -1 . Note that all the addition is performed under the modulo 3.

To indicate that this kind of circuit that explicitly utilizes the order information is meaningless for the implementations of Shor's factoring algorithm, Smolin et al. (2013) presented the factoring circuits by using an element with order 2. Because the order r is 2, it is sufficient to construct the superposition as follows:

$$\frac{1}{\sqrt{2}} \sum_{x=0}^1 |x\rangle |0\rangle \mapsto \frac{1}{\sqrt{2}} \sum_{x=0}^1 |x\rangle |x\rangle = \frac{1}{\sqrt{2}} (|00\rangle + |11\rangle).$$

Figure 9 depicts the entire circuit described in Smolin et al. (2013).

We can find the element with order 2 for a large composite N using the following algorithm.

Input: $k \in \mathbb{Z}$

Output: a $2k$ -bit composite N and an element a with order 2 modulo N

Step1: Find two distinct k -bit primes p and q . Compute $N = pq$.

Step2: Find a such that $a \equiv +1 \pmod{p}$ and $a \equiv -1 \pmod{q}$. Concretely, perform the following procedures to compute a .

Step2-1: Calculate $\bar{q} = q^{-1} \pmod{p}$.

Step2-2: Calculate $a \equiv -1 + 2\bar{q}q$.

Furthermore, we provide a SageMath (2020) code for the above-mentioned algorithm with 2048-bit RSA.

```

1 k=1024
2 p=random_prime(2^k-1, false, 2^(k-1))
3 q=random_prime(2^k-1, false, 2^(k-1))
4 N=p*q
5 a= crt(1, -1, p, q)

```

We can easily verify that it holds that $a \equiv +1 \pmod{p}$ and $a \equiv -1 \pmod{q}$. Because $a^2 \equiv 1 \pmod{p}$ and $a^2 \equiv 1 \pmod{q}$, we have $a^2 \equiv 1 \pmod{N}$, and the order of a is a divisor of 2, implying that the order is 1 or 2. Because $a \not\equiv 1 \pmod{N}$, we

Table 4 Level of quantum experiments for factoring

Research Group	Year	Target	Level
IBM (Vandersypen 2001)	2001	15	Level 2
U. of Bristol (Politi 2009)	2009	15	Level 1
UCSB (Lucero et al. 2012)	2012	15	Level 2
U. Innsbruck (Monz et al. 2016)	2016	15	Level 2
U. of Bristol (Martin-Lopez et al. 2012)	2012	21	Level 1

can assert that the order of a is exactly 2. Furthermore, as $\gcd(a^{2/2} - 1, N) = p$, we can find a prime factor p of N .

In Smolin et al. (2013), the authors presented the prime factorization of a 20, 000-bit composite, showing that this kind of oversimplification is meaningless for the implementation of Shor's factoring algorithm.

5 Summary and Concluding Remarks

We reviewed the resource estimation of quantum factoring based on Shor's algorithm. We then presented a survey of the state-of-the-art circuit construction. We also indicated some of them as inappropriate for factoring circuits because the order information was embedded in the circuits (Sect. 4). The others considerably utilized the property of the target composite, and hence, they have no extensibility to the general composite (Sect. 3).

More experiments on factoring based on Shor's algorithm will be conducted using various devices. As we mentioned in this paper, we have to carefully analyze the circuit construction.

Based on the current status of quantum experiments for factoring, we introduce the following three levels of circuit construction for quantum factoring.

Level 1 Quantum factoring: The order information is embedded in the circuit.

The experiment under Level 1 cannot be considered as a quantum experiment for factoring.

Level 2 Quantum factoring: The circuit relies considerably on the property of a target composite. The experiment under Level 2 can be considered as a quantum experiment for factoring, meaning that the compiled version of the circuits is acceptable. However, we cannot apply this circuit construction to the general composite, and hence, this circuit construction has no scalability.

Level 3 Quantum factoring: The circuit does not use any specific property of the target composite. The circuit under Level 3 is desirable.

Table 4 presents the levels for quantum factoring circuits shown in this paper. As can be seen, there is no experiment with Level 3.

