SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •
Special Focus on Security of Cyberspace

November 2015, Vol. 58 110102:1–110102:11
doi: 10.1007/s11432-015-5408-5

A public key cryptosystem based on data complexity under quantum environment

WU WanQing[1], ZHANG HuanGuo[1], WANG HouZhen[1,2*], MAO ShaoWu[1], JIA JianWei[1] & LIU JinHui[1]

[1]*Computer School, Wuhan University, Wuhan* 430072*, China;*
[2]*State Key Laboratory of Cryptology, Beijing* 100878*, China*

Received April 13, 2015; accepted September 1, 2015; published online September 29, 2015

Abstract Since the Shor algorithm showed that a quantum algorithm can efficiently calculate discrete logarithms and factorize integers, it has been used to break the RSA, EIGamal, and ECC classical public key cryptosystems. This is therefore a significant issue in the context of ensuring communication security over insecure channels. In this paper, we prove that there are no polynomial-size quantum circuits that can compute all Boolean functions (of which there are 2^{2^n} cases) in the standard quantum oracle model. Based on this, we propose the notion of data complexity under a quantum environment and suggest that it can be used as a condition for post-quantum computation. It is generally believed that NP-complete problems cannot be solved in polynomial time even with quantum computers. Therefore, a public key cryptosystem and signature scheme based on the difficulty of NP-complete problems and the notion of data complexity are presented here. Finally, we analyze the security of the proposed encryption and signature schemes.

Keywords public key cryptography, information security, NP-complete problem, complexity theory, quantum computation

Citation Wu W Q, Zhang H G, Wang H Z, et al. A public key cryptosystem based on data complexity under quantum environment. Sci China Inf Sci, 2015, 58: 110102(11), doi: 10.1007/s11432-015-5408-5

1 Introduction

In the early 1980s, Feynman presented a quantum Turing machine (QTM) based on quantum mechanics. With the rapid development of quantum information science, a number of quantum algorithms have been discovered in the past three decades.

Deutsch was the first to present a model of quantum computation and to point out the superiority of quantum computation to conventional electronic computation [1]. Bernstein presented a mathematical model of a QTM [2]. This became the theoretical basis of quantum computation. Simon proposed an example of a quantum algorithm [3]. Grover constructed a general quantum search algorithm [4], with complexity $O(\sqrt{n})$. In application to cryptography, the Grover algorithm reduces the length of the key by half. This presents a threat to existing cryptosystems. Shor's seminal article [5] proposed an effective quantum algorithm to solve the factorization and discrete logarithm problems. This algorithm therefore

* Corresponding author (email: whz@whu.edu.cn)

presents a serious threat to classical public key cryptosystems such as RSA, ELGamal, and ECC, which are based on the factorization and discrete logarithm problems. More generally, Mosca extended the Shor algorithm to abelian groups [6]. The primary ingredient of these algorithms is the efficient solution of a hidden subgroup problem about certain abelian groups. Hallgren pointed out that weak Fourier sampling succeeds for a similar reason when H is a normal subgroup of a non-abelian group G [7].

In summary, these quantum algorithms provide a new theoretical basis and tools for cryptanalysis. As a consequence, there is a need for new public key cryptosystems to take account of the challenges posed by quantum computation. Such post-quantum computation cryptosystems currently include the following:

1. Quantum cryptography based on quantum physics [8–13];
2. DNA-based cryptography [14–16];
3. Cryptography based on mathematically hard problems [17–20].

This paper concentrates mainly on public key cryptosystems based on mathematically hard problems.

At present, computational complexity theory deals mainly with time complexity and space complexity. In addition to time and space resources, data also comprises an important computational resource.

Data complexity was first used for differential cryptanalysis of the DES block cipher in electronic computation [21]. It is known that PC machines can break the DES with $\leqslant 8$ rounds in a few minutes. However, the standard DES (16 rounds) cannot be broken [22], since this would need at least 2^{47} chosen plaintext pairs, and the data complexity is large [23].

Although data complexity can be used for cryptanalysis, we shall attempt to use it to design a public key cryptosystem under a quantum environment. We obtain the following results.

1. In the quantum standard oracle model, there are no polynomial-size quantum circuits to calculate all Boolean functions. On the basis of this, we present a definition of data complexity in quantum computers.

2. Based on the difficulty of NPC and data complexity, we present a public key cryptosystem and signature scheme.

3. We analyze the security of the public key cryptosystem and signature scheme.

The main body of the paper is organized as follows: In Section 2, we introduce the relevant background knowledge. In Section 3, we present the definition of data complexity under a quantum environment. In Section 4, we propose some hard problems and a quantum one-way function. In Section 5, we present a public key cryptosystem and analyze its security. In Section 6, we propose a signature scheme and analyze its security. Section 7 provides a summary.

2 Preliminaries

2.1 Quantum computation, quantum measurement, and realization

The most widely used model of quantum computation involves a series of quantum logic gates and quantum measurements [24–30]. There is another model of quantum computation that involves only quantum measurements [31,32], but in this paper we shall consider only the first model.

A qubit state is a vector in a two-dimensional complex vector space; i.e., a qubit can be randomly expressed in any superposition state of $|0\rangle$ and $|1\rangle$. If a quantum system has n qubits, then the storage space of an n-qubit quantum computer is of 2^n dimensions.

Quantum measurement leads to "wave packet collapse" in quantum physics. Measurement causes quantum states to degrade to classical states with a probability value. The results of collapse are affected by previous measurements, and this is an irreversible process.

In a specific quantum system, the quantum superposition states are not permanent and the lengths of time for which they persist are different for different quantum systems. In 2008, Nicolas Gisin [33] produced a photon quantum state with a storage time of 1 ms in a solid-state device. At the annual meeting of the American Physical Society in early March 2012, researchers at IBM reported significant progress in quantum computing devices with regard to storage.

