2014 Tenth International Conference on Computational Intelligence and Security

Quantum Algorithm for Polynomial Root Finding Problem

Guodong Sun, Shenghui Su
College of Computer Science
Beijing University of Technology
Beijing, China
sgd-150@163.com

Maozhi Xu
School of Mathematics Sciences
Peking University
Beijing, China
Mzxu@pku.com

Abstract—**Quantum computation is a new computing model based on fundamental quantum mechanical principle. Grover's algorithm finds the solution for a searching problem in the square root time of exhaustive search. Brassard, Hoyer, Tapp's algorithm counts the number of solutions for a searching problem. Through exploiting the two quantum algorithms, we propose a quantum algorithm for solving a new cryptography problem----polynomial root finding problem, which could be used to design a cryptosystem. The algorithm will take O($\sqrt{M/t}$) steps for finding one of the *t* solutions to the problem, where *M* is the modular of the equation. The success rate of the algorithm is a constant and the cost of the algorithm depends on the calculations of modular exponentiation and the number of iterations.**

Keywords-Polynomial root finding problem; Quantum searching; Quantum counting; Signature algorithm

I. INTRODUCTION

In the past few years, quantum computation [1] based on fundamental quantum mechanical principle has attracted numerous attractions of people all over the world. The concept of quantum computation was firstly put forward by R.Feynman [2]. In early 1980s, he pointed out that computation in general could be done more efficiently if it made use of these quantum physical phenomena. After that, the first great quantum algorithm was proposed by Shor. In 1994, Shor [3, 4] invented a polynomial time quantum mechanical algorithm for solving the factorization problem and discrete logarithm problem which were and still are widely believed to have no polynomial solution on a classical computer. The results can be used to break the widely used RSA cryptosystem [5] and Diffie-Hellman key-exchange protocol. However, Shor's algorithm is not a universal algorithm and it can only solve a few computing problem. After the pioneering work of Shor, in 1996, Grover [6, 7] designed another quantum mechanical algorithm for searching a marker element in an unsorted database, providing a square root speedup for exhaustive search over classical algorithms. In the same year, Brassard, Hoyer and Tapp [8, 9] presented the idea of quantum counting which combined Shor's factoring algorithm and Grover's searching algorithm. These two quantum algorithms have been widely used to many cryptography applications, such as the cryptanalysis of block ciphers [10] and the exhaustive attack of DES [11].

Nowadays, with the development of the cryptography technology such as encryption, key distribution, digital signatures, and one-way functions, there are many applications such as secure encryption, signing contracts, and electronic voting in the information society. As is known to all, the security of these activities is guaranteed by public-key cryptography [12]. A great deal of public-key cryptosystems are believed to be secure against the classical computer, such as RSA and ECC [13]. However, in the case of a quantum computer is used, these cryptosystems are thoroughly unsecure due to the invention of Shor's algorithm. Besides, the strength of ciphers is challenged in virtue of Grover's searching algorithm. In order to confront the challenge of the quantum computer in the future, we must find as more quantum-resistant public-key cryptosystems as possible to ensure the information security in the quantum computer world.

In this paper, a new cryptographic problem named polynomial root finding problem, of which the hardness is the basis of the signature algorithm for the public-key cryptosystem REESSE1+ [14], is considered under the attack of a quantum computer. Specifically, we try to solve the polynomial root finding problem with the help of Grover's quantum searching algorithm and Brassard, Hoyer and Tapp's quantum counting algorithm.

Throughout the paper, unless otherwise specified, all the variables and constants are positive integers. $n \geq 80$ is the length of cipher, the sign % represents 'modulo', M is a large prime and $\bar{M}$ denotes M-1, lg x means the logarithm of x to the base 2, $|x|$ denotes the absolute value of an element x.

II. DESCRIPTION OF THE PROBLEM

In this section, we give some definitions of problems which are relevant to each other. These definitions may not have be used to a special schema, but they are potential candidates to design a public-key encryption scheme or a signature schema.

