IET Information Security

Research Article

Certifying multi-power RSA

ISSN 1751-8709
Received on 11th April 2017
Revised 3rd May 2018
Accepted on 30th August 2018
E-First on 25th February 2019
doi: 10.1049/iet-ifs.2018.5178
www.ietdl.org

Xiaona Zhang[1]*, Li-Ping Wang*[2]*, Jun Xu*[2] ✉

[1]*National Computer Network & Information Security Administrative Centre, No. 7, XibeiWang North Road, Haidian District, Beijing, People's Republic of China*

[2]*State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, Building 4, Minzhuang Road 87C, Haidian District, Beijing, People's Republic of China*

✉ *E-mail: xujun@iie.ac.cn*

Abstract: In this study, the authors present two rigorous algorithms to certify the trapdoor permutation property of the RSA function $\mathrm{RSA}_{N,e}(x) := x^e \bmod N$, where $N = p^r q$ is a multi-power RSA modulus with unknown factorisation and *r* is a known positive integer. Their work gives effective certification for a prime exponent *e* when $e \geq 2N^{((\gcd(r,\,e-1))/(r+1)^2)+\epsilon}$ and for a composite integer $e = e_1^{s_1} e_2^{s_2} \cdots e_u^{s_u}$ when $e_i \geq 2N^{((\gcd(r,\,e_i-1))/(r+1)^2)+\epsilon}$ for $i = 1, \ldots, u$, where e_i is a known prime, s_i is a positive integer, and $\epsilon > 0$ is some small enough constant. The algorithms apply Coppersmith's method for solving univariate modular polynomial equations and run in time $\mathcal{O}(\epsilon^{-7} C \log^2 N)$, where $C \leq ur^2$ is a constant number.

1 Introduction

1.1 Background

Trapdoor permutations, one of the most well known cryptographic primitives have many applications in public-key cryptosystems. Among the various applications of trapdoor functions, certified trapdoor permutations, which were first studied by Bellare and Yung [1], are of particular significance in many cryptographic schemes and protocols. For example, it can be used to build non-interactive zero knowledge protocols for any NP-statement [2–5], to construct ZAPS (a ZAP is a two-round, witness-indistinguishable protocol in which the first round, consisting of a message from the verifier to the prover, can be fixed ‘once and for all’ and applied to any instance [6]) and verifiable pseudo-random functions [6], to build round-optimal blind signatures [7], to build sequential aggregate signatures [8, 9] and so on.

The RSA trapdoor function is the most efficient certified trapdoor permutation to date. Given a composite integer *N* with unknown factorisation and an exponent *e*, the RSA function is defined as $\mathrm{RSA}_{N,e}$: $\mathbb{Z}_N^* \to \mathbb{Z}_N^*$, $x \mapsto x^e \bmod N$, where $\mathbb{Z}_N^* = \{x \in \mathbb{Z}_N : \gcd(x, N) = 1\}$ is the multiplicative group modulo an integer *N*. It is well known that the RSA function defines a permutation over the domain $\mathbb{Z}_N^*$ iff $\gcd(e, \phi(N)) = 1$; here, $\phi(N)$ is the Euler's totient function of *N*. Previous researchers have studied the certifiability of the standard RSA function [10–12]. Cachin *et al.* [10] showed that if $e > N$ and *e* is prime, the RSA function is a certified permutation. (This is, since *e* is a prime, it can never divide $\phi(N) < N$, and hence $\gcd(e, \phi(N)) = 1$.) However, considering the high costs of doing modular exponentiations, we usually not choose $e > N$ in practise. Kiltz *et al.* [12] showed that if $e < N^{1/4}$, the standard RSA function with $N = pq$ is a lossy trapdoor permutation (under the $\Phi-$ hiding assumption [10]) and hence it cannot be certified. In Asiacrypt 2012, Kakvi *et al.* [11] certified the trapdoor permutation property of the RSA function with the multi-power multi-prime RSA $N = \prod_{i=1}^{n} p_i^{k_i}$ for any prime exponent $e \geq N^{(1/4)+\epsilon}$ by using Coppersmith's univariate root-finding method, where $\epsilon > 0$ is some small enough constant. As one can see, for $N = p^r q$, there is no result for $e < N^{1/4}$, our analyses can reach this area for certain *r*, for example, when $r = 2$, the bound is $e \geq 2N^{(2/9)+\epsilon}$.

Multi-power RSA, whose modulus is $N = p^r q$, where *p* and *q* are primes of the same length, and *r* is a positive integer, is a fast variant of the standard RSA [13]. It was first proposed by Takagi in Crypto'98 [14]. Owing to its higher efficiency in both key generation and decryption, it has enjoyed more applications in cryptographic designs such as the Okamoto–Uchiyama cryptosystem [15], the efficient probabilistic public-key encryption and the efficient digital signature algorithms [16], and some lightweight cryptographic devices such as smart cards. These applications as well as its own nice property have made it a hot topic ever since its first introduction. Moreover, there are many analyses and attacks on this scheme, see [17–24].

1.2 Our contributions

In this paper, we study the permutation property of the RSA function with a modulus $N = p^r q$, where the factorisation of *N* is unknown and *r* is a known positive integer. For practical purposes, we assume that $e < N$ during our analyses.

We first study the case when *e* is a prime. In this case, $\gcd(e, \phi(N)) = 1$ is equivalent to $e \nmid \phi(N)$. We put forward an algorithm which outputs $e | \phi(N)$ or $e \nmid \phi(N)$ for a given prime $e \geq 2N^{(\gcd(r,\,e-1)/(r+1)^2)+\epsilon}$. In our analyses, an implicit information [Please see Section 3.1 for concrete elaborations.] is used, and we show in Remark 1 that the lower bound for *e* should only be $e \geq 2N^{(r/(r+1)^2)+\epsilon}$ if not use this implicit information. When $r = 1$, we get Kakvi *et al.*’s result [11], that is, $e \geq N^{(1/4)+\epsilon}$. When $r > 1$, our result usually gives a wider range of *e*, for example, when $r = 2$ the bound is $e \geq 2N^{(2/9)+\epsilon}$ and when $r = 3$ the bound is $e \geq 2N^{(1/16)+\epsilon}$.