One application of quantum computing devices is to simulate classical computers. In this regard, Matteo Mariantoni [34] successfully realized the von Neumann structure using a quantum circuit. However, the number of qubits was low. Although there are many obstacles to successful implementation of quantum computers, it is to be hoped that further developments in technology may overcome these, and we believe that quantum computers have great potential for further development.

2.2 Quantum complexity

Definition 1 ([2]). A QTM is defined by the triple $M = (\Sigma, Q, \delta)$, where $\Sigma = \{\Delta, 0, 1\}^n$ denotes a finite alphabet set with a blank symbol Δ; Q denotes a set of finite states including an initial state q_0 and final state $q_f \neq q_0$; δ denotes the quantum transition function, $\delta : Q \times \Sigma \times Q \times \Sigma \times \{L, R\} \to \mathbb{C}$, where $\mathbb{C}$ is the complex field.

Definition 2 ([35]). Let the function $f : \{0,1\}^m \to \{0,1\}^n$.

1. The standard quantum oracle: $U_f : |x\rangle|b\rangle \to |x\rangle|b \oplus f(x)\rangle$.

2. The Fourier quantum oracle: $P_f : |x\rangle|b\rangle \to \mathrm{e}^{\frac{2\pi i f(x)b}{2^n}}|x\rangle|b\rangle$, where x and b are respectively an m-qubit string and an n-qubit string.

The oracles U_f and P_f are equivalent [35]. For a QTM with an oracle, the computational power has the following limitations.

Theorem 1 ([19]). For any QTM $Q(A)$ for which $T(n) = O(2^{\frac{n}{2}})$ relative to the oracle with probability 1, BQTime($T(n)$) does not contain NP.

Theorem 2 ([19]). For any QTM $Q(B)$ for which $T(n) = O(2^{\frac{n}{3}})$ relative to the oracle with probability 1, BQTime($T(n)$) does not contain NP $\cap$ Co-NP.

3 Data complexity under a quantum environment

In addition to time and space resources, data is also an important computational resource. If any resources cannot satisfy calculational requirements, it will not be possible to effectively complete the process of calculation. We therefore introduce the notion of data complexity under a quantum environment, combining time and space complexity to design a public key cryptosystem.

Similar to the classical situation, the proposed data complexity consists of input data complexity and processing data complexity under a quantum environment. Input data complexity refers to the required input data for completing quantum algorithms. On the other hand, processing data complexity refers to the required data for running these quantum algorithms to handle the input data. For example, if the decomposed number is n-bit, then the input data complexity of the Shor algorithm is $O(n)$. From the description of a quantum circuit, factor decomposition requires $< cn^2(\log n)(\log\log n)$ quantum logic gates [32], where c is a constant. In n-qubit quantum circuits, each quantum gate has at most $O(2^n)$ states (or data). Thus, the processing data complexity of the Shor algorithm is at most $O(n^3 2^n)$ and so is not large.The input data complexity of the Grover algorithm is $O(\sqrt{N})$, where N denotes the number of data needed to search. When N tends to infinity, the Grover algorithm is no longer valid.

Before presenting the notion of data complexity, we first introduce some additional notation. Boolean functions are important in classical cryptography. In general, this uses Boolean circuits to describe the computational process of Boolean functions. In quantum computation, we can use quantum circuits (or quantum networks) for this task.

Quantum logic gates can be divided into single-qubit, two-qubit, and multi-qubit gates, according to the number of input qubits. Without loss of generality, some logic operations can be regarded as black boxes composed of groups of quantum logic gates.

The size of a quantum circuit refers to the number of quantum logic gates in the circuit. If two quantum circuits have the same input and output in each assignment, they are equivalent. The quantum circuit size is called minimum if there is no equivalent quantum circuit containing fewer quantum logic gates.

Let $N(n, S)$ denote the number of quantum circuits for computing different Boolean functions, which includes n qubits channels and S quantum logic gates.

Lemma 1. $N(n,S) < S(2^n)^S$ for all S.

Proof. Let a quantum computer be in a state on a system consisting of n qubits. Then, the quantum circuit has n quantum channels. A quantum logic gate has $C_n^1 + C_n^2 + \cdots + C_n^n \approx 2^n$ different cases in the quantum circuits, depending on the numbers of qubits as the input of quantum gates. Thus, S quantum logic gates have at most $(2^n)^S$ different combinations of quantum circuits. So, there are at most $S(2^n)^S$ quantum circuits, if the quantum circuit size is not greater than S, i.e., $N(n,S) < S(2^n)^S$.

Theorem 3. Let a quantum computer have n qubits. In the standard oracle model, there are no polynomial-size quantum circuits to compute Boolean functions.

Proof. Let the quantum circuit size be S and let there be n qubit quantum channels in the quantum circuits. By the above lemma, there are at most $S(2^n)^S$ quantum circuits, if the quantum circuit size is not greater than S. Let $S = \frac{2^n-n}{n}$; then $S(2^n)^S = \frac{2^n-n}{n} \cdot 2^{2^n-n} = \frac{2^n-n}{n2^n} \cdot 2^{2^n} \ll 2^{2^n}$. So, there are no polynomial-size quantum circuits to compute Boolean functions.

So far, we have failed to find any quantum circuits (or Boolean circuits) of specific Boolean functions that are super linear functions. However, Lemma 1 suggests that there are still many computational tasks that cannot be effectively completed by the quantum computer. Theorem 3 shows that quantum computation cannot complete the computational task if the data complexity is large ($\gg 2^n$). This can be used as a condition for post-quantum computation.

Theorem 3 provides good inspiration for designing cryptosystems. For this purpose, we select a function $f : F_p^n \to F_p^m$ whose input data complexity is large. We can then utilize it to construct a secure cryptosystem. Alternatively, we can consider some special algebraic constructions for designing a secure cryptosystem such that a quantum computer would need a massive amount of data to attack it.

Based on this theory, we introduce the notion of data complexity under a quantum environment in a standard quantum oracle.

