DE GRUYTER OPEN Acta Univ. Sapientiae, Informatica **7,** 2 (2015) 143–150

DOI: 10.1515/ausi-2015-0016

Factoring multi power RSA moduli with a class of secret exponents

Omar AKCHICHE
Laboratory of Mathematics, Cryptography and Mechanics, Fstm
University Hassan II of Casablanca, Morocco
email: omar.akchiche@hotmail.com

Omar KHADIR
Laboratory of Mathematics, Cryptography and Mechanics, Fstm
University Hassan II of Casablanca, Morocco
email: khadir@hotmail.com

Abstract. In this paper, we consider the RSA variant based on the key equation $ed \equiv 1 \pmod{\phi(N)}$ where $N = p^r q$, $r \geq 2$. We show that if the secret exponent d is close to any multiple of the prime factor p or its powers, then it is possible to factor N in polynomial time in $\log N$.

1 Introduction

Factoring large integers is a well established problem in number theory and cryptography. The security of many cryptosystems such as RSA [15] is based on its presumed difficulty. Fermat method is efficient to factor a product of two numbers that are close one to another (see e.g. [4]). In 1931, the continued fraction method was invented [5]. Pollard [11] described the $p-1$ method in 1974. Some years later, he discovered the ρ algorithm [12]. The first is known to be practical when the prime factors of $p-1$ are small for some prime divisor of N. The second applies a cycle detection technique. In [7], Lenstra employed the elliptic curves properties to get prime factors of large numbers.

Computing Classification System 1998: F.2.1 E.3
Mathematics Subject Classification 2010: 11Y05 94A60
Key words and phrases: integer factorization problem, multipower RSA, public key cryptography

The Quadratic Sieve was published in 1980s by Pomerance [13]. Nowadays, the most efficient factoring algorithm is the Number Field Sieve (NFS) [6, p. 103] that was elaborated by Pollard in 1988. Since then, the NFS has been further ameliorated.

In order to speed up the decryption/encryption time, it was suggested to use RSA with moduli $N = p^r q$ (see e.g. [2]). Numerous previous papers studied the security of such protocols. Boneh, Durfee, and Howgrave-Graham [1] established that only $\frac{1}{1+r}$ fraction of the bits of p suffice to recover the entire p. May [9] generalizes many cryptanalysis methods to schemes with $N = p^r q$. Some of the work in [9] was improved in [16] for $r \leq 5$. It was proved in [8] that leaking the bits of some blocks of the prime factors of a modulus $N = p^r q$ enables its factorization under certain circumstances. All the previous researches pointed out that integers $N = p^r q$ are more vulnerable than a standard RSA modulus, in particular, when r becomes large.

In this paper, the RSA variant based on the key equation $ed \equiv 1 \pmod{\phi(N)}$ where $N = p^r q$ is considered. We show that using a secret exponent d which is close to any multiple of p or its powers can lead to the factorization of the public modulus.

The article is organized as follows. In section 2, we state our main result after recalling the Coppersmith's theorem for finding small roots of univariate modular polynomials. In section 3, we generalize the method to other RSA-type systems. Finally, we conclude in section 4.

Throughout the sequel, for integers a, b and c, we write $a \equiv b \pmod{c}$ if c divides the difference $a - b$, and $a = b \mod c$ if a is the remainder in the division of b by c. We denote by $\gcd(a, b)$ the greatest common divisor of a and b. All the logarithms should be interpreted as logarithms to the base 2.

2 Our contribution

In this section, we describe our main result. However, we start by presenting the Coppersmith [3] result for computing small roots of modular polynomials. In particular, we use the slight generalized version as was depicted by May in [9, 10]. One can find in [10] a thorough treatment about the method and its implementation. This technique will be needed in establishing our Theorem 2.

Theorem 1 ([10]) *Let N be an integer of unknown factorization, which has a divisor $b \geq N^\beta$, $0 < \beta \leq 1$. Furthermore, let $f(x)$ be a univariate monic polynomial of degree δ and let $c \geq 1$. Then, we can find all solutions x_0 for*

the equation:

$$f(x_0) \equiv 0 \pmod{b} \text{ with } |x_0| \leq cN^{\frac{\beta^2}{\delta}}$$

in time $T = O(c\delta^5 \log^9 N)$.