Definition 1: Given $a \neq 0, 1$, $|b| + |c| \neq 0$, and $d \neq 0$, solving $ax^n + bx^{n-1} + cx + d \equiv 0(\% \ \bar{M})$ for x, $x \in [1, \bar{M}]$, is called the polynomial root finding problem, shortly PRFP.

From definition 1, we know that the variables a and d cannot be zero, b and c cannot be zero simultaneously. Thus, the PRFP is a special form of a n-degree polynomial that the relevant coefficients satisfy some conditions. In fact, the hardness of PRFP is the basis of the security of the famous public-key cryptosystem REESSE1+. In

978-1-4799-7434-4/14 $31.00 © 2014 IEEE
DOI 10.1109/CIS.2014.40

REESSE1+, the constant n is the size of a plaintext, it is usually chosen as 80, 96, 112, or 128 with lg $M \approx 696$, 864, 1030, or 1216. The difficulty of PRFP and TLP(Transcendental Logarithm Problem) [15] both guarantee the security of the signature algorithm for REESSE1+.

As for the congruence $ax^n + bx^{n-1} + cx + d \equiv 0(\% \bar{M})$, if both b and c are equal to zero, and $d \neq 0$, it becomes a new problem which is slightly easier than the PRFP.

Definition 2**:** Given $a \neq 0$ and $d \neq 0$, solving $ax^n + d \equiv 0(\% \bar{M})$ for x, $x \in [1, \bar{M}]$, is called the rooting finding problem, shortly RFP [16].

According to polynomial time Turing reduction [17], the difficulty of PRFP is not easier than that of PRFP since RFP can be easily solved when we use an oracle to solving the PRFP. However, when the reduction is in the opposite direction, the same result would not happen. Actually, the RFP has a close relation with the DLP(Discrete Logarithm Problem) in the computational number theory. The DLP problem is a hard problem since there is no efficient algorithm for it. However, when M is a prime, there are valid probabilistic and deterministic algorithms for RFP. The probabilistic algorithm is employed for any random solution and the deterministic is employed only for the trivial solution [18]. But when the factorization of M is not known, the RFP is also a really hard problem due to that we have not found a polynomial time algorithm in a classical computer. In an unpublished paper, we have designed two polynomial quantum mechanical algorithms for RFP [19].

When $n=2$, the RFP then become a famous problem---- Quadratic Residue Problem [20], which is the basis of the Rabin schema and N-S digital signature protocol.

III. Quantum Searching and Counting Algorithms

A. *Quantum Searching Algorithm*

In 1996, Grover invented the quantum algorithm for searching the solution of a problem. Assume that there are $N=2^n$ elements and F be a Boolean function of the set of these N elements. Suppose there be a unique element α that satisfies $F(\alpha)=1$, and the elements are represented by n bit binary strings. To identify the element α, the algorithm consists of the following steps:

1) Initialization: Prepare two quantum registers, initialize them as $|0>$ and $|0>^{\otimes n}$ respectively. Applying the Walsh-Hadamard transform [20] defined as $H = 1/\sqrt{2}\,(-1)^{ij}|i>|j>$ to each qubit of the initial state, where i, j=0,1. Then the following superposition is obtained

$$|\varphi> = 1/\sqrt{N} \sum_{j=0}^{N-1} |j>$$

2) Iteration: Repeat the following unitary operation O($\sqrt{N}$) times.

a) Apply the unitary operator $U_x^{\oplus}$ to the first and the second registers, where $U_x^{\oplus}$: $|x>\otimes|y> \rightarrow |x>\otimes|y \oplus f(x)>$, where $\oplus$ is the addition modulo 2.

b) Use the diffusion transform D which is defined as below:

$$D_{ij}=2/N, \text{ if } i \neq j;\ D_{ij}=-1+2/N.$$

The transform D can be constructed by $D = HRH$, where H is the Walsh-Hadamard transformation and R the rotation matrix is defined as follows:

$$R_{ij}=0 \text{ if } i \neq j;\ R_{ij}=1 \text{ if } i=j=0;\ R_{ij}=-1 \text{ if } i=j \neq 0.$$

3) Measurement: Measure the resulting state and the state α is observed with a probability determined by the amplitudes.