Definition 3. Let QTM be a quantum Turing machine with n qubits. The data complexity of QTM is a function $f : N \to N$, where $f(n)$ is the sum of input data and processing data in QTM. Let $n = n_1 + n_2$; then $f(n) = f_1(n_1) + f_2(n_2)$, where n_1 and n_2 are respectively the numbers of storage registers and calculation registers. The functions $f_1(n_1), f_2(n_2) \in N$ are respectively the numbers of input data and processing data.

Definition 4. Let functions $f, g : N \to R^+$. If there exists a positive integer n_0 such that

$$\lim_{n\to\infty} \frac{f(n)}{g(n)} = c$$

for all $n > n_0$, where c is a positive constant, then $g(n)$ is an asymptotic upper bound of $f(n)$; i.e., $f(n) = O(g(n))$.

Definition 5. Let QTM be a quantum Turing machine with $n = n_1 + n_2$ qubits, where n_1 and n_2 are respectively the numbers of storage registers and calculation registers in the standard quantum oracle. Suppose that QTM runs a quantum algorithm to solve certain problems with effective time $T \in R^+$; then $f(n) = f_1(n_1) + f_2(n_2) \in N$ is the sum of input data and processing data, where the functions $f_1(n_1)$ and $4f_2(n_2) \in N$ are respectively the numbers of input data and processing data. Let the functions $g_1(n_1)$ and $g_2(n_2) \in N$ be respectively the capacities of input and processing data in the quantum computer.

1. If

$$\lim_{n\to\infty} \left(\frac{f_1(n_1)}{g_1(n_1)} + \frac{f_2(n_2)}{g_2(n_2)} \right) = k$$

for all $n > n_0$, where k is a positive constant, then the problem is easy to calculate in QTM.

2. If there exists a positive integer n_0 such that

$$\lim_{n\to\infty} \left(\frac{f_1(n_1)}{g_1(n_1)} + \frac{f_2(n_2)}{g_2(n_2)} \right) = \infty,$$

then the problem is hard to calculate in QTM.

Definitions 3 and 5 are abstract concepts, but the ability of a quantum computer to solve problems can be illustrated by considering specific quantum circuits [1–7,34,36].

4 Hard problems and quantum one-way functions

4.1 Hard problems

In this section, we first describe some NPC problems and NP problems. Second, we construct hard problems.

Theorem 4 (Tensor decomposition of rank r over any field F [37–39]). Let F be any field. Given t_{ijk} and a positive integer r, where $1 \leqslant i \leqslant n_1$, $1 \leqslant j \leqslant n_2$, $1 \leqslant k \leqslant n_3$, compute the vector $v_e^{(l)} \in F^{n_e}$, $1 \leqslant l \leqslant r$, $1 \leqslant e \leqslant 3$ such that $t_{ijk} = \sum_{l=1}^{r} v_1^l(i) v_2^l(j) v_3^l(k)$. This is an NPC problem.

Theorem 5 ([37–39]). The tensor decomposition of rank $r = 1$ is a NP problem over a finite field F.

This indicates that there is no effective quantum algorithm to solve tensor problems [37–39], even though rank $r = 1$ over a finite field.

Definition 6 (Subset problem [40]). Given a set of values $M_1, M_2, \ldots, M_n \in Z^+$ and a sum $S \in Z^+$, compute $b_i \in \{0, 1\}$ such that $S = b_1 M_1 + b_2 M_2 + \cdots + b_n M_n$.

Definition 7. Given n-dimensional vectors $A = (a_i)_{1\times n}$ and $B = (b_i)_{1\times n}$, the wreath product $\odot$ operation can be defined as follows:

$$A \odot B = (a_i b_i)_{1\times n},$$

where $a_i, b_i \in Z_p,\ p \in Z$.

Theorem 6 (Wreath product unique decomposition problem for vector spaces). Given an n-dimensional vector $C = A \odot B = (a_i b_i)_{1\times n}$, a set of values $M_1, \ldots, M_n \in Z^+$, and a sum $S \in Z^+$, then an n-dimensional vector $A = (a_i)_{1\times n}$ satisfying $S = a'_1 M_1 + \cdots + a'_n M_n$ can be obtained from the vector C, where $a_i, b_i \in Z_p$, p is prime, and $i = 1, \ldots, n$. The a'_i are such that if $a_i \geqslant b_i$, then $a'_i = 1$; otherwise, if $a_i < b_i$, then $a'_i = 0$.

Proof. It is easy to see that this problem is an extension of the subset problem. The subset problem is a special case of the vector space wreath product decomposition problem under the restrictions $a_i = 1,\ b_i = 0$ or $a_i = 0,\ b_i = 1$. Thus, the problem can be seen to be NPC directly from the subset problem.

Corollary 1 (Wreath product unique decomposition problem for matrices). Given an n^2-dimensional matrix $C = A \odot B = (a_{ij} b_{ij})_{n\times n}$, a set of values $M_1, \ldots, M_{n^2} \in Z^+$, and a sum $S \in Z^+$, then it is possible to compute an n^2-dimensional matrix $A = (a_{ij})_{n\times n}$ satisfying $S = a'_{11} M_1 + \cdots + a'_{nn} M_{n^2}$ from the matrix C, where $a_{ij}, b_{ij} \in Z_p$, p is prime, and $i, j = 1, \ldots, n$. The a'_{ij} are such that if $a_{ij} \geqslant b_{ij}$, then $a'_{ij} = 1$; otherwise, if $a_{ij} < b_{ij}$, then $a'_{ij} = 0$.

Note 1. The wreath product decomposition problem over smaller domains is still an NPC problem. For example, this is so even if the sets have only two elements a and b, $a \neq b$ (or $a = 0, b = 1$).

4.2 Quantum one-way function

In this section, we introduce the concept of a quantum one-way function. Let

$$\mathrm{QFP} = \{f \mid \text{The QTM takes quantum polynomial time to calculate } f\}.$$

Definition 8 (Quantum one-way function [17]). A function f is called quantum one-way (QOW) if the following two conditions hold:

1. [Easy to compute] There exists a polynomial time QTM A, so that, on input x, A outputs $f(x)$ (i.e., $A(x) = f(x)$).