Now, we state the main contribution:

Theorem 2 *Let* $N = p^r q$ *where* $r \geq 2$ *is a given integer constant and* p, q *are primes of the same bit-size. We denote by* (e, d) *the public-key/secret-key pair satisfying* $ed \equiv 1 \pmod{\phi(N)}$. *Assume that there exist two integers* $i \geq 1$ *and* j *that verify* $|d - jp^i| \leq N^{\left(\frac{\min(i, r-1)}{r+1}\right)^2}$ *with* $(d - jp^i) \not\equiv e^{-1} \pmod{q}$. *Parameters* i *and* j *are not necessary known. Then, we can factor* N *in polynomial time in* $\log N$.

Proof. The RSA equation is $ed \equiv 1 \pmod{\phi(N)}$. Hence, there exists an integer k such that $ed = 1 + kp^{r-1}(p-1)(q-1)$. Let $d = jp^i + \Delta$ where $\Delta \in \mathbb{Z}$ and put $l = \min(i, r-1)$. Working modulo p^l, it follows that $e\Delta - 1 \equiv 0 \pmod{p^l}$. Setting the polynomial $f(x) = x - (e^{-1} \mod N)$, it is clear that Δ is a root of $f(x)$ modulo p^l. We assume that the inverse of e modulo N is well defined. Otherwise, we have already a non trivial divisor of N. When $q < p$, $N^{\frac{l}{r+1}} < p^l$. By hypothesis, $|\Delta| \leq N^{\left(\frac{l}{r+1}\right)^2}$. So, we can determine Δ by using Theorem 1 with $b = p^l$, $\beta = \dfrac{l}{r+1}$, $\delta = 1$ and $c = 1$. Since $\Delta \not\equiv e^{-1} \pmod{q}$, it is readily seen that $\gcd(|e\Delta - 1|, N)$ splits N as $e\Delta - 1$ and N are both multiples of p^l. If $p < q$, then $q < 2p$ as p and q are of the same bit-size. Thus, $\dfrac{N^{\frac{l}{r+1}}}{2} < p^l$. Let $\beta = \dfrac{l}{r+1} - \dfrac{1}{\log N}$. We have $|\Delta| \leq N^{\frac{\beta^2}{\delta}}$ whenever $|\Delta| \leq N^{\left(\frac{l}{r+1}\right)^2}$. This comes from observing that $N^{\left(\frac{l}{r+1}\right)^2} = 4N^{\left(\frac{l}{r+1}\right)^2 - \frac{2}{\log N}} < 4N^{\left(\frac{l}{r+1}\right)^2 - \frac{2l}{(r+1)\log N} + \frac{1}{\log^2 N}} = 4N^{\frac{\beta^2}{\delta}}$. Hence, we obtain Δ by applying Theorem 1 with $b = p^l$, $\beta = \dfrac{l}{r+1} - \dfrac{1}{\log N}$, $\delta = 1$ and $c = 4$. We then get the factorization of N by computing $\gcd(|e\Delta - 1|, N)$.

The running time of our method is dominated by that of Theorem 1 which is polynomial in $\log N$. So, the result is proved.

□

Theorem 2 leads to the following algorithm:

Input: A public multi power RSA key (N, e) where $N = p^r q$ for a given constant $r \geq 2$.

Output: The prime decomposition of N or "Failure".

1. Set the modular polynomial $f(x) = x - (e^{-1} \mod N)$.
2. Apply Coppersmith method as presented in Theorem 1 to compute all the integer roots of $f(x)$ modulo p^l where $l \leq r-1$ is a fixed integer. It is not required to know the value of p.
3. Denote by $x_i, i = 1, 2, \ldots, \mathtt{length}$, the solutions founded in step 2.
4. flag $\leftarrow 0$, $i \leftarrow 0$.
5. While flag $= 0$ and $i \leq \mathtt{length}$ do:
  5.1. $i \leftarrow i + 1$.
  5.2. $\Delta \leftarrow x_i$.
  5.3. $f \leftarrow \gcd(|e\Delta - 1|, N)$.
  5.4. If $1 < f < N$ then:
    5.4.1. flag $\leftarrow 1$.
    5.4.2. If f is not a prime power, then $q \leftarrow f$, $p \leftarrow \left(\frac{N}{q}\right)^{\frac{1}{r}}$.
    5.4.3. Else, determine the prime p that divides f, $q \leftarrow \frac{N}{p^r}$.
    5.4.4. Output (p, q).

