Cryptanalysis of a Public Key Cryptosystem Based on Data Complexity Under Quantum Environment

Zhengjun Jing(✉), Chunsheng Gu, and Peizhong Shi

School of Computer Engineering, Jiangsu University of Technology, ChangZhou, China
zhengjun_jing@163.com

Abstract. Shor presented a quantum algorithm to factor large integers and compute discrete logarithms in polynomial time. As a result, public key cryptosystems, such as RSA, ElGamal and ECC, which are based on these computational assumptions will become insecure with the advent of quantum computers. To construct a secure anti-quantum public-key cryptosystem, Wu et al. introduced the notion of data complexity under quantum environment. Based on the hardness of NP-complete problems and data complexity, they presented a new public key cryptosystem. Using Shor's quantum algorithm, we break their public key cryptosystem by directly solving the private key from the public key. Therefore, their public key cryptosystem is insecure in a quantum computer.

Keywords: Public-key cryptosystem · Data complexity · Quantum algorithm · Discrete logarithm · Integer factorization

1 Introduction

Public-key cryptography is indispensable in secure communications for an open networked environment such as the Internet. In 1976, Diffie and Helman introduced the notion of public-key cryptography in "New Directions in Cryptography" [1], and proposed a key exchange protocol based on discrete logarithms over an insecure channel. However, they did not present public-key cryptosystems in [1]. Subsequently, Rivest, Shamir, and Adleman [2] described a public-key encryption scheme and a signature scheme, called RSA, whose security depends on the difficulty of factoring. Afterwards, ElGamal [3] presented a public key cryptosystem and a signature scheme which are based on discrete logarithm assumption. Miller [4] discussed the use of elliptic curves in cryptography and

Supported by the National Natural Science Foundation of China (Nos. 61672270, 61602216, and 61702236), the Qing Lan Project for Young Researchers of Jiangsu Province of China (No. KYQ14004), and Jiangsu Overseas Research & Training Program for University Prominent Young & Middle-aged Teachers and Presidents.

© ICST Institute for Computer Sciences, Social Informatics and Telecommunications Engineering 2019
Published by Springer Nature Switzerland AG 2019. All Rights Reserved
J. Li et al. (Eds.): SPNCE 2019, LNICST 284, pp. 411–420, 2019.
https://doi.org/10.1007/978-3-030-21373-2_32

proposed an analogue of the Diffie-Hellman key exchange protocol. Furthermore, using the discrete logarithm on the elliptic curve, we can construct a public key encryption scheme and signature scheme similar to that of ElGamal. However, Shor [5] described a polynomial time quantum algorithm which is able to factor large integers and compute discrete logarithms. Consequently, the public-key cryptosystems RSA, ElGamal, and ECC, which are based on these computational problems become insecure with the advent of quantum computers.

The study in post-quantum cryptography has seen a series of activities that constructed many post-quantum public-key cryptosystems [6]. These schemes mainly include code-based public-key cryptosystems [7], lattice-based public-key cryptosystems [8–11], multivariate public-key cryptosystems [12,13], quantum public-key cryptosystems based on quantum physics [14,15], DNA-based public-key cryptosystems [16]. Although these schemes are believed to be resistant to quantum attacks, it is always better to provide more candidate post-quantum public-key schemes. Recently, Wu et al. [17] introduced the concept of data complexity to the public key cryptosystems under a quantum environment, and described a public key cryptosystem and a signature scheme based on the hardness of NPC problems and data complexity. They considered several possible quantum attacks for their schemes and claimed that their schemes are secure in a quantum computer. However, they did not provide any rigorous proof of security for their schemes. Therefore, it is necessary to further study the security of their schemes [17].