Step a and b of 2) is called a G iteration. It is proved that after m iterations of G iteration, the system is in the superposition

$$\sin(2m+1)\theta\,|\alpha> + \cos(2m+1)\theta\,|\beta>, \qquad (1)$$

where the angle θ satisfies $\sin\theta = 1/\sqrt{N}$ [8] and $|\beta>$ is the superposition of the states other than $|\alpha>$. Therefore, the probability that we observe the state $|\alpha>$ is $|\sin(2m+1)\theta|^2$. Through a simple calculation, we can easily see that $O(\sqrt{N})$ iterations are needed to observe the state $|\alpha>$ with a probability at least 1/2.

In fact, when the number of solutions is more than one, namely there are multiple solutions, we need iterate the process 2) $O(\sqrt{N/t})$ times with a probability of at least 1/2 to find one of the t solutions (see [8]).

B. *Quantum Counting Algorithm*

The quantum algorithm for counting problem was proposed by Brassard, Hoyer, and Tapp soon after Grover invented his searching algorithm. The counting problem is defined as: given a Boolean function F defined on a finite set $X = \{0,1,\ldots, N\text{-}1\}$, finding or estimating the number t, where $t = | F^{-1}(1) |$. The quantum counting algorithm is a combination of Shor's factoring algorithm and Grover's searching algorithm. The rough idea of the quantum counting algorithm can be described as follow. Since the amplitude for the desired states have a period depending upon the number t of the desired states, Grover's algorithm is used to amplify the amplitude firstly, and then Shor's algorithm is applied to compute the period. Finally, the number t can be calculated through this period. In general, the quantum algorithm for the counting problem needs O($\sqrt{N}$) oracle queries to F in order to estimate t, while the classical algorithm needs O(N) oracle queries to F on average.

Let t denote the number of desired states and P the number of trials of the G iterations, $t \leq N/2$. Let t' be the final result of the quantum counting algorithm when we try P times of G iterations($P \geq 4$), then error range of t is given by the following inequality:

$$|t - t'| < 2\pi / P\sqrt{tN} + N\pi^2/P^2 \qquad (2)$$

with success rate about $8/\pi^2$[9].

IV. The Quantum Algorithm for PRFP

In this section, we give the quantum attack algorithm for PRFP using the above quantum searching algorithm and counting algorithm. The algorithm consists of two steps, the first step is to count the number of solutions for PRFP $ax^n + bx^{n-1} + cx + d \equiv 0(\% \bar{M})$, and second step is to search a solution for PRFP.

A. The Outline of the Algorithm

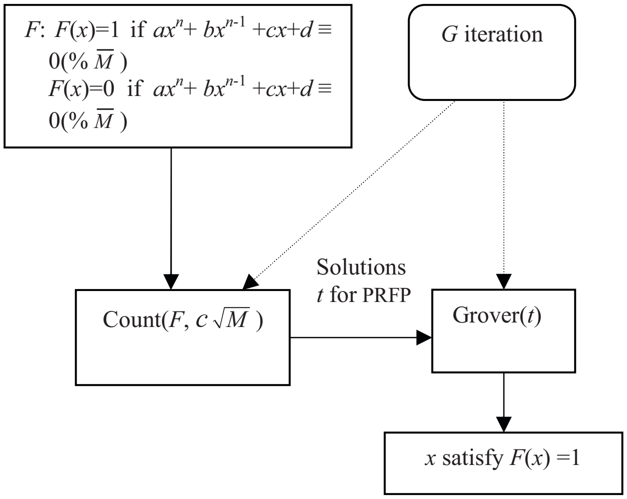
Obviously, when Grover's searching algorithm for multiple solutions is directly applied to solve the equation $ax^n + bx^{n-1} + cx + d \equiv 0(\% \overline{M})$, it seems very blindly. As is known, the success rate of Grover's algorithm does not monotonously increase with the number of G iteration. There exists a maximum point which is determined by the number of solutions for the problem, this point is the optimum value for the G iteration which can make the success rate of algorithm maximal. As for the PRFP equation $ax^n + bx^{n-1} + cx + d \equiv 0(\% \overline{M})$, it usually has more than one solution, and in some cases there may be no solution. So when we use Grover's algorithm, the number of solutions must be identified ahead of time. Otherwise, the success rate of the algorithm may be vanishingly small.