Then, we analyse the case when $e = e_1^{s_1} e_2^{s_2} \cdots e_u^{s_u}$ is a composite integer with known factorisation, where e_i is a prime and s_i is a positive integer. In this case, to certify the trapdoor permutation property of the multi-power RSA function, one needs to judge whether or not $\gcd(e, \phi(N)) = 1$. We derive an algorithm which outputs $\gcd(e, \phi(N)) = 1$ or not for any *e* satisfying that $e_i \geq 2N^{(\gcd(r,\,e_i-1)/(r+1)^2)+\epsilon}$ for $i = 1, \ldots, u$.

Besides, our certification algorithm for the prime exponent e also applies to study the validity of the $\Phi-$ hiding assumption. Moreover, our analysis of the composite e can be used to study the security of some cryptographic schemes, for example, Gentry *et al.*'s password authenticated key exchange (PAKE) protocol [25].

1.3 Organszation of the paper

The rest of this paper is organised as follows. In Section 2, we recall some preliminaries. Section 3 first presents our main algorithm to certify the RSA variant with a modulus $N = p^r q$ for e is a prime number, and then studies the case when e is a composite integer. Section 4 is the conclusion.

2 Preliminaries

In this section, we present some basics about lattice, the noted LLL algorithm, Coppersmith's method, Lu *et al.*'s theorem [26], and trapdoor permutations.

2.1 Lattice

Let $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_m$ be linear independent row vectors in $\mathbb{R}^n$, a lattice $\mathscr{L}$ spanned by them is

$$\mathscr{L} = \left\{ \sum_{i=1}^{m} k_i \boldsymbol{b}_i | k_i \in \mathbb{Z} \right\},$$

where $\{\boldsymbol{b}_1, \ldots, \boldsymbol{b}_m\}$ is a basis of $\mathscr{L}$ and $B = [\boldsymbol{b}_1^{\mathrm{T}}, \ldots, \boldsymbol{b}_m^{\mathrm{T}}]^{\mathrm{T}}$ is the corresponding basis matrix. The dimension and determinant of $\mathscr{L}$ are, respectively

$$\dim(\mathscr{L}) = m, \ \det(\mathscr{L}) = \sqrt{\det(BB^{\mathrm{T}})}.$$

Reduced vectors possess many elegant properties, such as the short norm and the orthogonality, thus, calculating the reduced basis of a given lattice is always a hot topic. The reduced basis for a two-rank lattice can be easily obtained by the Gauss algorithm. As for general lattices, the subsequently proposed reduction definitions all have to make a choice between computational efficiency and good reduction performances. The distinguished LLL algorithm [27] takes a good balance, outputting a basis reduced enough for many applications in polynomial time.

Lemma 1: Let $\mathscr{L}$ be a lattice of dimension ω. In polynomial time, the LLL algorithm outputs reduced basis vectors $\boldsymbol{v}_1 \ldots \boldsymbol{v}_w$ that satisfy

$$\| \boldsymbol{v}_1 \| \ \leq \cdots \leq \| \boldsymbol{v}_i \| \ \leq 2^{((\omega(\omega-1))/(4(\omega+1-i)))} \det(\mathscr{L})^{(1/(\omega+1-i))}, \quad 1 \leq i \leq \omega.$$

In practise, it is widely known that the *LLL* algorithm tends to output the vectors whose norms are much smaller than theoretically predicted.

2.2 Finding small roots

In his seminal work [28] in 1996, Coppersmith described a method for finding small roots of univariate modular polynomial equations in polynomial time based on lattice basis reduction. Coppersmith showed that for a monic univariate polynomial $f(x)$ of degree d, one can find any root x_0 of $f(x) \equiv 0(\bmod N)$ in polynomial time if $|x_0| < N^{1/d}$. The essence of Coppersmith's method is to find integral linear combinations of polynomials, which share a common root modulo some integer such that the result has small coefficients. Construct a lattice by defining a lattice basis via these polynomial's coefficient vectors, and use lattice basis reduction algorithms (such as LLL algorithm [27]), one may obtain a polynomial with sufficiently small norm possessing the desired root over the integers and one can then find the desired root using standard root-finding algorithms. Howgrave-Graham [29] showed the condition to quantify the term sufficiently small.

Lemma 2: Let $g(x) \in \mathbb{Z}[x]$ be an integer polynomial that consists of at most ω monomials and N is a composite integer with unknown factorisation. Suppose that

i. $g(y) \equiv 0 \bmod N$ for $|y| \leq X$ and
ii. $\| g(xX) \| < (N/\sqrt{\omega})$ Then $g(y) = 0$ holds over integers.

In Asiacrypt'15, Lu *et al.* [26] used the idea of Coppersmith's method to analyse the type of univariate modular equations with unknown modulus, and by using a novel way to select appropriate polynomials in constructing the desired lattice; they got the following result, which will be used during our analyses. For detailed knowledge about lattice and Coppersmith's method, please see [27–31].

Theorem 1: For every $0 < \epsilon \leq (\beta(2u + v - uv\beta)/7)$, let N be a composite integer (of unknown factorisation) with a divisor q^u ($q \geq N^\beta$, $u \geq 1$). Let $f(x) \in \mathbb{Z}[x]$ be a univariate linear polynomial whose leading coefficient is coprime to N. Then, one can find all the solutions y of the equation $f(x) = 0 \bmod q^v$ with $v \geq 1$, $|y| \leq (1/2)N^\eta$ if $\eta < uv\beta^2 - \epsilon$. The time complexity is $\mathcal{O}(\epsilon^{-7} v^2 \log^2 N)$ [Here we make some modifications to Lu *et al.*'s original theorem [26], and a detailed analysis is presented in the Appendix.].