Our main contribution is to prove that the public key cryptosystem and signature scheme proposed by Wu et al. [17] are insecure on a quantum computer. Our key observation is that there exists a polynomial time quantum algorithm that transforms the public key of their schemes [17] into a system of linear equations. This is because the matrix operations $\odot, \otimes$ used by their schemes are both component-wise multiplication. Consequently, by the definitions of $\odot, \otimes$, we can generate a linear system of the private key for their schemes on a quantum computer. Then, applying the Gaussian elimination method, we can obtain the private key from solving this linear system. Therefore, the public key cryptosystem and signature scheme in [17] are not immune to quantum attacks.

The remainder of this paper is organized as follows. We first give some preliminaries in Sect. 2, and describe the public key cryptosystem (PKC) [17] in Sect. 3. We present the cryptanalysis of PKC in Sect. 4. Finally, in Sect. 5 we conclude this paper and provide some suggestions for improvement.

2 Preliminaries

In this paper, quantum algorithms are only used to decompose large integers and to compute discrete logarithms over a finite field, all other algorithms are classical ones. For simplicity, we use Shor's quantum algorithm as a black box algorithm, and do not define quantum computation in this paper.

Data complexity, that is used for the differential attack of DES, refers to an attack algorithm requires the number of plaintext-ciphertext pairs. Wu et al. [17]

introduced the notion of data complexity under a quantum environment to construct an anti-quantum public key cryptosystem. The data complexity of quantum Turing machine (QTM) defined by [17] means the sum of input data and processing data of an algorithm in QTM. Since we do not require data complexity in our cryptanalysis, consequently we do not provide the related definitions of data complexity for simplicity.

In the following, we first give some notations and definitions of the related operations in this paper. Then, we briefly review the discrete logarithm problems and the integer factorization problems, and present Shor's quantum algorithm with two lemmas.

2.1 Notations

Throughout this paper, let n be the security parameter. We write $[n] = \{1, 2, \cdots, n\}$. Let p be a prime, $\mathbb{Z}_p = \mathbb{Z}/p\mathbb{Z}$, and $\mathbb{Z}_p^* = \mathbb{Z}_p \setminus \{0\}$. By convention, vectors are in column form. We use bold lower-case letters like $\mathbf{a}$ to denote column vectors, and bold upper-case letters like $\mathbf{A}$ to denote matrices. We use the superscript T to denote the transpose of vector or matrix, e.g. $\mathbf{a}^T, \mathbf{A}^T$.

Given an element $g \in \mathbb{Z}_p^*$ and matrices $\mathbf{A} = (a_{i,j}), \mathbf{B} = (b_{i,j}) \in \mathbb{Z}_p^{*(m\times m)}$, we define some operations of matrices in $\mathbb{Z}_p^*$ that are used in this paper as follows:

$$
\begin{aligned}
&g^{\mathbf{A}} = (g^{a_{i,j}})_{m\times m}, i, j \in [m],\\
&\mathbf{A}^{-1} = (a_{i,j}^{-1})_{m\times m}, i, j \in [m],\\
&\mathbf{A} \odot \mathbf{B} = (a_{i,j} b_{i,j})_{m\times m}, i, j \in [m],\\
&\mathbf{A}^t = \underbrace{\mathbf{A} \odot \mathbf{A} \cdots \odot \mathbf{A}}_{t} = (a_{i,j}^t)_{m\times m}, i, j \in [m],
\end{aligned}
$$

$$
\mathbf{A} \otimes \mathbf{B} = \begin{pmatrix} a_{1,1}\mathbf{B} & a_{1,2}\mathbf{B} & \cdots & a_{1,m}\mathbf{B} \\ a_{2,1}\mathbf{B} & a_{2,2}\mathbf{B} & \cdots & a_{2,m}\mathbf{B} \\ \vdots & \vdots & \cdots & \vdots \\ a_{m,1}\mathbf{B} & a_{m,2}\mathbf{B} & \cdots & a_{m,m}\mathbf{B} \end{pmatrix} \in \mathbb{Z}_p^{*(m^2\times m^2)}.
$$

2.2 Discrete Logarithm Problem

The discrete logarithm problem defined in $\mathbb{Z}_p^*$ is computationally intractable for classic computers. That is, there is no polynomial time algorithm for the discrete logarithm problem on a classical computer. However, there exists an efficient quantum algorithm that solves the discrete logarithm problem.