2. [Hard to invert] For every probabilistic polynomial time QTM, Adv, every polynomial poly, and all sufficiently large n,

$$\Pr[\mathrm{Adv}(f(x)) \in f^{-1}(f(x))] < 1/\mathrm{poly}(n).$$

The probability is taken over the distribution of x, the (classical) coin flips of Adv, and quantum observation of Adv.

Definition 9 (Tensor operation). Let $x_1 = (a_0, a_1, \ldots, a_{n-1}),\ x_2 = (b_0, b_1, \ldots, b_{n-1})$, where $a_i,\ b_i$ are over the field F; then $x_1 \otimes x_2 = (a_0 b_0, a_0 b_1, \ldots, a_0 b_{n-1}, \ldots, a_{n-1} b_0, a_{n-1} b_1, \ldots,\ a_{n-1} b_{n-1})$.

Let A, B, C be n^2-dimensional matrices over the finite field F_q and M be a n^6-dimensional matrix over F_q, where q is a large prime. Let the function $f : F_q^{n^6} \to F_q^{n^6}$ satisfy $f(A, B, C) = A \otimes B \otimes C = M$. By Theorems 4 and 5, the function $f(A, B, C) \in \mathrm{QFP}, f^{-1}(A, B, C) \notin \mathrm{QFP}$. Thus, the function $f(A, B, C)$ can be regarded as a quantum one-way function.

5 Quantum public key cryptosystem

Before further discussion, we define some notation. Let the unit element be $I = (I_{ij})_{n\times n}$ satisfying $I \odot A = A$, where $I_{ij} = 1$. The symbol $M^e = M \odot M \odot \cdots \odot M,\ e \in Z^+$ represents the product of e matrices M under the $\odot$ operation. The following discussion concerns objects over a finite field.

5.1 Proposed scheme

Theorem 7. Let $f = (f_{ij})_{n\times n}, g = (g_{ij})_{n\times n}, \alpha = (\alpha_{ij})_{n\times n}, \beta = (\beta_{ij})_{n\times n},\ i, j = 1, \ldots, n$, where $f_{ij}, g_{ij}, \alpha_{ij}, \beta_{ij} \in Z_p$, p is prime. Then $(f \otimes g) \odot (\alpha \otimes \beta) = (f \odot \alpha) \otimes (g \odot \beta)$.

Proof. When $n = 1$, this is ordinary multiplication. Let $n > 1$. Then the left-hand side of the equation is

$$\begin{pmatrix} f_{11}g & \cdots & f_{1n}g \\ \cdots & \cdots & \cdots \\ f_{n1}g & \cdots & f_{nn}g \end{pmatrix} \odot \begin{pmatrix} \alpha_{11}\beta & \cdots & \alpha_{1n}\beta \\ \cdots & \cdots & \cdots \\ \alpha_{n1}\beta & \cdots & \alpha_{nn}\beta \end{pmatrix} = \begin{pmatrix} f_{11}g \odot \alpha_{11}\beta & \cdots & f_{1n}g \odot \alpha_{1n}\beta \\ \cdots & \cdots & \cdots \\ f_{n1}g \odot \alpha_{n1}\beta & \cdots & f_{nn}g \odot \alpha_{nn}\beta \end{pmatrix}$$
$$= \begin{pmatrix} (f_{11}\alpha_{11})g \odot \beta & \cdots & (f_{1n}\alpha_{1n})g \odot \beta \\ \cdots & \cdots & \cdots \\ (f_{n1}\alpha_{n1})g \odot \beta & \cdots & (f_{nn}\alpha_{nn})g \odot \beta \end{pmatrix}.$$

The right-hand side is

$$\begin{pmatrix} f_{11}\alpha_{11} & \cdots & f_{1n}\alpha_{1n} \\ \cdots & \cdots & \cdots \\ f_{n1}\alpha_{n1} & \cdots & f_{nn}\alpha_{nn} \end{pmatrix} \otimes \begin{pmatrix} g_{11}\beta_{11} & \cdots & g_{1n}\beta_{1n} \\ \cdots & \cdots & \cdots \\ g_{n1}\beta_{n1} & \cdots & g_{nn}\beta_{nn} \end{pmatrix} = \begin{pmatrix} (f_{11}\alpha_{11})g \odot \beta & \cdots & (f_{1n}\alpha_{1n})g \odot \beta \\ \cdots & \cdots & \cdots \\ (f_{n1}\alpha_{n1})g \odot \beta & \cdots & (f_{nn}\alpha_{nn})g \odot \beta \end{pmatrix}.$$

So the left- and right-hand sides are equal, since the commutative law is satisfied over Z_p.

Key generation

1. Select randomly a large prime $p > 2^m p_1^{r_1} \cdots p_s^{r_s}$, where $p_1, \ldots, p_s$ are odd primes and $r_1, \ldots, r_s \in Z^+$.

2. Select nonzero integers $t_1, t_2, t_3 \in \{0, 1, \ldots, \varphi(p)\}$ and $t_1 \neq t_2 \neq t_3$, where $\varphi(p)$ is the Euler function of p.

3. Select three m^2-dimensional matrices $A = (a_{ij})_{m\times m},\ B = (b_{ij})_{m\times m},\ D = (d_{ij})_{m\times m}$, where $a_{ij}, b_{ij}, d_{ij} \in Z_p$.