2.3 Basics of trapdoor permutations

A one-way function is a function that is easy to compute but hard to invert [3]. A one-way permutation is a one-way function that is also a permutation. A trapdoor function is a collection of one-way function $\{f_k: D_k \to R_k\}(k \in K)$, where K, D_k, R_k are subsets of binary strings $\{0, 1\}^*$ satisfying the following conditions:

- There exists a probabilistic polynomial time (PPT) sampling algorithm *Gen* s.t. $\mathrm{Gen}(x^n) = (k, t_k)$ with $k \in K \cap \{0, 1\}^n$ and $t_k \in \{0, 1\}^*$ satisfies $|t_k| < p(n)$, where $p(\cdot)$ is a certain polynomial. Each t_k is called the trapdoor corresponding to k. Each trapdoor can be efficiently sampled.
- Given an input k, there also exists a PPT algorithm that outputs $x \in D_k$. That is, each D_k can be efficiently sampled.
- For any $k \in K$, there exists a PPT algorithm that correctly computes f_k.
- For any $k \in K$, there exists a PPT algorithm A s.t. for any $x \in D_k$, let $y = A(k, f_k(x), t_k)$, and then we have $f_k(y) = f_k(x)$. That is, given trapdoor, it is easy to invert.
- For any $k \in K$, without trapdoor t_k, for any PPT algorithm, the probability to correctly invert f_k (i.e. given $f_k(x)$, find a pre-image x' such that $f_k(x') = f_k(x)$ is negligible.

If each function in the collection above is a one-way permutation, then the collection is also called a trapdoor permutation.

3 Certification algorithms on a multi-power RSA

In this section, we elaborate our algorithms for certifying the trapdoor permutation property of the multi-power RSA function. For a multi-power RSA modulus $N = p^r q$, where p and q are of the same bit length and r is a known positive integer, the essence of the certification procedure is to decide whether $e | \phi(N)$ or not. If $e \nmid \phi(N)$, then the RSA function defines a certified trapdoor permutation over $\mathbb{Z}_N^*$; otherwise, it is not a trapdoor permutation. First, we study the case when e is a prime integer, and then, we analyse the case when e is a composite integer with known factorisation.

```
Require: a RSA modulus $N = p^r q$, and a prime $e \geq 2N^{\frac{gcd(r,e-1)}{(r+1)^2}+\epsilon}$
Ensure: $e|\phi(N)$ or $e \nmid \phi(N)$
Run Theorem 1 to find all the roots of $f(x) = ex + 1 \equiv 0 \bmod q$, where $q^r|N$
if there exists one root of $f(x) \equiv 0 \bmod q$, marked as $x_0$, satisfying that $gcd(ex_0 + 1, N) \neq 1$ then
    return $e|\phi(N)$
else
    Calculate $a \equiv N \bmod e$ and $b = gcd(r, e-1)$
    Run the extended Euclidean algorithm to get the value of $d_1$ which satisfies that $rd_1 + (e-1)d_2 = b$ and calculate $c := a^{d_1} \bmod e$
    Run Theorem 1 to find all the roots of $g(y) = ey + c \equiv 0 \bmod q^b$, where $q^r|N$
    if there exists one root of $g(y) \equiv 0 \bmod q^b$, marked as $y_0$, satisfying that $gcd(ey_0 + c, N) \neq 1$ then
        return $e|\phi(N)$
    else
        return $e \nmid \phi(N)$
    end if
end if
```

Fig. 1 *Algorithm 1: certification algorithm on the multi-power RSA function for a prime exponent e*

3.1 e is a prime integer

In this section, we elaborate our algorithm for certifying the trapdoor permutation property of the multi-power RSA function with a prime integer e. We describe our main result in the following theorem.

Theorem 2: Let $N = p^r q$ be an integer with unknown factorisation and $e < N$ be a prime integer satisfying that $\gamma = \log_N(e/2) \geq \left(\gcd(r, e-1)/(r+1)^2\right) + \epsilon$ and $\gcd(e, N) = 1$. Here, p and q are primes of equal bit length and r is a known positive integer. We can decide whether e divides $\phi(N)$ or not in $\mathcal{O}(\epsilon^{-7} C \log^2 N)$, where $C = (\gcd(r, e-1))^2$.

Proof: From $N = p^r q$, we get that $\phi(N) = p^{r-1}(p-1)(q-1)$. It is obvious that $e \nmid p^{r-1}$ because $\gcd(e, N) = 1$. Then, the problem of testing whether or not $e|\phi(N)$ reduces to decide if e can divide $(p-1)$ or $(q-1)$.