A group G is cyclic if and only if there exists an element $g \in G$ such that for every element $a \in G$, there exists an integer x such that $g^x = a$. In this paper, we call g a generator of G.

Definition 1 (generator, [18]). Given a prime p, an integer $g \in \mathbb{Z}_p^*$ is called a generator of $\mathbb{Z}_p^*$ if $p - 1$ is the smallest positive integer such that $g^{p-1} = 1 \mod p$.

Definition 2 (discrete logarithm problem, [18]). Given a prime p, a generator $g \in \mathbb{Z}_p^*$, and an integer $a \in \mathbb{Z}_p^*$, the discrete logarithm problem is to find an integer $x \in [p-1]$ such that $a = g^x \mod p$.

Lemma 1 (Shor, [5]). Given a prime p, a generator g, and any number $a \in \mathbb{Z}_q^*$, there exists a polynomial time quantum algorithm, which finds the exponent $\overline{a}$ such that $a = g^{\overline{a}} \mod p$.

2.3 Integer Factorization Problem

Integer factorization problem is a product of decomposing a composite number into smaller integers. The factorization is called prime factorization, if these smaller integers must be prime numbers. According to the fundamental arithmetic theorem, any integer greater than one has a unique prime factorization. Similarly, there is no polynomial time algorithm for the integer factorization problem for classical computers. But there exists an efficient quantum algorithm for the integer factorization problem.

Definition 3 (integer factorization problem, [18]). Given an integer $n \in \mathbb{Z}$, factor n into primes, namely, $n = \prod_{i=1}^{k} p_i^{e_i}, e_i \in \mathbb{N}$.

Lemma 2 (Shor, [5]). Given an integer n, there exists a polynomial time quantum algorithm, which factors n into primes, namely, $n = \prod_{i=1}^{k} p_i^{e_i}, e_i \in \mathbb{N}$.

3 Public Key Cryptosystem (PKC)

In this section, we adaptively describe the public key cryptosystem (PKC) in [17]. This public key scheme consists of three algorithms: KeyGen, Encryption, and Decryption.

KeyGen

(1) Choose a prime $p > 2^m p_1^{r_1} \cdots p_s^{r_s}$, where $p_1, \cdots, p_s$ are odd primes and $r_1, \cdots, r_s \in \mathbb{N}$.

(2) Randomly choose three different integers $t_1, t_2, t_3 \in [\varphi(p)]$, where $\varphi(p) = p - 1$ is the Euler function of p.

(3) Randomly choose three $m \times m$-dimensional matrices

$$\mathbf{A} = (a_{i,j})_{m\times m}, \mathbf{B} = (b_{i,j})_{m\times m}, \mathbf{D} = (d_{i,j})_{m\times m},$$

where $a_{i,j}, b_{i,j}, d_{i,j} \in \mathbb{Z}_p^*$.

(4) Compute $\mathbf{Y}_1, \mathbf{Y}_2, \mathbf{Y}_3$ as follows:

$$\begin{cases} \mathbf{Y}_1 = \mathbf{A}^{t_1} \odot \mathbf{B}^{t_2} \odot \mathbf{D}^{t_3} \mod p, \\ \mathbf{Y}_2 = \mathbf{B}^{t_1} \odot \mathbf{D}^{t_2} \odot \mathbf{A}^{t_3} \mod p, \\ \mathbf{Y}_3 = \mathbf{D}^{t_1} \odot \mathbf{A}^{t_2} \odot \mathbf{B}^{t_3} \mod p, \end{cases}$$

such that $y_{1_{i,j}}, y_{2_{i,j}}, y_{3_{i,j}} \geq 2^m, i, j \in [m]$. Otherwise, it returns to Step (3).