4. Compute $Y_1 = A^{t_1} \odot B^{t_2} \odot D^{t_3} \pmod p$, $Y_2 = B^{t_1} \odot D^{t_2} \odot A^{t_3} \pmod p$, $Y_3 = D^{t_1} \odot A^{t_2} \odot B^{t_3} \pmod p$ such that every $y_{ij}^{(1)}, y_{ij}^{(2)}, y_{ij}^{(3)} \geqslant 2^m$, where $Y_1 = (y_{ij}^{(1)})_{m\times m},\ Y_2 = (y_{ij}^{(2)})_{m\times m},\ Y_3 = (y_{ij}^{(3)})_{m\times m},\ y_{ij}^k \in Z_p, k = 1, 2, 3,\ i, j = 1, \ldots, m$. Otherwise it returns to the third step.

The public key comprises matrices A, B, D, Y_1, Y_2, Y_3 and a prime number p; the private key comprises numbers t_1, t_2, t_3.

Encryption

1. Let there be given an m^6-dimensional plaintext $M = (m_{ij})_{m^3\times m^3},\ m_{ij} \in Z_p$.

2. Select nonzero integers $s_1, s_2, s_3 \in \{0, 1, \ldots, \varphi(p)\}$ and $s_1 \neq s_2 \neq s_3$.

3. Compute $U = Y_1^{s_1} \otimes Y_2^{s_2} \otimes Y_3^{s_3} \pmod p$, $C_1 = A^{s_1} \otimes B^{s_2} \otimes D^{s_3} \pmod p$, $C_2 = B^{s_1} \otimes D^{s_2} \otimes A^{s_3} \pmod p$, $C_3 = D^{s_1} \otimes A^{s_2} \otimes B^{s_3} \pmod p$, $C = U \odot M \pmod p$.

The ciphertext is a four-tuple (C_1, C_2, C_3, C).

Decryption

1. Compute $V = C_1^{t_1} \odot C_2^{t_2} \odot C_3^{t_3} \pmod p$.

2. Recover plaintext M through computing m^6 congruence equations $v_{ij}m_{ij} = c_{ij} \pmod p$, where $v_{ij} \in V,\ c_{ij} \in C,\ m_{ij} \in M$.

5.2 Correctness of decryption

Let $U = (u_{ij})_{m\times m}$. During the process of decryption, the first step computes $V = C_1^{t_1} \odot C_2^{t_2} \odot C_3^{t_3}$ $=(A^{s_1t_1} \otimes B^{s_2t_1} \otimes D^{s_3t_1}) \odot (B^{s_1t_2} \otimes D^{s_2t_2} \otimes A^{s_3t_2}) \odot (D^{s_1t_3} \otimes A^{s_2t_3} \otimes B^{s_3t_3})=(A^{s_1t_1} \odot B^{s_1t_2} \odot D^{s_1t_3}) \otimes (A^{s_2t_3} \odot B^{s_2t_1} \odot D^{s_2t_2}) \otimes (A^{s_3t_2} \odot B^{s_3t_3} \odot D^{s_3t_1}) =Y_1^{s_1} \otimes Y_2^{s_2} \otimes Y_3^{s_3}=U \pmod p$ by Theorem 7. Then, the second step computes the congruence equations. It is known that the general solutions of the congruence equations are $x = x' + \frac{p}{d}t,\ t = 0, 1, \ldots, d-1$, where $d = (u_{ij}, p)$ and x' is a particular solution. As $(u_{ij}, p) = 1$, there is only a solution in the value space.

5.3 Security analysis of the scheme

We provide an initial analysis of the security of our scheme by considering several possible attack approaches. We assume that the eavesdropper *Eve* uses a quantum computer with m qubits to attack this scheme and that the computational capability of the quantum computer is $O(2^m)$, i.e., it can complete these $O(2^n)$ computations instantaneously. It is clear that our scheme is analogous to RSA public key cryptography. We believe that it is immune to classical attack. Thus, we only consider its security against a quantum attack, including the Shor and Grover algorithms.

1. Finding secret keys from public keys. *Eve* has only the public information in a passive attack. Let $\delta_p(a)$ be the smallest positive integer satisfying $a^x = 1 \pmod p$, where $(p, a) = 1$. In number theory, this is a primitive root modulo p, if p is an odd prime. Then $\delta_p(a) = \varphi(p)$ and $a \in Z_p$ is a primitive root.

Eve computes $Y_1 \odot Y_2 \odot Y_3=(A \odot B \odot D)^{t_1+t_2+t_3} \pmod p$ and obtains $t = t_1 + t_2 + t_3$ by the Shor algorithm. However, in the worst case, the period of t is $\varphi(p)$ and the possible decomposition result has $C^2_{\varphi(p)-4} = \frac{(\varphi(p)-4)(\varphi(p)-5)}{2} = \frac{(p-5)(p-6)}{2} \approx O(p^2) \geqslant O(2^{2m})$. So the input data complexity is not less than $O(2^{2m})$. By Definition 5,

$$\lim_{m\to\infty} \frac{2^{2m}}{2^m} = \infty.$$

Thus, it seems to be impossible for *Eve* to find the right t_1, t_2, t_3 from all solutions, even with a quantum computer.

In addition, the complexity of finding solutions is no less than $O(2^m)$ through the Grover algorithm, and it seems to be impossible to guess the solution, since the numbers have exponentially large possibilities.

In the other method, *Eve* can directly compute t_1, t_2, t_3 from $Y_1 = A^{t_1} \odot B^{t_2} \odot D^{t_3} \pmod p$. This is equivalent to solving the group of equations $a_{ij}^{t_1}b_{ij}^{t_2}d_{ij}^{t_3} = y_{ij} \pmod p$. Then, this procedure is equivalent to solving the following set of nonlinear equations using the quantum computer:

$$\begin{cases} a_{ij}^{t_1} = y_{ij}^{(1)} \pmod p, \\ b_{ij}^{t_2} = y_{ij}^{(2)} \pmod p, \\ d_{ij}^{t_3} = y_{ij}^{(3)} \pmod p, \\ y_{ij} = y_{ij}^{(1)}y_{ij}^{(2)}y_{ij}^{(3)}, \end{cases}$$

where t_1, t_2, t_3 are variables. Because $y_{ij} = 2^m p_1^{r_1'} \cdots p_s^{r_s'}$, the number of $y_{ij}^{(1)}, y_{ij}^{(2)}, y_{ij}^{(3)}$ exceeds $C_r^0 + C_r^1 + \cdots + C_r^r = 2^r$, where $r_1' \leqslant r_1, \ldots, r_s' \leqslant r_s$ and $r = m + r_1' + \cdots + r_s'$. Thus, the data complexity is not less than $O(2^r)$. Assuming that the quantum computer can be regarded as a black box with the most powerful computational ability, by Definition 5 it still cannot complete all the computations needed to solve the set of nonlinear equations. Thus, the private key is safe.