According to the above analysis, we propose a algorithm which is a combination of quantum counting algorithm and Grover's algorithm to solve the equation $ax^n + bx^{n-1} + cx + d \equiv 0(\% \overline{M})$. The idea of the algorithm can be describe as: we firstly count the number of solutions for the equation using quantum counting algorithm. Then Grover's searching algorithm is applied to search a solution for the problem. The detail of the algorithm is described as below:

1) Run the quantum counting algorithm. If the algorithm output zero, run the algorithm for a number of times, if the output is still zero, then $t = 0$. That is to say, the equation $ax^n + bx^{n-1} + cx + d \equiv 0(\% \overline{M})$ has no solution; if not so, execute the following step *2)*;

2) Set the number of G iterations according to step *1)*, execute the quantum searching algorithm and one solution for the equation is obtained.

In the light of above analysis, for clarity the outline of the algorithm is described in the "Fig.1".



F: $F(x)$=1 if $ax^n + bx^{n-1} + cx + d \equiv 0(\% \overline{M})$
$F(x)$=0 if $ax^n + bx^{n-1} + cx + d \equiv 0(\% \overline{M})$
G iteration
Solutions
t for PRFP
Count(F, $c\sqrt{M}$)
Grover(t)
x satisfy $F(x)$ =1

Figure 1. The outline of the algorithm for PRFP.

In Figure 1, the function F is the quantum oracle to identify the solution for the counting algorithm. The oracle is used to identify the solution for the equation $ax^n + bx^{n-1} + cx + d \equiv 0(\% \overline{M})$. The parameter P of quantum counting algorithm is set as $c\sqrt{M}$, where c be a constant. By the execution of quantum counting algorithm, the number t for the equation is identified, then the number of G iteration for Grover's algorithm of multiple solutions is set by t. Finally, a solution is obtained by the quantum searching algorithm. From the figure, we can see that G iteration is used both in quantum counting and searching algorithm.

B. The detail of the algorithm

The algorithm consists of two steps: Counting the number t and searching a solution x. We will not give the counting part of the algorithm. The detail of the counting algorithm can be seen in Brassard, Hoyer and Tapp's paper. In our application, the counting algorithm is only used as a mechanical execution, with proper object function F and parameter P. In the following, we just give the definition of the oracle and the detail of searching a solution for PRFP.

1) The design of the oracle

The quantum algorithm for PRFP is an algorithm based on a black-box oracle. The oracle is used to identify the solution for PRFP. In fact, the oracle is a boolean function F which is defined on the domain $[1, \overline{M}]$. The detail of the oracle for PRFP is defined as below.

$$F(x) = 1 \text{ if } ax^n + bx^{n-1} + cx + d \equiv 0(\% \overline{M})$$
$$\text{or } F(x) = 0 \text{ if } ax^n + bx^{n-1} + cx + d \not\equiv 0(\% \overline{M}).$$

Through the verification of the oracle, the solution for PRFP can be found. We will not consider the detail of the function F since it is easy to identify a solution for PRFP in the classical circuit model. In fact, in the quantum algorithm, the oracle is taken as a quantum operator U.