(5) Output a public key $pk = \{p, \mathbf{A}, \mathbf{B}, \mathbf{D}, \mathbf{Y}_1, \mathbf{Y}_2, \mathbf{Y}_3\}$ and a private key $sk = \{t_1, t_2, t_3\}$.

Encryption

(1) Given the public key pk, let $\mathbf{M} = (m_{i,j})_{m^3 \times m^3}$ with $m_{i,j} \in \mathbb{Z}_p^*$ be an m^6-dimensional plaintext.
(2) Randomly choose three different integers $s_1, s_2, s_3 \in [p-1]$.
(3) Compute $\mathbf{U} = \mathbf{Y}_1^{s_1} \otimes \mathbf{Y}_2^{s_2} \otimes \mathbf{Y}_3^{s_3} \mod p$.
(4) Compute $\{\mathbf{C}, \mathbf{C}_1, \mathbf{C}_2, \mathbf{C}_3\}$ as follows:

$$\begin{cases} \mathbf{C} &= \mathbf{U} \odot \mathbf{M} \mod p, \\ \mathbf{C}_1 &= \mathbf{A}^{s_1} \otimes \mathbf{B}^{s_2} \otimes \mathbf{D}^{s_3} \mod p, \\ \mathbf{C}_2 &= \mathbf{B}^{s_1} \otimes \mathbf{D}^{s_2} \otimes \mathbf{A}^{s_3} \mod p, \\ \mathbf{C}_3 &= \mathbf{D}^{s_1} \otimes \mathbf{A}^{s_2} \otimes \mathbf{B}^{s_3} \mod p. \end{cases}$$

(5) Output a four-tuple ciphertext $ct = \{\mathbf{C}, \mathbf{C}_1, \mathbf{C}_2, \mathbf{C}_3\}$.

Decryption

(1) Given the private key sk, let $ct = \{\mathbf{C}, \mathbf{C}_1, \mathbf{C}_2, \mathbf{C}_3\}$ be a four-tuple ciphertext.
(2) Compute $\mathbf{V} = \mathbf{C}_1^{t_1} \odot \mathbf{C}_2^{t_2} \odot \mathbf{C}_3^{t_3} \mod p$.
(3) Output the plaintext $\mathbf{M} = \mathbf{V}^{-1} \odot \mathbf{C} \mod p$.

4 Cryptanalysis of PKC

In this section, we present a polynomial time quantum algorithm that finds the private key from the public key of PKC. As a result, the PKC in [17] is insecure on a quantum computer.

Our main idea is that, by the component-wise multiplication of matrix operation $\odot$ in the public key, we transform a system of exponential equations into a system of linear equations using the Shor's quantum algorithms, and solve the private key using Gaussian elimination.

To implement the above transformation, the key is how to efficiently find a generator g of $\mathbb{Z}_p^*$. In the following, we give three well-known lemmas to efficiently generate a generator g of $\mathbb{Z}_p^*$. Concretely speaking, Lemma 3 shows that $\mathbb{Z}_p^*$ has many generators, Lemma 4 gives a method of determining whether an element is a generator of $\mathbb{Z}_p^*$, and Lemma 5 describes a quantum polynomial time algorithm that produces a generator of $\mathbb{Z}_p^*$ using the quantum polynomial time algorithm of integer factorization in Lemma 2.

Lemma 3. Suppose that p is a prime, then there exist $\varphi(p-1)$ generators in $\mathbb{Z}_p^*$.

Proof. According to Theorem 7.28 in [18] (or Theorem 2.18 in [19]), $\mathbb{Z}_p^*$ is a cyclic group for a prime p. Hence, $\mathbb{Z}_p^*$ has at least a generator.