First, we discuss the case when $e|(p-1)$. In this scenario, there exists an integer x satisfying that $ex + 1 = p$, then, we have

$$ex + 1 \equiv 0 \bmod p, \quad \text{where} \quad p^r|N. \tag{1}$$

This polynomial possesses the same structure as the polynomial studied in Theorem 1. By setting $u = r$, $v = 1$, $\beta = (1/(r+1))$, we obtain the solvable range for x, that is, $|x| \leq (1/2)N^{(r/(r+1)^2)-\epsilon}$. Since $e = ((p-1)/x)$, this upper bound on x indicates that we can find the roots of the above equation, and then get the knowledge of p when

$$e \geq 2N^{(1/(r+1)^2)+\epsilon}.$$

Next, we study the case when $e|(q-1)$. Set $a \equiv N \bmod e$, noted that $1 \equiv q \bmod e$, and $N = p^r q$, we get that

$$a \equiv p^r \bmod e, \tag{2}$$

which is because of that $a \equiv p^r q \bmod e \equiv (p^r \bmod e)(q \bmod e) = (p^r \bmod e)1$. Set $b = \gcd(r, e-1)$. According to *the extended Euclidean algorithm*, there exist two integers d_1 and d_2 satisfying that $rd_1 + (e-1)d_2 = b$. Then, we have

$$\begin{aligned} p^b \bmod e &= p^{rd_1+(e-1)d_2} \bmod e \\ &= (p^r)^{d_1}(p^{e-1})^{d_2} \bmod e \\ &= (p^r)^{d_1} \bmod e \\ &= a^{d_1} \bmod e. \end{aligned}$$

Here, we use the *Euler's theorem* that $p^{e-1} \bmod e = 1$. Since a and d_1 can be directly calculated, the value of $a^{d_1} \bmod e$ is immediately computable, and we set $c := a^{d_1} \bmod e$ for convenience. Hence, there exists an integer y satisfying that $ey + c = p^b$, then, we have

$$ey + c \equiv 0 \bmod p^b, \quad \text{where} \quad p^r|N. \tag{3}$$

This polynomial is the same type of the polynomial studied in Theorem 1. By setting $u = r$, $v = b$, $\beta = (1/(r+1))$, we obtain the solvable range for y, that is, $|y| \leq (1/2)N^{(br/(r+1)^2)-\epsilon}$. Since $e = ((p^b - c)/y)$, this upper bound on y indicates that we can find the roots of the above equation, and then get the knowledge of q when

$$e \geq 2N^{(b/(r+1)^2)+\epsilon}.$$

In the above analyses, the most time-consuming step is to find the roots of the derived univariant polynomial equation, which can be accomplished in $\mathcal{O}(\epsilon^{-7} b^2 \log^2 N)$ by applying Theorem 1. □

In practise, we run Theorem 1 to solve (1) and (3), and mark their roots as x_0 and y_0. If either $\gcd(ex_0 + 1, N) \neq 1$ or $\gcd(ey_0 + c, N) \neq 1$ stands true, we get that $e|\phi(N)$, and $\gcd(e, \phi(N)) \neq 1$; hence, the corresponding RSA function is not a certified trapdoor permutation. Otherwise, if for all x_0 and y_0, there is $\gcd(ex_0 + 1, N) = 1$ and $\gcd(ey_0 + c, N) = 1$, which means that $e \nmid \phi(N)$ and $\gcd(e, \phi(N)) = 1$, thus the corresponding RSA function defines a certified trapdoor permutation. We write the certification algorithm in Algorithm 1 (see Fig. 1).

Remark 1: We use an implicit information that $a \equiv N \bmod e$ when checking that whether $e|(q-1)$ or not. If we directly apply Theorem 1 to solve the equation $ex + 1 = q$ without using this information, the result would be $e \geq 2N^{(r/(r+1)^2)+\epsilon}$. We give a brief description to show the deduction process for this bound. From $e|(q-1)$, we get that there exists an integer z satisfying that $ez + 1 = q$. Then there is

$$ez + 1 \equiv 0 \bmod N, \quad \text{where} \quad q|N.$$

Applying Theorem 1 with $u = 1$, $v = 1$, $\beta = (1/(r+1))$, we obtain the root bound for z, that is, $|z| \leq (1/2)N^{(1/(r+1)^2)-\epsilon}$. From $e = ((q-1)/z)$, we derive the constraints on e, that is $e \geq 2N^{(r/(r+1)^2)+\epsilon}$. Compared to our result given in Theorem 2, which is $e \geq 2N^{\left(\gcd(r, e-1)/(r+1)^2\right)+\epsilon}$, this result covers a smaller value range for e. This shows the contribution of the implicit information.

Remark 2: Our Algorithm 1 (Fig. 1) can be directly applied to testify the validity of the $\Phi -$ hiding assumption, based on which many cryptographic schemes are built [10, 25, 32–34]. Basically speaking, this assumption says that, given a composite N with unknown factorisation, it is difficult to tell whether a given prime e can divide $\phi(N)$ or not. According to Algorithm 1 (Fig. 1), the $\Phi -$ hiding assumption fails for a multi-power RSA modulus with $e \geq 2N^{(\gcd(r, e-1)/(r+1)^2)+\epsilon}$.

Remark 3: In Asiacrypt'08, Schridde and Freisleben [35] discussed the validity of the $\Phi -$ hiding assumption on the multi-power RSA modulus $N = p^r q$ by using the Jacobi symbol, where $r > 0$ is an even integer. The essence of their work is to testify whether or not a given prime e can divide $(p-1)$. [The case whether $e|(q-1)$ or not is analysed, and their algorithm cannot factor N when $e|(p-1)$.] Our Algorithm 1 (Fig. 1) works for both odd r and even r, and can directly factor N as long as $e|\phi(N)$.

3.2 e is a composite integer with known factorisation

In this section, we study the permutation property of the RSA function under the condition that e is a composite integer, and its factorisation $e = e_1^{s_1} e_2^{s_2} \cdots e_u^{s_u}$ is public. Here, e_i is a prime and s_i is a positive integer.

The key point of the certification process is to judge whether $\gcd(e, \phi(N)) = 1$ holds true. If it is, then the corresponding RSA function defines a certified trapdoor permutation. Otherwise, the RSA function is not a trapdoor permutation. We put our main result in the following corollary.