Acknowledgements This research was partially supported by JST CREST Grant Number JPMJCR14D6, Japan and JSPS KAKENHI Grant Number JP16H02780. The authors thank Dr. Tetsuya Izu, who gave the information about quantum factoring circuits. They also thank Prof. Naoki Yamamoto and Prof. Yutaka Shikano for helpful discussions.

References

- T. Häner, M. Roetteler, K.M. Svore, Factoring using $2n + 2$ qubits with Toffoli based modular multiplication. *Quantum Inf. Comput.* **17**(7&8), 673–684 (2017)
 IBM Q, <https://www.research.ibm.com/ibm-q/>
- T. Kleinjung, K. Aoki, J. Franke, A. Lenstra, E. Thome, J. Bos, P. Gaudry, A. Kruppa, P. Montgomery, D.A. Osvik, H. te Riele, A. Timofeev, P. Zimmermann, Factorization of a 768-bit RSA modulus in *Proceedings of CRYPTO2010*, LNCS 6223 (2010), pp. 333–350
- N. Kunihiro, Exact analyses of computational time for factoring in quantum computers, in *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. E88-A, No. 1 (2005), pp. 105–111
- R. Kurama, N. Kunihiro, New quantum algorithms for modular inverse and its application on the elliptic curve discrete logarithm problem, in *The Poster Presentation of AQIS2019* (2019)
- E. Lucero, R. Barends, Y. Chen, J. Kelly, M. Mariantoni, A. Megrant, P. O’Malley, D. Sank, A. Vainsencher, J. Wenner, T. White, Y. Yin, A.N. Cleland, J.M. Martinis, Computing prime factors with a Josephson phase qubit quantum processor. *Nat. Phys.* **8**, 719–723 (2012)
- E. Martin-Lopez, A. Laing, T. Lawson, R. Alvarez, X.-Q. Zhou, J.L. O’Brien, Experimental realisation of Shor’s quantum factoring algorithm using qubit recycling. *Nat. Photonics* **6**, 773–776 (2012)
- T. Monz, D. Nigg, E.A. Martinez, M.F. Brandl, P. Schindler, R. Rines, S.X. Wang, I.L. Chuang, R. Blatt, Realization of a scalable Shor algorithm. *Science* **351**(6277), 1068–1070 (2016)
- M.A. Nielsen, I.L. Chuang, *Quantum Computation and Quantum Information* (Cambridge University Press, Cambridge, 2000)
- NIST, Post-Quantum Cryptography, <https://csrc.nist.gov/projects/post-quantum-cryptography>
- A. Politi, J.C.F. Matthews, J.L. O’Brien, Shor’s quantum factoring algorithm on a photonic chip. *Science* **325**(5945), 1221 (2009)
- J. Preskill, Quantum computing in the NISQ era and beyond. *Quantum* **2**, 79 (2018)
- R.L. Rivest, A. Shamir, L.M. Adelman, A method for obtaining digital signature and public-key cryptosystems, MIT-LCS-TM-082 (1977)
- M. Roetteler, M. Naehrig, K.M. Svore, K. Lauter, Quantum resource estimates for computing elliptic curve discrete logarithms, in *Proceedings of ASIACRYPT 2017*, LNCS 10625 (2017), pp. 241–270
- SageMath – Open-Source Mathematical Software System, <http://www.sagemath.org/>
- P. Shor, Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM J. Comput.* **26**(5), 1484–1509 (1997)
- J.A. Smolin, G. Smith, A. Vargo, Oversimplifying quantum factoring. *Nature* **499**, 163–165 (2013)
- Y. Takahashi, N. Kunihiro, A quantum circuit for Shor’s factoring algorithm using $2n + 2$ qubits. *Quantum Inf. Comput.* **6**(2), 184–192 (2006)
- L.M.K. Vandersypen, M. Steffen, G. Breyta, C.S. Yannoni, M.H. Sherwood, I.L. Chuang, Experimental realization of Shor’s quantum factoring algorithm using nuclear magnetic resonance. *Nature* **414**, 883–887 (2001)

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.