Now, assume that g is a generator of $\mathbb{Z}_p^*$. Namely, $g^{p-1} = 1 \mod p$ and for any $1 \leq k < p-1$, $g^k \neq 1 \mod p$. Therefore, if r and $p-1$ are relatively prime, then $g_1 = g^r$ is also a generator of $\mathbb{Z}_p^*$. This is because $p-1$ is the smallest positive integer such that $g_1^{p-1} = 1 \mod p$. Again since $\varphi(p-1)$ integers in $[p-1]$ are prime to $p-1$, the result follows. □

Lemma 4. Suppose that p is a prime and $p-1 = \prod_{i=1}^{k} p_i^{e_i}$ is the prime factorization of $p-1$. Then g is a generator of $\mathbb{Z}_p^*$ if and only if for each $i \in [k]$, $g^{\frac{p-1}{p_i}} \neq 1 \mod p$.

Proof. It is easy to verify that if g is a generator of $\mathbb{Z}_p^*$, then for each $i \in [k]$, $g^{\frac{p-1}{p_i}} \neq 1 \mod p$.

Now, we show the opposite direction. Without loss of generality, let r be the smallest positive integer such that $g^r = 1 \mod p$. By contradiction, assume $r < p-1$. Since p is a prime, $g^{p-1} = 1 \mod p$. If $r \nmid (p-1)$, then there exist two positive integers s, k such that $p-1 = kr + s$ and $0 < s < r$. So, we have $g^s = 1 \mod p$. This contradicts that r is the smallest positive integer such that $g^r = 1 \mod p$. Hence, $r|(p-1)$. As a result, there exists a prime p_i which satisfies $r|\frac{p-1}{p_i}$. So, $g^{\frac{p-1}{p_i}} = 1 \mod p$. This contradicts the condition that for each $i \in [k]$, $g^{\frac{p-1}{p_i}} \neq 1 \mod p$. Thus, $r = p-1$. Consequently, g is a generator of $\mathbb{Z}_p^*$. □

Lemma 5. Given a prime p, there exists a probabilistic polynomial time quantum algorithm which finds a generator g of $\mathbb{Z}_p^*$.

Proof. Using the polynomial time quantum algorithm in Lemma 2, we factor $p-1$ into primes. Without loss of generality, let $p-1 = \prod_{i=1}^{k} p_i^{e_i}$.

According to Lemma 4, we can find a generator g of $\mathbb{Z}_p^*$ as follows.

(1) Randomly select $g \in \mathbb{Z}_p^*$.

(2) If $g^{\frac{p-1}{p_i}} \neq 1 \mod p$ for each $i \in [k]$, then g is a generator of $\mathbb{Z}_p^*$. Otherwise, repeat from (1).

Obviously, selecting and testing an element in the steps (1), (2) take polynomial time in n.

In order to find a generator, we analyze how many random elements need to be selected. By Lemma 3, the number of generators in $\mathbb{Z}_p^*$ is $\varphi(p-1)$. Again since $\varphi(p-1) > \frac{p-1}{6 \log\log(p-1)}$ by Theorem 15 in [20], the probability that a random element is a generator is about $\frac{1}{6 \log\log(p-1)}$. Therefore, we expect to have to select $O(\log\log p)$ random candidate elements for g to get a generator with overwhelming probability.

It is easy to verify that the above algorithm is a probabilistic polynomial time quantum algorithm. □

By Lemma 5, we can efficiently compute a generator g of the cyclic group $\mathbb{Z}_p^*$. For simplicity, we assume that a generator g of $\mathbb{Z}_p^*$ is given in the following Lemmas 6 and 7.

Lemma 6. Given a prime p, a generator g, and a matrix $\mathbf{A} = (a_{i,j}) \in {\mathbb{Z}_p^*}^{(m \times m)}$, there exists a polynomial time quantum algorithm, which finds the exponent matrix $\overline{\mathbf{A}}$ such that $\mathbf{A} = g^{\overline{\mathbf{A}}} \mod p$.