Corollary 1: Let $N = p^r q$ be an integer with unknown factorisation and $e = e_1^{s_1} e_2^{s_2} \cdots e_u^{s_u} < N$ be a composite integer satisfying that $\gcd(e, N) = 1$, where e_i is a known prime and s_i is a positive integer. Here, p and q are primes of equal bit length, and $r > 0$ is an integer. We can decide $\gcd(e, \phi(N))$ equals to 1 or not under the condition that $e_i \geq 2N^{\left(\gcd(r,\, e_i-1)/(r+1)^2\right)+\epsilon}$ for $i = 1, \ldots, u$, where ϵ is some small enough positive constant. The time complexity of our algorithm is $\mathcal{O}(4\epsilon^{-7} C \log^2 N)$, where $C = u(\max\{\gcd(r, e_i - 1)\})^2$.

Proof: If $\gcd(e, \phi(N)) \neq 1$, there exists at least one prime factor e_i such that $\gcd(e_i, \phi(N)) = e_i$, which is equivalent to $e_i|\phi(N)$. According to Algorithm 1 (Fig. 1), we can find out $e_i|\phi(N)$ or $e_i \nmid \phi(N)$ as long as $e_i \geq 2N^{\left(\gcd(r,\, e_i-1)/(r+1)^2\right)+\epsilon}$. On the other hand, if none of the e_i can divide $\phi(N)$, then we get that $\gcd(e, \phi(N)) = 1$.

In our analyses, the most time-consuming step is to find the roots of the derived univariant polynomial equation, which can be accomplished in $\mathcal{O}(\epsilon^{-7} b_i^2 \log^2 N)$ by applying Theorem 1, where $b_i = \gcd(r, e_i - 1)$, and in the worst case, we need to run this step for each e_i; thus, the time complexity of our algorithm is $\mathcal{O}(\epsilon^{-7} u b_i^2 \log^2 N)$. □

We describe the certification procedure in Algorithm 2 (see Fig. 2).

```
Require: a RSA modulus $N = p^r q$, and a composite $e = e_1^{s_1} e_2^{s_2} \cdots e_u^{s_u}$ with the condition that $e_i \geq 2N^{\frac{gcd(r, e_i - 1)}{(r+1)^2} + \epsilon}$
    for $i = 1, \cdots, u$.
Ensure: $gcd(e, \phi(N)) = 1$ or $gcd(e, \phi(N)) \neq 1$
for $i$ in $1..u$ do
    run Algorithm 1 on each $e_i$
    if the Algorithm 1 on $e_i$ outputs $e_i | \phi(N)$, which indicates that
        $gcd(e, \phi(N)) \neq 1$ then
        break
        return $gcd(e, \phi(N)) \neq 1$
    end if
end for
return $gcd(e, \phi(N)) = 1$
```

Fig. 2 *Algorithm 2: certification algorithm on the multi-power RSA function for a composite integer e with known factorisation*

Remark 4: In CCS'05, Gentry *et al.* [25] proposed a new technique for designing PAKE protocols, the security of which can be reduced to a simple variant of the $\Phi -$ hiding assumption. For a parameter set given in their work, $u_1 = u_2 = 1$, the security of their scheme is then reduced to the hardness of checking whether a given composite integer e with prime factorisations can divide $(p-1)$ or not, where p is the prime factor of a multi-power RSA modulus $N = p^r q$. In this case, we can directly apply Algorithm 2 (Fig. 2) to break their protocol when $e_i \geq 2N^{(1/4)+\epsilon}$.

4 Conclusion

In this paper, we put forward two algorithms to certify the permutation property of the multi-power RSA function. Our analyses show that, for a prime exponent $e < N$, we can certify this RSA variant as long as $e \geq 2N^{\left(\gcd(r,\, e-1)/(r+1)^2\right)+\epsilon}$. Our work usually covers a wider range of e than Kakvi *et al.*'s result [11] for $r > 1$. Moreover, for a composite integer $e = e_1^{s_1} e_2^{s_2} \cdots e_u^{s_u}$ with known factorisation, our algorithm can tell the trapdoor permutation property of this variant RSA function when $e_i \geq 2N^{\left((\gcd(r,\, e_i-1))/(r+1)^2\right)+\epsilon}$. Besides, our certification algorithm for the prime exponent e can be directly used to study the validity of the $\Phi -$ hiding assumption, and our analysis on the composite e can be used to study the security of some cryptographic schemes, for example, Gentry *et al.'s* PAKE protocol [25]. In addition, we revisit Lu *et al.*'s original theorem and present the corresponding modified proof in the Appendix.

5 Acknowledgments

The authors thank anonymous reviewers for their helpful comments and suggestions. This research was supported by the National Natural Science Foundation of China (Grant nos. 61732012 and 61502488), the National Cryptography Development Fund (MMJJ 20170124), and the Fundamental Theory and Cutting Edge Technology Research Programme of Institute of Information Engineering, CAS (Grant no. Y7Z0341103).

6 References

[1] Bellare, M., Yung, M.: 'Certifying permutations: noninteractive zero-knowledge based on any trapdoor permutation', *J. Cryptol.*, 1996, **9**, (3), pp. 149–166

[2] Feige, U., Lapidot, D., Shamir, A.: 'Multiple non-interactive zero knowledge proofs based on a single random string (extended abstract)'. 31st Annual Symp. Foundations of Computer Science, St. Louis, MO, USA, 22–24 October 1990, vol. **I**, pp. 308–317

[3] Goldreich, O.: '*The foundations of cryptography – volume 1, basic techniques*' (Cambridge University Press, 2001)

[4] Goldreich, O.: '*The foundations of cryptography – volume 2, basic applications*' (Cambridge University Press, 2004)

[5] Goldreich, O.: 'Basing non-interactive zero-knowledge on (enhanced) trapdoor permutations: the state of the art', in Avigad, L., Bellare, M., Brakerski, Z. (Eds.): '*Studies in complexity and cryptography. Miscellanea on the interplay between randomness and computation*' (2011), pp. 406–421