6. If $i > \mathtt{length}$, then output "Failure".

For a multi power RSA modulus $N = p^r q$, it is generally recommended to choose a small value of r. Indeed, the more r is large, the less the cryptosystem is secure, see e.g. [1, 9, 8, 16]. Setting $r = 2$, we obtain the next corollary:

Corollary 3 *Let $N = p^2 q$ where p and q are primes of the same bit-size. We denote by (e, d) the public-key/secret-key pair satisfying $ed \equiv 1 \pmod{\phi(N)}$. Assume that there are two integers $i \geq 1$ and j such that $|d - jp^i| \leq N^{\frac{1}{9}}$ with $(d - jp^i) \not\equiv e^{-1} \pmod{q}$. Then, we can factor N in polynomial time in $\log N$.*

The bound $N^{\left(\frac{\min(i, r-1)}{r+1}\right)^2}$ in Theorem 2 is optimal for $i \geq r-1$. Under this situation, it is roughly equal to N when r becomes larger.

In the next section, we investigate the threat of our method to other RSA variants.

3 Extension of our result

The straightforward multi power RSA is obtained by taking $N = p^r q$ in the standard RSA key equation $ed \equiv 1 \pmod{\phi(N)}$. The Takagi crypotsystem [17] is based on $ed_p \equiv 1 \pmod{p-1}$ and $ed_q \equiv 1 \pmod{q-1}$. For this protocol, we have:

Proposition 4 *Let $N = p^r q$ where $r \geq 2$ is a given integer constant and p, q are primes of the same bit-size. We denote by (e, d_p) the public-key/secret-key pair satisfying $ed_p = 1 + k_p(p-1)$, i.e. $ed_p \equiv 1 \pmod{p-1}$. Assume that $|d_p - p| \leq N^{\frac{1}{(r+1)^2}}$ with $(d_p - p) \not\equiv (1-k_p)e^{-1} \pmod{q}$. Then, we can factor N in time $eO(\log^9 N)$.*

Proof. By definition, $ed_p = 1 + k_p(p-1)$ for some integer k_p. Put $d_p = p + \Delta$. Hence $e\Delta + k_p - 1 \equiv 0 \pmod{p}$. The parameter k_p lands in the set $\{1, 2, \ldots, e-1\}$. Indeed, $k_p = \dfrac{ed_p - 1}{p-1} < e$. Put $f(x) = x + (k_p - 1)(e^{-1} \mod N)$. For the true guess for k_p, Δ is a root of the polynomial $f(x)$ modulo p.

If $q < p$, then $N^{\frac{1}{1+r}} < p$. By hypothesis, $|\Delta| \leq N^{\frac{1}{(1+r)^2}}$. So, it is possible to find Δ by applying Theorem 1 to $f(x)$ with $\beta = \dfrac{1}{1+r}$, $\delta = 1$ and $c = 1$. Moreover, $\Delta \not\equiv (1-k_p)e^{-1} \pmod{q}$. Thus, $\gcd(|e\Delta + k_p - 1|, N)$ is a prime power that divides N, since both $e\Delta + k_p - 1$ and N are multiples of p. The most time consuming part is Coppersmith method which has a running time $O(\log^9 N)$. All the steps must be repeated for each trial k_p. Therefore, the whole complexity is $eO(\log^9 N)$.

Now, suppose that $p < q$. The primes p and q are of the same bit-size, so $q < 2p$. It follows that $\dfrac{N^{\frac{1}{1+r}}}{2} < p$. We have $N^{\frac{1}{(r+1)^2}} = 4N^{\left(\frac{1}{r+1}\right)^2 - \frac{2}{\log N}} < 4N^{\left(\frac{1}{r+1}\right)^2 - \frac{2}{(r+1)\log N} + \frac{1}{\log^2 N}} = 4N^{\frac{\beta^2}{\delta}}$. Using Theorem 1 with $f(x) = x + (k_p - 1)(e^{-1} \mod N)$, $\beta = \dfrac{1}{1+r} - \dfrac{1}{\log N}$, $\delta = 1$ and $c = 4$ leads to recovering Δ. Like in the previous case, $\gcd(|e\Delta + k_p - 1|, N)$ is a prime power divisor of N which achieves the proof.