Proof. For each entry $a_{i,j}, i,j \in [m]$ in $\mathbf{A}$, we compute $\overline{a}_{i,j}$ using the quantum algorithm of discrete logarithm in Lemma 1 such that $a_{i,j} = g^{\overline{a}_{i,j}} \mod p$. Thus, $\overline{\mathbf{A}} = (\overline{a}_{i,j})_{m\times m}$ and $g^{\overline{\mathbf{A}}} = (g^{\overline{a}_{i,j}})_{m\times m} = (a_{i,j})_{m\times m} = \mathbf{A} \mod p$. The Lemma 6 follows. $\square$

Lemma 7. Given a prime p, a generator g, and a matrix $\mathbf{A} = (a_{i,j}) \in {\mathbb{Z}_p^*}^{(m \times m)}$, then $\mathbf{A}^t = g^{\overline{\mathbf{A}} \times t} \mod p$, where $\mathbf{A} = g^{\overline{\mathbf{A}}} \mod p$.

Proof. Given $\mathbf{A} = (a_{i,j})_{m\times m}, i,j \in [m]$, according to the definition of $\mathbf{A}^t$, we have

$$\begin{aligned}
\mathbf{A}^t &= \underbrace{\mathbf{A} \odot \mathbf{A} \cdots \odot \mathbf{A}}_{t} \\
&= (a_{i,j}^t)_{m\times m} \\
&= ((g^{\overline{a}_{i,j}})^t)_{m\times m} \\
&= (g^{\overline{a}_{i,j} \times t})_{m\times m} \\
&= g^{(\overline{a}_{i,j})_{m\times m} \times t} \\
&= g^{\overline{\mathbf{A}} \times t} \mod p.
\end{aligned}$$

The Lemma 7 follows. $\square$

We are now in a position to prove the main theorem.

Theorem 1. Given the public key $pk = \{p, \mathbf{A}, \mathbf{B}, \mathbf{D}, \mathbf{Y}_1, \mathbf{Y}_2, \mathbf{Y}_3\}$, there exists a probabilistic polynomial time quantum algorithm, which solves the secret key $sk = \{t_1, t_2, t_3\}$.

Proof. According to KeyGen, p is a prime. By Lemma 5, we can efficiently find a generator g of the cyclic group $\mathbb{Z}_p^*$.

Thus, given pk, we can compute $\overline{\mathbf{A}}, \overline{\mathbf{B}}, \overline{\mathbf{D}}, \overline{\mathbf{Y}}_1, \overline{\mathbf{Y}}_2, \overline{\mathbf{Y}}_3$ using Lemma 6 such that

$$\begin{cases} \mathbf{A} = g^{\overline{\mathbf{A}}} \mod p, \\ \mathbf{B} = g^{\overline{\mathbf{B}}} \mod p, \\ \mathbf{D} = g^{\overline{\mathbf{D}}} \mod p, \end{cases} \quad \begin{cases} \mathbf{Y}_1 = g^{\overline{\mathbf{Y}}_1} \mod p, \\ \mathbf{Y}_2 = g^{\overline{\mathbf{Y}}_2} \mod p, \\ \mathbf{Y}_3 = g^{\overline{\mathbf{Y}}_3} \mod p. \end{cases}$$

Since $\mathbf{A} = (a_{i,j})_{m\times m} = (g^{\overline{a}_{i,j}})_{m\times m}$, $\mathbf{B} = (b_{i,j})_{m\times m} = (g^{\overline{b}_{i,j}})_{m\times m}$, and $\mathbf{D} = (d_{i,j})_{m\times m} = (g^{\overline{d}_{i,j}})_{m\times m}$, then by Lemma 7, we get

$$\begin{aligned}
\mathbf{A}^{t_1} &= g^{t_1 \overline{\mathbf{A}}} \\
\mathbf{B}^{t_2} &= g^{t_2 \overline{\mathbf{B}}} \\
\mathbf{D}^{t_3} &= g^{t_3 \overline{\mathbf{D}}}
\end{aligned}$$