[6] Dwork, C., Naor, M.: 'ZAPS and their applications'. 41st Annual Symp. Foundations of Computer Science, FOCS 2000, Redondo Beach, CA, USA, 12–14 November 2000, pp. 283–293

[7] Garg, S., Rao, V., Sahai, A.*, et al.*: 'Round optimal blind signatures'. 31st Annual Cryptology Conf. Advances in Cryptology – CRYPTO 2011, Santa Barbara, CA, USA, 14–18 August 2011, pp. 630–648

[8] Bellare, M., Namprempre, C., Neven, G.: 'Unrestricted aggregate signatures'. 34th Int. Colloquium Automata, Languages and Programming, ICALP 2007, Wroclaw, Poland, 9–13 July 2007, pp. 411–422

[9] Lysyanskaya, A., Micali, S., Reyzin, L.*, et al.*: 'Sequential aggregate signatures from trapdoor permutations'. Int. Conf. Theory and Applications of Cryptographic Techniques Advances in Cryptology – EUROCRYPT 2004, Interlaken, Switzerland, 2–6 May 2004, pp. 74–90

[10] Cachin, C., Micali, S., Stadler, M.: 'Computationally private information retrieval with polylogarithmic communication'. Advances in Cryptology EUROCRYPT 99, Prague, Czech Republic, 1999, (LNCS, **1592**), pp. 402–414

[11] Kakvi, S., Kiltz, E., May, A.: 'Certifying RSA'. Advances in Cryptology – ASIACRYPT 2012, Berlin, Heidelberg, 2012, (LNCS, **7658**), pp. 404–414

[12] Kiltz, E., ONeill, A., Smith, A.: 'Instantiability of RSA-OAEP under chosen-plaintext attack'. Advances in Cryptology – CRYPTO 2010, Santa Barbara, CA, USA, 2010, (LNCS, **6223**), pp. 295–313

[13] Rivest, R.L., Shamir, A., Adleman, L.M.: 'A method for obtaining digital signatures and public-key cryptosystems', *Commun. ACM*, 1978, **21**, (2), pp. 120–126

[14] Takagi, T.: 'Fast RSA-type cryptosystem modulo $p^k q$'. 18th Annual Int. Cryptology Conf. Advances in Cryptology – CRYPTO '98, Santa Barbara, CA, USA, 23–27 August 1998, pp. 318–326

[15] Okamoto, T., Uchiyama, S.: 'A new public-key cryptosystem as secure as factoring'. Int. Conf. Theory and Application of Cryptographic Techniques Advances in Cryptology – EUROCRYPT '98, Espoo, Finland, 31 May–4 June 1998, pp. 308–318

[16] The EPOC and the ESIGN Algorithms. IEEE P1363: Protocols from Other Families of Public-Key Algorithms, 1998

[17] Boneh, D., Durfee, G., Howgrave-Graham, N.: 'Factoring $N=p^r q$ for large r'. 19th Annual Int. Cryptology Conf. Advances in Cryptology – CRYPTO '99, Santa Barbara, CA, USA, 15–19 August 1999, pp. 326–337

[18] May, A.: 'Secret exponent attacks on RSA-type schemes with moduli $N = p^r q$'. Seventh Int. Workshop on Theory and Practice in Public Key

Cryptography Public Key Cryptography – PKC 2004, Singapore, 1–4 March 2004, pp. 218–230

[19] Nitaj, A., Rachidi, T.: 'New attacks on RSA with moduli $N = p^rq$'. Proc. – In Honor of Thierry Berger First Int. Conf. Codes, Cryptology, and Information Security, C2SI 2015, Rabat, Morocco, 26–28 May 2015, pp. 352–360

[20] Santosh, K.R., Narasimham, C., Shettys, P.: 'Cryptanalysis of multi-prime RSA with two decryption exponents', *Int. J. Electron. Inf. Eng.*, 2016, **4**, pp. 40–44.

[21] Sarkar, S.: 'Small secret exponent attack on RSA variant with modulus $N = p^rq$', *Des. Codes Cryptogr.*, 2014, **73**, (2), pp. 383–392

[22] Sarkar, S.: 'Revisiting prime power RSA', *Discrete Appl. Math.*, 2016, **203**, pp. 127–133

[23] Takayasu, A., Kunihiro, N.: 'How to generalize RSA cryptanalyses'. 19th IACR Int. Conf. Practice and Theory in Public-Key Cryptography Public-Key Cryptography – PKC 2016, Taipei, Taiwan, 6–9 March 2016, pp. 67–97, Part II

[24] Zheng, M.C., Hu, H.G.: 'Cryptanalysis of prime power RSA with two private exponents', *Sci. China Inf. Sci.*, 2015, **58**, (11), pp. 1–8

[25] Gentry, C., Mackenzie, P., Ramzan, Z.: 'Password authenticated key exchange using hidden smooth subgroups'. Proc. 12th ACM Conf. Computer and Communications Security CCS 2005, Alexandria, VA, USA, 7–11 November 2005, pp. 299–309

[26] Lu, Y., Zhang, R., Peng, L., *et al.*: 'Solving linear equations modulo unknown divisors: revisited'. 21st Int. Conf. Theory and Application of Cryptology and Information Security Advances in Cryptology – ASIACRYPT 2015, Auckland, New Zealand, 29 November–3 December 2015, pp. 189–213, Part I

[27] Lenstra, A.K., Lenstra, H.W., Lovász, L.: 'Factoring polynomials with rational coefficients', *Math. Ann.*, 1982, **261**, (4), pp. 515–534

[28] Coppersmith, D.: 'Finding a small root of a bivariate integer equation; factoring with high bits known'. Int. Conf. Theory and Application of Cryptographic Techniques, Advances in Cryptology – EUROCRYPT '96, Saragossa, Spain, 12–16 May 1996, pp. 178–189