2. Finding plaintexts from ciphertexts. It appears to be impossible to guess the nonzero integers s_1, s_2, s_3, since the numbers have exponentially large possibilities for the same reasons as above. *Eve* has only the ciphertexts (C_1, C_2, C_3, C) in the passive attack. By the definition of tensor operations, $A^{s_1} \otimes B^{s_2} \otimes D^{s_3} \pmod p$ can be converted into $A^{s_1} \odot B^{s_2} \odot D^{s_3} \pmod p$ through selection of special

elements. Therefore, the data complexity is not less than $O(2^m)$, and the security is similar to that in the previous discussion.

The probability of the plaintext M from $C = U \odot M \pmod p$ is not more than $\frac{1}{2^{m^2}}$. It appears to be impossible to guess the m_{ij}. Thus, it is computationally secure with respect to quantum computation.

In addition, direct decomposition of the ciphertexts $C_i, i = 1, 2, 3$ and C is also an intractable task in smaller domains according to Theorems 5 and 6 and Note 1.

Therefore, the plaintext is safe.

6 Signature scheme

The above encryption scheme can be used to construct a digital signature. This is possible for the following reasons.

The receiver *Bob* selects nonzero integers s_1, s_2, s_3 and computes $C_1 = A^{s_1} \otimes B^{s_2} \otimes D^{s_3} \pmod p$, $C_2 = B^{s_1} \otimes D^{s_2} \otimes A^{s_3} \pmod p$, $C_3 = D^{s_1} \otimes A^{s_2} \otimes B^{s_3} \pmod p$. Then, *Bob* sends C_1, C_2, C_3 to the signer *Alice*.

Alice computes $V = C_1^{t_1} \odot C_2^{t_2} \odot C_3^{t_3} \pmod p = (v_{ij})_{m^3 \times m^3}$. *Alice* solves the congruence equation $u_{ij} m'_{ij} = m_{ij} \pmod p$ and obtains the new matrix $M' = (m'_{ij})_{m^3 \times m^3}$. So $D(M, K_d) = M'$, where K_d is a private key.

Bob computes $U = Y_1^{s_1} \otimes Y_2^{s_2} \otimes Y_3^{s_3} \pmod p$ and verifies $E(D(M, K_d), K_e) = U \odot M' = M \pmod p$, where K_e is a public key and M is plaintext. Thus,

$$E(D(M, k_d), k_e) = D(E(M, k_e), k_d) = M.$$

6.1 Signature scheme

The public key is $K_e = (A, B, D, Y_1, Y_2, Y_3)$ and the private key is $K_d = (t_1, t_2, t_3)$. Let M be plain text and HASH be any hash function. Our signature scheme has three participants: the signer *Alice*, the receiver *Bob*, and a justice center (JC). JC appears only in the case of a dispute. The signature scheme has three phases: Initialization, Signature, and Verification. The specific process is as follows.

Initialization:

JC selects three different nonzero integers $l_1, l_2, l_3 \in \{0, 1, \ldots, \varphi(p)\}$, and computes $\overline{C_1} = A^{l_1} \otimes B^{l_2} \otimes D^{l_3} \pmod p$, $\overline{C_2} = B^{s_1} \otimes D^{s_2} \otimes A^{s_3} \pmod p$, $\overline{C_3} = D^{s_1} \otimes A^{s_2} \otimes B^{s_3} \pmod p$. JC saves these data l_1, l_2, l_3. Then, *Alice* computes $\overline{V} = \overline{C_1}^{t_1} \odot \overline{C_2}^{t_1} \odot \overline{C_3}^{t_1} \pmod p$ and sends the $\overline{V}$ to JC through secure and reliable channels, while JC saves $\overline{V}$ and the public key $K_e = (A, B, D, Y_1, Y_2, Y_3)$.

Signature process:

1. *Bob* selects nonzero integers $s_1, s_2, s_3, r_1, r_2, r_3 \in \{0, 1, \ldots, \varphi(p)\}$, with $s_1 \neq s_2 \neq s_3$ and $r_1 \neq r_2 \neq r_3$. *Bob* computes $C_1 = A^{s_1} \otimes B^{s_2} \otimes D^{s_3} \pmod p$, $C_2 = B^{s_1} \otimes D^{s_2} \otimes A^{s_3} \pmod p$, $C_3 = D^{s_1} \otimes A^{s_2} \otimes B^{s_3} \pmod p$, $C'_1 = A^{r_1} \otimes B^{r_2} \otimes D^{r_3} \pmod p$, $C'_2 = B^{r_1} \otimes D^{r_2} \otimes A^{r_3} \pmod p$, $C'_3 = D^{r_1} \otimes A^{r_2} \otimes B^{r_3} \pmod p$. Then *Bob* sends $C_1, C_2, C_3, C'_1, C'_2, C'_3, T_{BA}$ to *Alice* and sends $s_1, s_2, s_3, r_1, r_2, r_3, T_{BA}$ to JC through secure and reliable channels, where T_{BA} is a time stamp in the digital signature scheme.

2. *Alice* signs the message M with the private key K_d. *Alice* selects three nonzero integers $s'_1, s'_2, s'_3 \in \{0, 1, \ldots, \varphi(p)\}$, and computes $U = Y_1^{s'_1} \otimes Y_2^{s'_2} \otimes Y_3^{s'_3} \pmod p$, $\overline{M} = M \odot U \pmod p$. Then, *Alice* sends the three nonzero integers s'_1, s'_2, s'_3, T_{AB} to JC through secure and reliable channels, where T_{AB} is a time stamp.