Consequently, we have

$$\mathbf{Y}_1 = \mathbf{A}^{t_1} \odot \mathbf{B}^{t_2} \odot \mathbf{D}^{t_3} = g^{t_1\overline{\mathbf{A}}+t_2\overline{\mathbf{B}}+t_3\overline{\mathbf{D}}} = g^{\overline{\mathbf{Y}}_1} \mod p.$$

Using same methods, we can obtain

$$\mathbf{Y}_2 = \mathbf{B}^{t_1} \odot \mathbf{D}^{t_2} \odot \mathbf{A}^{t_3} = g^{t_1\overline{\mathbf{B}}+t_2\overline{\mathbf{D}}+t_3\overline{\mathbf{A}}} = g^{\overline{\mathbf{Y}}_2} \mod p,$$
$$\mathbf{Y}_3 = \mathbf{D}^{t_1} \odot \mathbf{A}^{t_2} \odot \mathbf{B}^{t_3} = g^{t_1\overline{\mathbf{D}}+t_2\overline{\mathbf{A}}+t_3\overline{\mathbf{B}}} = g^{\overline{\mathbf{Y}}_3} \mod p.$$

Therefore, we can generate a system of linear equation as follows:

$$\begin{cases} t_1\overline{\mathbf{A}} + t_2\overline{\mathbf{B}} + t_3\overline{\mathbf{D}} &= \overline{\mathbf{Y}}_1 \mod (p-1) \\ t_1\overline{\mathbf{B}} + t_2\overline{\mathbf{D}} + t_3\overline{\mathbf{A}} &= \overline{\mathbf{Y}}_2 \mod (p-1) \\ t_1\overline{\mathbf{D}} + t_2\overline{\mathbf{A}} + t_3\overline{\mathbf{B}} &= \overline{\mathbf{Y}}_3 \mod (p-1). \end{cases}$$

Since there exist $3\,m^2$ $(m \geq 1)$ linear equations in the above equation system, we can solve three unknown variables $t_1, t_2, t_3 \in [p-1]$.

It is not difficult to verify that the above computations take a polynomial time in quantum computers. □

Furthermore, we can obtain the following result.

Theorem 2. Given the public key $pk = \{p, \mathbf{A}, \mathbf{B}, \mathbf{D}, \mathbf{Y}_1, \mathbf{Y}_2, \mathbf{Y}_3\}$ and a four-tuple ciphertext $ct = \{\mathbf{C}, \mathbf{C}_1, \mathbf{C}_2, \mathbf{C}_3\}$, then exists a polynomial time quantum algorithm, which recovers the plaintext $\mathbf{M}$ from the ciphertext ct.

Proof. Using Theorem 1, we can compute the private key sk from the public key pk. Then, we directly decrypt the ciphertext ct using the private key sk, and obtain the corresponding plaintext $\mathbf{M}$ in ct. □

5 Conclusions

In this paper, we present a polynomial time quantum algorithm that finds the private key from the public key of PKC in [17]. Furthermore, we also provide a polynomial-time quantum algorithm to solve the private key of the signature scheme in [17]. Consequently, their public key cryptosystem is insecure in a quantum computer.

Our results show that there is still much work to be done to construct secure anti-quantum public key cryptosystem using data complexity. Since our attack mainly depends on component-wise multiplication in matrix operations $\odot, \otimes$, a possible improvement is to change matrix operations $\odot, \otimes$ to prevent attackers from generating discrete logarithm problems based on public keys.

Acknowledgements. This work was supported by the National Natural Science Foundation of China (Nos. 61672270, 61602216, and 61702236), the Qing Lan Project for Young Researchers of Jiangsu Province of China (No. KYQ14004), and Jiangsu Overseas Research & Training Program for University Prominent Young & Middle-aged Teachers and Presidents.