[29] Howgrave-Graham, N.: 'Finding small roots of univariate modular equations revisited', in Darnell, M. (Ed.): '*Crytpography and coding*', Lecture Notes in Computer Science, vol. **1355**, (Springer, Berlin, Heidelberg, 1997), pp. 131–142

[30] Coppersmith, D.: 'Finding a small root of a univariate modular equation'. Int. Conf. Theory and Application of Cryptographic Techniques, Advances in Cryptology – EUROCRYPT '96, Saragossa, Spain, 12–16 May 1996, pp. 155–165

[31] Coppersmith, D.: 'Small solutions to polynomial equations, and low exponent RSA vulnerabilities', *J. Cryptol.*, 1997, **10**, (4), pp. 233–260

[32] Cachin, C.: 'Efficient private bidding and auctions with an oblivious third party'. Proc. Sixth ACM Conf. Computer and Communications Security, CCS '99, Singapore, 1–4 November 1999, pp. 120–127

[33] Gentry, C., Ramzan, Z.: 'Single-database private information retrieval with constant communication rate'. 32nd Int. Colloquium Automata, Languages and Programming ICALP 2005, Lisbon, Portugal, 11–15 July 2005, pp. 803–815

[34] Hemenway, B., Ostrovsky, R., Strauss, M.J., *et al.*: 'Public key locally decodable codes with short keys'. Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques – 14th Int. Workshop, APPROX 2011, and 15th Int. Workshop, RANDOM 2011, Princeton, NJ, USA, 17–19 August 2011, pp. 605–615

[35] Schridde, C., Freisleben, B.: 'On the validity of the ϕ-hiding assumption in cryptographic protocols'. Advances in Cryptology – ASIACRYPT 2008, Melbourne, Australia, 2008 (LNCS, **5350**), pp. 344–354

[36] Nguyen, P.Q., Stehlé, D.: 'Floating-point LLL revisited'. 24th Annual Int. Conf. Theory and Applications of Cryptographic Techniques Advances in Cryptology – EUROCRYPT 2005, Aarhus, Denmark, 22–26 May 2005, pp. 215–233

[37] May, A.: 'New RSA vulnerabilities using lattice reduction methods'. PhD thesis, University of Paderborn, 2003

7 Appendix

Here, we give a modified proof for Lu *et al.*'s theorem [26], where we show that the constraint 'N be a sufficiently large composite integer' is not necessary, and a condition that $\epsilon \le (\beta(2u + v - uv\beta)/7)$ should be added in Lu *et al.*'s theorem [26]. We present the modified theorem as follows.

Theorem: (Lu et al.'s Theorem [26]): For every $0 < \epsilon \le (\beta(2u + v - uv\beta)/7)$, let N be a composite integer (of unknown factorisation) with a divisor $q^u (q \ge N^\beta, u \ge 1)$. Let $f(x) \in \mathbb{Z}[x]$ be a univariate linear polynomial whose leading coefficient is coprime to N. Then, one can find all the solutions y of the equation $f(x) = 0 \bmod q^v$ with $v \ge 1$, $|y| \le (1/2)N^\eta$ if $\eta < uv\beta^2 - \epsilon$. The time complexity is $\mathcal{O}(\epsilon^{-7}v^2\log^2 N)$.

Proof: Set $f(x) = x + a \bmod q^v$, [Here, we assume that $f(x)$ is a monic univariate linear polynomial. Since it is easy to make $f(x)$ to be monic by multiplying $a_0^{-1} \bmod N$ to $f(x)$, here $a_0 \ne 1$ is the leading coefficient of $f(x)$. Once the inverse of a_0^{-1} does not exist, one can find a non-trivial factor of N.] and $X = (1/2)N^{uv\beta^2 - \epsilon}$ is the upper bound of the roots for $f(x) = 0 \bmod q^v$. Build a cluster of polynomials $g_k(x) = f^k(x)N^{\{\max\{\lceil v(t-k)/u \rceil, 0\}\}}$, where $k = 0, \ldots, m$, and $t = \tau m$ for $\tau \in [0, 1)$, which will be optimised later. From this construction, we get that $g_k(x) = 0 \bmod q^{vt}$ share the same roots as $f(x) = 0 \bmod q^v$. Use coefficients of $g_k(Xx)$ as basis row vectors to build a lattice $\mathscr{L}$ and arrange the polynomials in an increasing order, that is, $g_{k_1} > g_{k_2}$ if $k_1 > k_2$. Built this way, each polynomial introduces one and only one new monomial, that is, $X^kx^kN^{\{\max\{\lceil v(t-k)/u \rceil, 0\}\}}$. Thus, we obtain a lower triangular lattice, whose determinant can be easily calculated as the product of the diagonal items. Run the LLL algorithm on $\mathscr{L}$, and use $\boldsymbol{v}$ to represent the first output vector by the *LLL* algorithm. Then, the polynomial corresponding to $\boldsymbol{v}$ possess the same roots as $g_k(x) = 0 \bmod q^{vt}$ and have small enough norms which will make it hold over integers. Thus, one can use the root-finding algorithms to get the desired roots.