□

For an RSA-type modulus $N = p^r q$, the primes p and q are not symmetric. We can establish:

Proposition 5 *Let $N = p^r q$ where $r \geq 2$ is a given integer constant and p, q are primes of the same bit-size. We denote by (e, d_q) the public-key/secret-key*

pair satisfying $ed_q = 1 + k_q(q-1)$, *i.e.* $ed_q \equiv 1 \pmod{q-1}$. *Assume that* $|d_q - q| \leq N^{\frac{1}{(r+1)^2}}$ *with* $(d_q - q) \not\equiv (1-k_q)e^{-1} \pmod{p^u}$ *for all* $u \leq r$. *Then, we can factor* N *in time* $eO(\log^9 N)$.

Proof. The RSA key equation satisfies $ed_q = 1 + k_q(q-1)$ for some integer k_q. The following process will be repeated for each candidate for k_q. If $d_q = q + \Delta$ where $\Delta \in \mathbb{Z}$, then $e\Delta + k_q - 1 \equiv 0 \pmod{q}$. We define the function $f(x) = x + (k_q - 1)(e^{-1} \mod N)$. Let $p < q$. Thus, $N^{\frac{1}{1+r}} < q$. By hypothesis, $|\Delta| \leq N^{\frac{1}{(r+1)^2}}$. Setting $\beta = \dfrac{1}{1+r}$, $\delta = 1$ and $c = 1$, we efficiently determine Δ by Theorem 1. If $q < p$, $\dfrac{N^{\frac{1}{1+r}}}{2} < q$ since p and q are of the same bit-size. One shows that it suffices that $|\Delta| \leq N^{\frac{1}{(r+1)^2}}$ in order to use Coppermsith's result with $\beta = \dfrac{1}{1+r} - \dfrac{1}{\log N}$, $\delta = 1$ and $c = 4$.

As both $e\Delta + k_q - 1$ and N are divided by q, the condition $(d_q - q) \not\equiv (1-k_q)e^{-1} \pmod{p^u}$ for all $u \leq r$ guarantees that $\gcd(|e\Delta + k_q - 1|, N) = q$. All the steps are executed at most e times given that $k_q < e$. Hence, the running time of the method is $eO(\log^9 N)$ which demonstrates the result.

□

Suppose that a private exponent d_p or d_q satisfies the hypothesis of Proposition 4 or 5 respectively. It is clear that if there exists an oracle that outputs the values of k_p or k_q such that $ed_p = 1 + k_p(p-1)$ or $ed_q = 1 + k_q(q-1)$, then N can be factored in polynomial time in $\log N$.

In the following proposition, we apply the technique for RSA systems that use Chinese remainder theorem in decrypting, CRT-RSA (see e.g. [14] for an explicit description). We obtain:

Proposition 6 *Let* $N = pq$ *an RSA modulus where* p *and* q *are primes of the same bit-size. We denote by* (e, d_p) *the public-key/secret-key pair satisfying* $ed_p = 1 + k_p(p-1)$, *i.e.* $ed_p \equiv 1 \pmod{p-1}$. *Assume that* $|d_p - p| \leq N^{\frac{1}{4}}$ *with* $(d_p - p) \not\equiv (1-k_p)e^{-1} \pmod{q}$. *Then, we can factor* N *in time* $eO(\log^9 N)$.

Proof. By the RSA key equation, $ed_p = 1 + k_p(p-1)$ where $k_p \in \mathbb{N}$. It follows that $ed_p + k_p - 1 \equiv 0 \pmod{p}$. We know that $d_p = p + \Delta$, so $e\Delta + k_p - 1 \equiv 0 \pmod{p}$. The modulus $N = pq$ is balanced. Consider the polynomial $f(x) = x + (k_p - 1)(e^{-1} \mod N)$ whose degree is $\delta = 1$. The value of $|\Delta|$ is upper bounded by $N^{\frac{1}{4}}$. It is possible to compute efficiently Δ by Theorem 1 with

$\beta = \frac{1}{2}$, $c = 1$ if $q < p$, and $\beta = \frac{1}{2} - \frac{1}{\log N}$, $c = 2$ if not. By hypothesis, $\Delta \not\equiv (1 - k_p)e^{-1} \pmod{q}$, so $\gcd(|e\Delta + k_p - 1|, N) = p$. We must execute the method for each candidate for k_p. As $k_p = \frac{ed_p - 1}{p - 1}$ and $d_p < p - 1$, $k_p < e$. So, the running time is $eO(c \log^9 N)$ where $c = 1$ if $q < p$ and $c = 2$ otherwise. □

Let d_p a private exponent that fulfil the hypothesis of Proposition 6. If the value of k_p such that $ed_p = 1 + k_p(p - 1)$ is leaked, then we can efficiently compute the prime decomposition of N.