References

1. Diffie, W., Hellman, M.E.: New directions in cryptography. IEEE Trans. Inf. Theory **IT–22**, 644–654 (1976)
2. Rivest, R.L., Shamir, A., Adleman, L.M.: A method for obtaining digital signatures and public-key cryptosystems. Commun. ACM **21**(2), 120–126 (1978)
3. ElGamal, T.: A public key cryptosystem and a signature scheme based on discrete logarithms. In: Blakley, G.R., Chaum, D. (eds.) CRYPTO 1984. LNCS, vol. 196, pp. 10–18. Springer, Heidelberg (1985). https://doi.org/10.1007/3-540-39568-7_2
4. Miller, V.S.: Use of elliptic curves in cryptography. In: Williams, H.C. (ed.) CRYPTO 1985. LNCS, vol. 218, pp. 417–426. Springer, Heidelberg (1986). https://doi.org/10.1007/3-540-39799-X_31
5. Shor, P.W.: Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. SIAM J. Comput. **26**, 1484–1509 (1997)
6. Buchmann, J.A., Butin, D., Göpfert, F., Petzoldt, A.: Post-quantum cryptography: state of the art. In: Ryan, P.Y.A., Naccache, D., Quisquater, J.-J. (eds.) The New Codebreakers. LNCS, vol. 9100, pp. 88–108. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-49301-4_6
7. McEliece, R.J.: A public-key cryptosystem based on algebraic coding theory. Deep Space Network Progress Report 44, pp. 114–116 (1978)
8. Peikert, C.: Lattice cryptography for the internet. In: Mosca, M. (ed.) PQCrypto 2014. LNCS, vol. 8772, pp. 197–219. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-11659-4_12
9. Stehlé, D., Steinfeld, R.: Making NTRU as secure as worst-case problems over ideal lattices. In: Paterson, K.G. (ed.) EUROCRYPT 2011. LNCS, vol. 6632, pp. 27–47. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-20465-4_4
10. Lyubashevsky, V., Peikert, C., Regev, O.: On ideal lattices and learning with errors over rings. In: Gilbert, H. (ed.) EUROCRYPT 2010. LNCS, vol. 6110, pp. 1–23. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-13190-5_1
11. Regev, O.: On lattices, learning with errors, random linear codes, and cryptography. In: STOC 2005, pp. 84-93 (2005)
12. Tao, C., Diene, A., Tang, S., Ding, J.: Simple matrix scheme for encryption. In: Gaborit, P. (ed.) PQCrypto 2013. LNCS, vol. 7932, pp. 231–242. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-38616-9_16
13. Petzoldt, A., Chen, M.-S., Yang, B.-Y., Tao, C., Ding, J.: Design principles for HFEv- based multivariate signature schemes. In: Iwata, T., Cheon, J.H. (eds.) ASIACRYPT 2015. LNCS, vol. 9452, pp. 311–334. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-48797-6_14
14. Bennett, C.H., DiVincenzo, D.P., Smolin, J.A., Wootters, W.K.: Mixed-state entanglement and quantum error correction. Phys. Rev. A **54**, 3824–3851 (1996)
15. Shi, J.J., Shi, R.H., Guo, Y., Peng, X.Q., Tang, Y.: Batch proxy quantum blind signature scheme. Sci. China Inf. Sci. **56**, 1–9 (2013). 052115
16. Lai, X.J., Lu, M.X., Qin, L., Han, J.S., Fang, X.W.: Asymmetric encryption and signature method with DNA technology. Sci. China Inf. Sci. **53**(3), 506–514 (2010)
17. Wu, W.Q., Zhang, H.G., Wang, H.Z., Mao, S.W., Jia, J.W., Liu, J.H.: A public key cryptosystem based on data complexity under quantum environment. Sci. China Inf. Sci. **58**, 1–11 (2015). 110102

18. Shoup, V.: A Computational Introduction to Number Theory and algebra, 2nd edn. Cambridge University Press, Cambridge (2008)
19. Goldwasser, S., Bellare, M.: Lecture Notes on Cryptography (2008). http://cseweb.ucsd.edu/mihir/papers/gb.html
20. Rosser, J., Schoenfield, L.: Approximate formulas for some functions of prime numbers. Ill. J. Math. **6**, 64–94 (1962)