We can calculate the dimension of $\mathscr{L}$ as $d = m + 1$, and the determinant of $\mathscr{L}$ as

$$\det(\mathscr{L}) = X^{S_X}N^{S_N},$$

where $S_X = (m(m + 1)/2)$

$$S_N = \sum_{k=0}^{t-1}\left(\frac{v(t-k)}{u} + c_k\right) = \frac{v\tau m(\tau m + 1)}{2u} + \sum_{k=0}^{t-1} c_k, \quad \text{here} \quad c_k \in [0, 1).$$

According to Howgrave-Graham's lemma, the constraint that $\|\boldsymbol{v}\| \le (q^{v\tau m}/\sqrt{d})$ shall be satisfied to ensure that the equation corresponding to $\boldsymbol{v}$ holds over integers. Since the LLL algorithm outputs the desired $\boldsymbol{v}$ satisfying $\|\boldsymbol{v}\| \le 2^{((\omega - 1)/4)} \det(\mathscr{L})^{1/d}$ in polynomial time, we require the following condition:

$$2^{((\omega-1)/4)} \det(\mathscr{L})^{1/d} \le \frac{q^{v\tau m}}{\sqrt{d}}. \tag{4}$$

Put the values of $\det(\mathscr{L})$ and d in (4) and considering that $q \ge N^\beta$, we can obtain

$$2^{(m(m+1)/4)}(m+1)^{((m+1)/2)}X^{(m(m+1)/2)} < N^{v\beta\tau m(m+1) - (v\tau m(\tau m+1)/2u) - \sum_{k=0}^{l-1} c_k} \tag{5}$$

which can be simplified as

$$X \le 2^{-(1/2)}(m+1)^{-(1/m)}N^{2v\beta\tau - (v\tau(\tau m+1)/u(m+1)) - \left(2\sum_{k=0}^{t-1} c_k/(m(m+1))\right)}$$

.

Here, $(m+1)^{-(1/m)} = d^{-(1/(d-1))} = 2^{-(\log_2 d/(d-1))} \ge 2^{-(1/2)}$, for $d \ge 7$. Hence, the above inequality can be simplified as

$$X \le \frac{1}{2}N^{2v\beta\tau - ((v\tau(\tau m+1))/(u(m+1))) - \left(2\sum_{k=0}^{t-1} c_k/(m(m+1))\right)}.$$

Set $\tau = u\beta$, and noting that $\sum_{k=0}^{t-1} c_k < \tau m$, the exponent of N can be lower bounded by

$$uv\beta^2 - \frac{\beta(2u + v - uv\beta)}{m+1}.$$

Thus, by choosing $m \ge m^* = \lceil \beta(2u + v - uv\beta)/\epsilon \rceil - 1$, one can get the upper bound for X, that is

$$X \leq \frac{1}{2}N^{uv\beta^2 - \epsilon},$$

Here, ϵ is any chosen value that satisfying $0 \leq \epsilon \leq (\beta(2u + v - uv\beta)/7)$. Hence, we can ensure that $d = m + 1 \geq 7$.

The running time of this algorithm is dominated by the LLL algorithm, which is polynomial in the dimension of the lattice and in the maximal bit size of the entries. Here is a lower bound for the dimension of $\mathscr{L}$, that is

$$d = m + 1 \geq \frac{\beta(2u + v - uv\beta)}{\epsilon}.$$

Since $u\beta < 1$, we obtain $d = \mathcal{O}(\epsilon^{-1})$. The maximal bit size of the entries in $\mathscr{L}$ is bounded by

$$\max\left\{\frac{vt}{u}\log(N),\ muv\beta^2\log(N)\right\}$$
$$< \max\{v\beta d\log(N),\ duv\beta^2\log(N)\}.$$

Since $u\beta < 1$ and $d = \mathcal{O}(\epsilon^{-1})$, the bit size of the entries can be upper bounded by

$$\max\{\mathcal{O}(v\beta\epsilon^{-1})\log(N),\ \mathcal{O}(v\beta\epsilon^{-1})\log(N)\} = \mathcal{O}(v\epsilon^{-1})\log(N).$$

Nguyen and Stehlé proposed a modified version of the LLL algorithm called L^2 algorithm in [36]. The L^2 algorithm achieves the same approximation quality for a shortest vector as the LLL algorithm and has an improved worst-case running time complexity. The running time is $\mathcal{O}(d^5(d + \log b_d \log b_d))$, where $\log b_d$ is the maximal bit size of entries in the lattice. Thus, we can obtain the running time of this algorithm

$$\mathcal{O}(\epsilon^{-5}(\epsilon^{-1} + v\epsilon^{-1}\log(N))v\epsilon^{-1}\log(N)) = \mathcal{O}(\epsilon^{-7}v^2\log^2 N).$$

Thus, the shortest vector output by the LLL algorithm gives a univariate polynomial $g(x)$ such that $g(y) = 0$ shares the same roots as $f(x) = 0 \bmod p^v$, and one can get the desired roots over the integers. □

Remark 5: In Lu *et al.*'s original proof for this theorem in [26], the authors let m grow to infinity to obtain the asymptotic bound, and they constrain N to be sufficiently large to make the powers of 2 and $m + 1$ to be negligible. We say that these two settings are actually not required. First, we will show that the powers of 2 and $m + 1$ are negligible when m goes to infinity, and there is no need N to be sufficiently large.

Here, we compute the value of $2^{-(1/2)}(m + 1)^{-(1/m)}$ for m goes to infinity

$$\lim_{m \to \infty}(m + 1)^{1/m} = \lim_{m \to \infty} \mathrm{e}^{(\ln(m+1)/m)} = \mathrm{e}^{\lim_{m \to \infty}(\ln(m+1)/m)}$$
$$= e^0 = 1.$$

Hence, $\lim_{m \to \infty} 2^{-(1/2)}(m + 1)^{-(1/m)} = 2^{-(1/2)}$, which is a constant independent of N.

Second, when m grows to infinity, the dimension of $\mathscr{L}$, $d = m + 1$, goes to infinity, then the LLL algorithm cannot work in polynomial time. From our modified proof, one can find that the powers of 2 and $m + 1$ are bounded by 1/2 for $d \geq 7$. Both the constraints N to be sufficiently large and m grow to infinity are not required. Actually, the proof for Coppersmith's original theorem for finding solutions of univariate polynomial equations is similar to the proof of Lu *et al.*'s theorem, and may give a detailed analysis in his Ph.D. Thesis [37].