3. *Alice* computes $V = C_1^{t_1} \odot C_2^{t_2} \odot C_3^{t_3} \pmod p$, $W = (C'_1)^{t_1} \odot (C'_2)^{t_2} \odot (C'_3)^{t_3} \pmod p$. *Alice* solves the congruence equations and obtains M' and $\overline{M}'$ from $V \odot M' = \overline{M} \pmod p$, $W \odot \overline{M}' = M' \pmod p$. *Alice* computes $c = \mathrm{HASH}(\overline{M} \oplus M') \oplus \mathrm{HASH}(M' \oplus \overline{M}')$. So *Alice* sends the signature $\langle \overline{M}', \overline{M}, M, c, T_{AB} \rangle$ to *Bob*.

Verify signature:

Bob verifies the signature with the public key K_e. *Bob* computes $\overline{U}' = Y_1^{r_1} \otimes Y_2^{r_2} \otimes Y_3^{r_3} \pmod p$, $\overline{U}' \odot \overline{M}' = N' \pmod p$, $\overline{U} = Y_1^{s_1} \otimes Y_2^{s_2} \otimes Y_3^{s_3} \pmod p$ and $\overline{U} \odot N' = N \pmod p$. If $N = \overline{M}$ and

$r = \mathrm{HASH}(N \oplus N') \oplus \mathrm{HASH}(N' \oplus \overline{M}') = c$, then the signature is successful. Otherwise, the signature is false.

Note 2. The above tuple (s_1', s_2', s_3') is one-time.

6.2 Security of the signature

If a forger is to successfully forge the signature, they must obtain M'. Otherwise, the result is not equal to c in the HASH function. The forger does not know the matrices V and W, since t_1, t_2, t_3 are secret keys. The forger can obtain $V \odot W = \overline{M} \odot (\overline{M}')^{-1} \pmod p$. The decomposition results of V and W are large. Furthermore, the correct probability of M' satisfying the hash value c from $V \odot M' = \overline{M} \pmod p$ is not more than $\frac{1}{2^{m^2}}$. Even assuming that the quantum computer can be regarded as a black box with the most powerful computational ability, i.e., that it can complete an $O(2^m)$ computation of the hash function on a transient, by Definition 5 it still cannot complete all the computations of the hash value c. Therefore, the forger cannot easily forge the signature, and the latter is secure.

A dishonest legitimate receiver cannot effectively forge the signature. Since the random numbers (s_1', s_2', s_3') are not known, a dishonest receiver cannot obtain the correct U. The dishonest receiver can only use the previous U to forge a signature. However, it is possible to check whether the signature is fake through JC. The dishonest receiver offers the signature $\langle \overline{M}', \overline{M}, M, c, T_{AB} \rangle$. JC first finds (s_1', s_2', s_3') from the time stamp T_{AB}, and tests the correctness of U from $\overline{M} = M \odot U \pmod p$. Then, JC tests the correctness of the private key t_1, t_2, t_3 from $\overline{V} = \overline{C_1}^{t_1} \odot \overline{C_2}^{t_1} \odot \overline{C_3}^{t_1} \pmod p$. Third, JC obtains M' by computing $V = C_1^{t_1} \odot C_2^{t_2} \odot C_3^{t_3} \pmod p$ and solving the congruence equation $V \odot M' = \overline{M} \pmod p$. Finally, JC tests the correctness of $W \odot \overline{M}' = M' \pmod p$. If it is not correct, then this shows that the signature has been forged by the receiver.

In addition, the signer cannot deny the signature. If $W \odot \overline{M}' = M' \pmod p$ holds, the signature is true.

7 Summary

Existing classical public key cryptosystems (e.g., RSA, EIGamal, and ECC) face serious threats from recent advances in quantum computation. Under a quantum computational environment, it is still necessary to protect information security and it is an important task to determine which kinds of public key cryptosystems can be used under such environments.

Since quantum computation is a parallel approach, it has a large calculational capacity, and quantum computers can efficiently solve some problems that are believed to be intractable on any classical computer. In this sense, we have proved that there are no polynomial-size quantum circuits that can compute Boolean functions. We have proposed a notion of data complexity under a quantum environment and have shown that quantum computation cannot complete the computational task if the data complexity is large ($\gg 2^n$). This can be used as a condition for post-quantum computation.

Thus, we have utilized complexity theory, including data complexity, time complexity, and space complexity, to design a public key cryptosystem. In general, it is widely believed that NP-complete problems cannot be solved in polynomial time even with quantum computers [17,19,41,42]. We have therefore tried to construct a new quantum public key cryptosystem based on data complexity and the NPC problem. This scheme can not only encrypt, but also realize the signature. Finally, we have analyzed the security of the encryption and signature schemes. This concept of data complexity is a new research field, and there are still many problems that remain to be investigated.

Acknowledgements

This work was supported by National Natural Science Foundation of China (Grant Nos. 61303212, 61303024, 61170080), State Key Program of National Natural Science of China (Grant Nos. 61332019, U1135004), Major State Basic Research Development Program of China (Grant No. 2014CB340600), Foundation of Science and

Technology on Information Assurance Laboratory (Grant No. KJ-14-002), and Fundamental Research Funds for the Central Universities (Grant No. 2012211020213).