2) Searching a solution for PRFP

We now mount the attack using Grover's algorithm. For a concrete a, b, c and d, which satisfy the condition $a \neq 0, 1$, $|b| + |c| \neq 0$, and $d \neq 0$, we will find $x \in [1, \overline{M}]$ satisfy the equation $ax^n + bx^{n-1} + cx + d \equiv 0(\% \overline{M})$. Suppose t is the number of solutions in the counting algorithm, the following procedure finds one of the t solutions to the equation $ax^n + bx^{n-1} + cx + d \equiv 0(\% \overline{M})$.

a) Initialization.

Prepare two quantum registers, initialize the first register by a superposition of all possible x ($x \in [1, \overline{M}]$) with the same amplitude, that is to say, the first register is in the following state.

$$|X> = 1/\sqrt{2^q} \sum_{j=0}^{2^q-1} |x>$$

where integer q satisfy $M \leq 2^q \leq M^2$.

The second register is initialized as $|0>$.

b) Using the operator U.

In the next, apply the operator U to the system which is defined in the last section. The effect of the operator U is: $U|x>\otimes|y> \rightarrow |x>\otimes|y \oplus F(x)>$.

It is known that the second register $|y>$ is set as $|0>$. After we use the operator U, an XOR operation is occurred to the second register. Therefore, if $F(x) = 1$, the second register change its state from |0> to |1>, otherwise unchanged.

c) Take G iterations.

Note that t is the number of solutions for the equation $ax^n + bx^{n-1} + cx + d \equiv 0(\% \bar{M})$. According to section 3.1, we need O($\sqrt{2^q/t}$) iterations (Step *a* and *b* of *2)* in Grover's algorithm).

d) Measurement.

Measuring the system and we can observe one of the desired states with a probability at least 1/2. The observed state is the solution for $ax^n + bx^{n-1} + cx + d \equiv 0(\% \bar{M})$.

V. Analysis of the Algorithm

In the above section, we proposed a quantum mechanical algorithm for solving PRFP. In this section, we evaluate the algorithm exactly and give its running time and success rate.

A. The Success Rate

The success rate of the algorithm depends on the probability of counting the number of solutions and searching one of the solutions for PRFP

In the counting algorithm, the parameter P determines the success rate according to (2). No mater what the value of P, the success rate is at least $8/\pi^2$. So the success rate is mainly determined by the searching algorithm. Using the result from the counting algorithm, the roughly number of iterations is confirmed. Executing the searching algorithm, we can obtain a solution at a probability at least 1/2. The product of the two is the total success rate $4/\pi^2$. In the following, we analyze the success rate of the searching process exactly.

Commonly, the number of solutions for $ax^n + bx^{n-1} + cx + d \equiv 0(\% \bar{M})$ is no more than a small constant. For convenience, we suppose there are $t = 2^4$ solutions and lg$M$ ≈1024, take G iterations as $\sqrt{2^{1024}/2^4} = 2^{510}$. According to (1), after 2^{510} iterations, there is

$$
\begin{aligned}
& G^{2^{512}}|X> \\
& \rightarrow \sin(2^{511}+1)\theta\,|\alpha> + \cos(2^{511}+1)\theta\,|\beta> \\
& \approx \sin((2^{511}+1)/2^{510})|\alpha> + \cos((2^{511}+1)/2^{510})|\beta> \\
& \approx 0.909|\alpha> - 0.416|\beta>
\end{aligned}
$$

It is well known that when M is large enough, sin θ can be approximated as θ, and hence there is

$$\theta \approx \sin\theta = 1/\sqrt{M} = 1/2^{512}$$

Since the probability is the square of the amplitude, the desired state, namely one of the k solutions is obtained with 82.6 percent.

B. The Time Complexity

The main cost of the algorithm depends on the calculations of modular exponentiation and the number of G iterations. The modular exponentiation operation is: given n, x, and r, find x^r(mod n). The best classical method for doing this is to repeatedly square of x (mod n) to get x^{2^i} (mod n) for $i \leq$ lg r, and then multiply a subset of these powers (mod n) to get x^r(mod n). If x is a l-bit numbers, this requires O(l) squaring and multiplications of l-bit munbers (mod n). Actually, the best method for gate arrays for multiplication is the Schönhage-Strassen algorithm [21], which needs $O(k \lg k \lg \lg k)$ gates to multiply two k-bit integers. Thus, asymptotically, modular exponentiation requires O(l^2 lg$l$ lg lg$l$) time. Making this reversible to suit the quantum algorithm would cost the same amount of quantum gates in space.