4 Conclusion

In this paper, we proposed an attack against the RSA variant based on the key equation $ed \equiv 1 \pmod{\phi(N)}$ where $N = p^r q$, $r \geq 2$. We showed that if d is close to any multiple of the prime factor p or its powers, then N can be factored in polynomial time in $\log N$, and thus the cryptosystem is completely broken.

Acknowledgements

This work is supported by the MMSyOrientation project.

References

[1] D. Boneh, G. Durfee, and N. Howgrave-Graham, Factoring $N = p^r q$ for large r, *Advances in Cryptology (CRYPTO'99)*, *Lecture Notes in Computer Science* **1666** (1999) 326–337. ⇒144, 146

[2] D. Boneh and H. Shacham, Fast variants of RSA, *CryptoBytes* **5,** 1 (2002) 1–9. ⇒144

[3] D. Coppersmith, Small solutions to polynomial equations, and low exponent RSA vulnerabilities, *J. Cryptology* **10,** 4 (1997) 233–260. ⇒144

[4] B. De Weger, Cryptanalysis of RSA with small prime difference, *Appl. Algebra Engrg. Comm. Comput.* **13,** 1 (2002) 17–28. ⇒143

[5] Derrick H. Lehmer and Richard E. Powers, On factoring large numbers, *Bull. Amer. Math. Soc.* **37,** 10 (1931) 770–776. ⇒143

[6] Arjen K. Lenstra, Hendrik W. Lenstra Jr., *The development of the number field sieve*, Lecture Notes in Mathematics **1554,** Springer-Verlag, 1993. ⇒144

[7] Hendrik W. Lenstra Jr., Factoring integers with elliptic curves, *Ann. of Math.* **126,** 3 (1987) 649–673. ⇒143

[8] Y. Lu, R. Zhang, and D. Lin, Factoring multi-power RSA modulus $N = p^r q$ with partial known bits, *Information Security and Privacy (ACISP 2013)*, *Lecture Notes in Computer Science* **7959** (2013) pp. 57–71. ⇒144, 146

[9] A. May, Secret exponent attacks on RSA-type schemes with moduli $N = p^r q$, *Public Key Cryptography (PKC 2004)*, *Lecture Notes in Computer Science* **2947** (2004) 218–230. ⇒144, 146

[10] A. May, Using LLL-reduction for solving RSA and factorization problems, *The LLL Algorithm, Phong Q. Nguyen and Brigitte Vallée, eds.*, *Information Security and Cryptography*, Springer Berlin Heidelberg, (2010) 315–348. ⇒144

[11] John M. Pollard, Theorems on factorization and primality testing, *Math. Proc. Cambridge Philos. Soc.* **76,** 3 (1974) 521–528. ⇒143

[12] John M. Pollard, A monte carlo method for factorization, *BIT* **15,** 3 (1975) 331–334. ⇒143

[13] C. Pomerance, The quadratic sieve factoring algorithm, *Advances in Cryptology (EUROCRYPT'84)*, *Lecture Notes in Computer Science* **209** (1985) 169–182. ⇒144

[14] J.J. Quisquater and C. Couvreur, Fast decipherment algorithm for RSA public-key cryptosystem, *Electronics letters* **18,** 21 (1982) 905–907. ⇒148

[15] R. L. Rivest, A. Shamir, and L. Adleman, A method for obtaining digital signatures and public-key cryptosystems, *Communications of the ACM* **21,** 2 (1978) 120–126. ⇒143

[16] S. Sarkar, Small secret exponent attack on RSA variant with modulus $N = p^r q$, *Des. Codes Cryptogr.* **73**, 2 (2014) 383–392. ⇒144, 146

[17] T. Takagi, Fast RSA-type cryptosystem modulo $p^k q$, *Advances in Cryptology (CRYPTO'98)*, *Lecture Notes in Computer Science* **1462** (1998) 318–326. ⇒147

Received: August 8, 2015 • Revised: October 26, 2015