References

1 Deutsch D, Jozsa R. Rapid solution of problems by quantum computation. Math Phys Sci, 1992, 439: 553–558
2 Bernstein E, Vazirani U. Quantum complexity theory. SIAM J Comput, 1997, 26: 1411–1473
3 Simon D R. On the power of quantum computation. SIAM J Comput, 1997, 26: 1474–1483
4 Grover L K. Quantum mechanics helps in searching for a needle in haystack. Phys Rev Lett, 1997, 79: 325–328
5 Shor P W. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. SIAM J Comput, 1997, 26: 1484–1509
6 Mosca M, Ekert A. The hidden subgroup problem and eigenvalue estimation on a quantum computer. In: Proceedings of the 1st NASA International Conference on Quantum Computing and Quantum Communication. Berlin: Springer, 1999
7 Hallgren S, Russell A, Ta-Shma A. The hidden subgroup problem and quantum computation using group representations. SIAM J Comput, 2003, 32: 916–934
8 Bennett C H, Brassard G. Quantum cryptography:public key distribution and coin tossing. In: Proceedings of IEEE International Conference on Computers, Systems and Signal Processing, Bangalor, 1984. 10–12
9 Bennett C H, Brassard G, Crépeau C, et al. Teleporting an unknown quantum state via dual classical and Einstein-Podolsky-Rosen channels. Phys Rev Lett, 1993, 70: 1895–1899
10 Bennett C H, DiVincenzo D P, Smolin J A, et al. Mixed-state entanglement and quantum error correction. Phys Rev A, 1996, 54: 3824–3851
11 Leung D W. Quantum vernam cipher. Quantum Inf Comput, 2002, 2: 14–34
12 Shi J J, Shi R H, Guo Y, et al. Batch proxy quantum blind signature scheme. Sci China Inf Sci, 2013, 56: 052115
13 Xiao F Y, Chen H W. Construction of minimal trellises for quantum stabilizer codes. Sci China Inf Sci, 2013, 56: 012306
14 Gehani A, Labean T H, Reif J H. DNA-based cryptography. In: Proceedings of the 5th Annual Meeting on DNA Based Computers, Cambridge, 2003. 233–249
15 Lu M X, Lai X J, Xiao G Z, et al. A symmetric key cryptography with DNA technology. Sci China Ser F-Inf Sci, 2007, 50: 324–333
16 Lai X J, Lu M X, Qin L, et al. Asymmetric encryption and signature method with DNA technology. Sci China Inf Sci, 2010, 53: 506–514
17 Okamoto T, Tanaka K, Uchiyama S. Quantum public-key cryptosystems. In: Proceedings of 20th Annual International Cryptology Conference, Santa Barbara, 2000. 147–165
18 Bernstein D J, Buchmann J, Dahmen E. Post-quantum Cryptography. Berlin: Springer, 2000
19 Wang H Z, Zhang H G, Wang Z Y, et al. Extended multivariate public key cryptosystems with secure encryption function. Sci China Inf Sci, 2011, 54: 1161–1171
20 Mu L W, Liu X C, Liang C L. Improved construction of LDPC convolutional codes with semi-random parity-check matrices. Sci China Inf Sci, 2014, 57: 022304
21 Biham E, Shamir A. Differential cryptanalysis of DES-like cryptosystems. J Cryptol, 1991, 4: 3–72
22 Biham E, Shamir A. Differential cryptanalysis of the full 16-round DES. In: Proceedings of the 12th Annual International Cryptology Conference, Santa Barbara, 1993. 487–496
23 Feng D G. Cryptanalysis (in Chinese). Beijing: Tsinghua University Press, 2000
24 Bennett C H, Brassard G, Vazirani U, et al. Strengths and weaknesses of quantum computing. SIAM J Comput, 1997, 26: 1510–1523
25 Sleator T, Weinfurter H. Realizable universal quantum logic gates. Phys Rev Lett, 1995, 74: 4087–4090
26 Barenco A, Deutsch D, Ekert A, et al. Conditional quantum dynamics and logic gates. Phys Rev Lett, 1995, 74: 4083–4086
27 Monroe C, Meekhof D M, King B E, et al. Demonstration of a fundamental quantum logic gate. Phys Rev Lett, 1995, 75: 4714–4717
28 Vedral V V, Barenco A, Ekert A. Quantum networks for elementary arithmetic operations. Phys Rev A, 1996, 54: 147–153
29 Beckman D, Chari A N, Devabhaktuni S, et al. Efficient networks for quantum factoring. Phys Rev A, 1996, 54: 1034–1063
30 Christof Z. Fast versions of Shor's quantum factoring algorithm. arXiv: quant-ph/9806084
31 Parker S, Plenio M B. Efficient factorization with a single pure qubit and $\log N$ mixed qubits. Phys Rev Lett, 2000, 85: 3049–3052
32 Susan L. Protecting Information: from Classical Error Correction to Quantum Cryptograph. Cambridge: Cambridge University Press, 2006

33 de Riedmatten H, Afzelius M, Staudt M U, et al. A solid-state light-matter interface at the single-photon level. Nature, 2008, 456: 773–777
34 Mariantoni M, Wang H, Yamamoto T, et al. Implementing the quantum von Neumann architecture with superconducting circuits. Science, 2011, 334: 61–65
35 Kashefi E, Kent A, Vedral V, et al. Comparison of quantum oracles. Phys Rev A, 2002, 65: 050304
36 Nielsen M A, Chuang I L. Quantum Computation and Quantum Information. Cambridge: Cambridge University Press, 2010
37 Hastad J. Tensor rank is NP-complete. J Algorithms, 1990, 11: 644–654
38 Hillar C J, Lim L-H. Most tensor problems are NP hard. J ACM, 2013, 60: 45
39 Mao S, Zhang H G, Wu W Q, et al. A resistant quantum key exchange protocol and its corresponding encryption scheme. China Commun, 2014, 11: 124–134
40 Schneier B. Applied Cryptography: Protocols, Algorithms, and Source Code in C. New York: Wiley, 1996
41 Wu W Q, Zhang H G, Mao S W, et al. Quantum algorithm to find invariant linear structure of MD hash functions. Quantum Inf Process, 2015, 14: 813–829
42 Wu W Q, Zhang H G, Wang H Z, et al. Polynomial-time quantum algorithms for finding the linear structures of Boolean function. Quantum Inf Process, 2015, 14: 1215–1226