In the algorithm counting part of the algorithm, according to (2), The running time of the algorithm is proportional to P. Take $P = c\sqrt{M}$, where c is a constant, the deviation range will not exceed $2\pi/c\sqrt{t} + \pi^2/c^2$. Then in the counting part of the algorithm we need about P iterations, namely O($\sqrt{M}$) iterations are needed. In the quantum searching part of the algorithm, about O($\sqrt{M/t}$)iterations are needed. Therefore, the total number of G iterations is O($\sqrt{M}$).

Finally, we evaluate the cost of the algorithm for PRFP in the REESSE1+. Recall that the probability that we observe the desired state is given by the square of the absolute value of the amplitude for α in (1). Solving the equality

$$\square\sin(2m+1)\theta\,| \geq 1/\sqrt{2},\ 0 \leq \theta \leq \pi/2,$$

we get $m \geq \pi 2^{q/2-3}$, where $q = $ lg$M$.

We know that in the REESSE1+, for the equation $ax^n + bx^{n-1} + cx + d \equiv 0(\% \bar{M})$, n is usually chosen as 80, 96, 112 or 128 with lg $M \approx$ 696, 864, 1030, or 1216, then q = 1024, 1024, 2048 or 2048 accordingly. We give the exact time of iterations in Table 1 when n = 80, 96, 112, or 128.

TABLE I. Exact Times of Iterations

n(bit)	lg$M$(bit)	2^q	m
80	696	2^{1024}	$\approx\pi 2^{509}$
96	864	2^{1024}	$\approx\pi 2^{509}$
112	1030	2^{2048}	$\approx\pi 2^{1021}$
128	1216	2^{2048}	$\approx\pi 2^{1021}$

From Table 1, we know that in the REESSE1+ public-key cryptosystem, the time complexity of the quantum algorithm to solve the equation $ax^n + bx^{n-1} + cx + d \equiv 0(\% \bar{M})$ is growing with the size of the modular. The bit length of the modular is greater; the time consumption of the quantum algorithm to the equation is longer. For example, we need iterate $m = \pi 2^{509}$ times in the case of lgM = 696, while $\pi 2^{1021}$ times are needed when lgM = 1030. This is mainly because the range of x is changed with the modular M. Normally there is not a direct relation between the parameter n and the times m of iterations. The parameter n influences the number of solutions for the equation, and further the times of iterations. Besides, we can easily notice that the time complexity of the quantum algorithm is roughly the square root of the time complexity of the brute force attack. For instance, when lg$M$ =696, the time complexity of the quantum algorithm is O($\pi 2^{509}$), while the brute force attack needs O($\pi 2^{1024}$). So, our proposed quantum algorithm for PRFP is a sub-exponential algorithm on the parameter lg$M$.

VI. CONCLUSION

In this paper, we proposed a quantum mechanical algorithm for solving the polynomial root finding problem (PRFP). So far, we have not found an effective classical algorithm to solving the PRFP. Since the root finding problem(RFP) can be reduced to the PRFP, it is easily to understand that this algorithm can be used to solve the RFP. The PRFP is a really hard problem which could be used to design a public-key encryption system or a digital signature schema. Our algorithm runs in sub-exponential time complexity and with a nearly constant success rate. Our result indicates that even when an effective quantum is designed, the schema based on PRFP is still secure if the parameters is selected large enough.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for their help and advice. This work is supported by MOST with project 2007CB311100 and 2009AA01Z441.

REFERENCES

[1] P. Xia, "Quantum Computing," Journal of Computer Research and Development, vol. 38, Oct. 2001, pp. 1153-1171.

[2] R. P. Feynman, "Simulating Physics with Computers," International Journal of Theoretical Physics, vol. 21, 1982, pp. 467-488.

[3] P. W. Shor, "Algorithms for quantum computation: Discrete logarithms and factoring," Proc. of the 35th Annual Symp on Foundations of Computer Science. New Mexico, IEEE Computer Society Press, 1994, pp. 124-134.

[4] P. W. Shor, "Polynomial-time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer," SIAM Journal on Computing, vol. 26, 1997, pp. 1484-1509.

[5] R. L. Rivest, A. Shamir, and L. M. Adleman, "A Method for Obtaining Digital Signatures and Public-key Cryptosystems," Communications of the ACM, vol. 21, 1978, pp. 120-126.

[6] L. K. Grover, "A Fast Quantum Mechanical Algorithm for Database Search," Proceeding of 28th ACM Symposium on Theory of Computation(STOC'96), New York, ACM Press, 1996, pp. 212-219.

[7] L. K. Grover, "Quantum mechanics helps in searching for a needle in a haystack," Phys. Rev. Lett., vol. 79, 1997, pp. 323.

[8] M.Boyer, G.Brassard, P.Hoyer, and A.Tapp, "Tight bounds on quantum searching," PhysComp'96, 1996, pp. 36-43.

[9] G.Brassard, P.Hoyer, and A.Tapp, "Quantum Counting," Int.Coll. Automata, Language and Programming(ICALP'98), LNCS 1443, 1998, pp. 820-831.

[10] Y. Akihiro, and I. Hirokazu, "Quantum Cryptanalysis of Block Ciphers," vol. 1166, 2000, pp. 235-243.

[11] H. D. Phaneendra, R. C. Vidya, and M. S. Shivakumar, "Applying Quantum Search to Known-Plaintext Attack on Two-key Triple Encryption," Intelligent Information Processing III, vol. 228, 2006, pp. 171-178.

[12] W. Diffie, and M. E. Hellman, "New Directions in Cryptography," IEEE Transaction of Information Theory, vol. 22, 1976, pp. 644-654.

[13] I. F. Blake, G. Seroussi, and N. P. Smart, Elliptic Curves in Cryptography, Cambridge University Press, Cambridge, UK, 1999.

[14] S. Su, and S. Lü, "A Public Key Cryptosystem Based on Three New Provable Problems," Theoretical Computer Science, vol. 426-427, Apr. 2012, pp. 91-117.

[15] S. Su, S. Lü, and X. Fan, "Asymptotic Granularity Reduction and Its Application," Theoretical Computer Science, vol. 412, Sep. 2011, pp. 5274-5386.

[16] S. Y. Yan, Number Theory for Computing, 2nd ed., Springer-Verlag, New York, 2002, pp. 225-228.

[17] A. Menezes, P.V. Oorschot, and S. Vanstone, "Handbook of Applied Cryptography," CRC Press, London, UK, 1997.

[18] S. Su, and S. Lü, "To Solve the High Degree Congruence x ^ n ≡ a (mod p) in GF(p)," Proc. of Int. Conference on Computation Intelligence and Security, IEEE press, 2007, pp. 672-676.

[19] G. Sun, S. Su, and M. Xu, "Quantum Mechanical Algorithms for Solving Root Finding Problem," unpublished.

[20] H. Wang, S. Zhang, Y. Zhao, K. Yeon, "Quantum Mechanical Algorithm for Solving Quadratic Residue Equation," Int J Theor Phys, vol. 48, 2009, pp. 3262-3267.

[21] A. Nielsen, and L. C. Isaac, Quantum Computation and Quantum Information, 1st ed., Cambridge University Press, Cambridge, Apr. 2003, pp. 171-200.

[22] A. Schönhage, "Asymptotically fast algorithms for the numerical multiplication and division of polynomials with complex coefficients," in Computer Algebra EUROCAM'82, Lecture Notes in Computer Science No. 144(J. Calmet, ed.) Springer- Verlag, 1982, pp. 3-15.