An Attack on Small Private Keys of RSA Based on Euclidean Algorithm

March 3, 2019

Abstract

In this paper, we describe an attack on RSA cryptosystem which is based on Euclid's algorithm. Given a public key (n, e) with corresponding private key d such that e has the same order of magnitude as n and one of the integers $k = (ed - 1)/\phi(n)$ and $e - k$ has at most one-quarter as many bits as e, it computes the factorization of n in deterministic time $O((\log n)^2)$ bit operations.

MSC 2010: 94A60, 11T71, 11Y16.
Keywords: RSA cryptosystem ; Euclid's algorithm ; Integer factorization.

1 Introduction

Let p and q be two odd primes of the same size and $n = pq$. Consider integers e, d with $1 < e, d < \phi(n)$ such that $ed \equiv 1 \pmod{\phi(n)}$. Then (n, e) and d are the public and the private key, respectively, for a typical RSA public-key cryptosystem. The encryption and decryption algorithms are given by $C = M^e \mod n$ and $M = C^d \mod n$, respectively. In order to accelerate the operations involving the private key d in some devices, like for example a smart card, one might use a short secret exponent. On the other hand, in 1990, Wiener [11] proposed a polynomial time algorithm for breaking a typical RSA cryptosystem provided that $d < n^{1/4}/3$. In this case, d is the denominator of some convergent of the continued fraction expansion of e/n. The computation of the continued fraction expansion of e/n needs time $O((\log n)^2)$ bit operations and the total number of convergents is of order $O(\log n)$. Since Wiener's approach for testing convergents requires time $O((\log n)^2)$, the overall time complexity of Wiener's attack is $O((\log n)^3)$ bit operations. In [7, Section 5], Wiener's attack is presented as a bivariate linear equation problem and one can find d via a shortest vector computation in a two-dimensional lattice in time $O((\log n)^2)$.

Extensions of Wiener's attack that allows the RSA cryptosystem to be broken when d is a few bits longer than $n^{1/4}$ are described in [3, 4, 9, 10]. Furthermore, attacks based on Coppersmith's lattice-based technique for finding small roots of modular polynomials equations using LLL-algorithm are proposed in [1, 2] in case where e is very closed to n. These lattice attacks are applicable provided that $d < n^{0,292}$. Note however that these attacks are not rigorous and so this bound is not strictly proved.

In 2004, Hinek [6] proved that, in case where $\sqrt{6}(\phi(n)-d) < n^{1/4}$, Wiener's attack works. Furthermore, he showed that if the attacks in [1, 2] work for all $d < n^{\delta}$, then the attacks also work for $d > \phi(n) - n^{\delta}$.

In this note, we consider the case where the public exponent e has the same order of magnitude as n and one of the integers $k = (ed-1)/\phi(n)$ and $e-k$ has at most one-quarter as many bits as e. Using the equation $ed - k\phi(n) = 1$ and the extended Euclidean algorithm, we describe an efficient simple deterministic algorithm for the computation of the factorization of n in time $O((\log n)^2)$ bit operations. Although our attack appears to be equivalent to that of Wiener, it is simpler as it uses only the extended Euclidean algorithm and is easily presented in an undergraduate cryptography lesson.

The paper is organized as follows. In Section 2 we present our results and we describe our attack. Section 3 is devoted to the proof of our results. Finally, an example is given in Section 4.

2 An Attack Based on Euclidean Algorithm

Let p and q be two odd primes of the same size ℓ and $n = pq$. Consider positive integers e, d with $1 < e, d < \phi(n)$ such that $ed \equiv 1 \pmod{\phi(n)}$. Then (n, e) is the public key and d the private key for a RSA cryptosystem. Set $a = n + 1 \bmod e$ and $\Delta = \gcd(e, a)$. The extended Euclidean algorithm for e and a gives integers $q_i > 0$ $(i = 1, \ldots, m)$ and r_i $(i = 0, \ldots, m+1)$ such that $r_0 = e$, $r_1 = a$, $r_m = \Delta$, $r_{m+1} = 0$ and

$$r_{i-1} = r_i q_i + r_{i+1}, \quad 0 < r_{i+1} < r_i.$$

Further, there are integers s_i, t_i with $|t_i| < e/r_{i-1}$ and $|s_i| < a/r_{i-1}$ satisfying

$$s_i e + a t_i = r_i, \quad (i = 2, \ldots, m+1).$$

(See [8]). Set $\mu_i = \gcd(t_i, r_i)$ and $t'_i = t_i/\mu_i$ $(i = 0, \ldots, m+1)$. Our attack is based on the following result:

Theorem 1 *Let $e > n/c$, where c is an integer ≥ 1, and $k = (ed-1)/\phi(n)$. Suppose that k or $e-k$ is $\leq e^{1/4}/6\sqrt{c}$. Then, we have $\Delta < e^{3/4}$, and $k = |t'_j|$, $p + q = (a + |t'_j|^{-1}) \bmod e$ or $k = e - |t'_j|$, $p + q = (a + (e - |t'_j|)^{-1}) \bmod e$, respectively, where j is such that r_j is the larger remainder $< e^{3/4}$.*

Theorem 1 yields the design of the following deterministic algorithm for the computation of the factorization of n:

EUCLID-ATTACK
Input: A RSA public key (n, e) with $e > n/c$.
Output: The primes p and q.

1. Compute $a = (n+1) \bmod e$.
2. Using the extended Euclidean algorithm for e and a, compute the bigger remainder r_j among them which are $< e^{3/4}$ and the associated integers s_j, t_j such that $s_j e + a t_j = r_j$.
3. Compute $\mu_j = \gcd(t_j, r_j)$ and next $t'_j = t_j/\mu_j$.

4. Compute $\beta_1 = (a + |t'_j|^{-1}) \bmod e$ and next the solutions u_1 and v_1 of equation $X^2 - \beta_1 X + n = 0$. If u_1 and v_1 are positive integers, then output (u_1, v_1). Otherwise, go to the next step.

5. Compute $\beta_2 = (a + (e - |t'_j|)^{-1}) \bmod e$ and next the solutions u_2 and v_2 of equation $X^2 - \beta_2 X + n = 0$. If u_2 and v_2 are positive integers, then output (u_2, v_2). Otherwise, output FAIL.

Theorem 2 *Let $e > n/c$, where c is a positive integer, and $k = (ed-1)/\phi(n)$. Suppose that k or $e - k$ is $\leq e^{1/4}/6\sqrt{c}$. Then the above algorithm computes correctly the primes p and q in time $O((\log e)^2)$ bit operations.*

In order to avoid the attacks to small decryption exponent, a class of RSA encryption exponents e with corresponding $k = e - 1$ is analyzed in [5]. In this case the decryption exponent d is $\geq 2\phi(n)/3$. Since $k = e - 1$, Theorem 2 yields that the computation of the factorization of n, and so the computation of d, can be easily achieved.

Suppose now that $n/(c-1) > e > n/c$ with $c \geq 2$. We have:

$$\frac{d}{k} = \frac{ed}{ek} = \frac{k\phi(n)+1}{ek} < \frac{n-1}{e} + \frac{1}{ek} < c.$$

If $k \leq e^{1/4}/6\sqrt{c}$, then we obtain:

$$d < kc \leq \frac{\sqrt{c}\, e^{1/4}}{6} < \frac{\sqrt{c}}{6(c-1)^{1/4}}\, n^{1/4}.$$

Thus, for $c = 10$, we get $d < n^{1/4}/3$. On the other hand, we have:

$$k = \frac{ed-1}{\phi(n)} < \frac{ed}{\phi(n)} < \frac{2ed}{n} < \frac{2}{c-1}\, d.$$

If $d < n^{1/4}/12$, then we deduce:

$$k = \frac{ed+1}{\phi(n)} < \frac{ed}{\phi(n)} < \frac{2ed}{n} < \frac{e}{n^{3/4}6} < \frac{e^{1/4}}{(c-1)^{3/4}},$$

and so, for $c \geq 4$, we get $k \leq e^{1/4}/6\sqrt{c}$. Thus, we see that the efficacy of our approach is comparable with Wiener's method. Furthermore, as we have mentioned in the Introduction, Wiener's method needs $O((\log n)^3)$ bits operations while our approach $O((\log n)^2)$.

The solutions of the linear Diophantine equation $dx - y\phi(n) = 1$ are $x(T) = e + T\phi(n)$ and $y(T) = k + Td$, where $T \in \mathbb{Z}$. Consider now the inequalities:

$$6^4(k + Td)^4 < (e + T\phi(n)) \quad \text{and} \quad 6^4(e - k + T(\phi(n) - d))^4 < (e + T\phi(n)).$$

We remark that for T large enough, for instance $T > \phi(n)^{1/3}$, the above inequalities are not satisfied. Thus, in case that we replace the pubic key e by $x(T) = e + T\phi(n)$ with $T > \phi(n)^{1/3}$ our attack does not work. Note that Wiener's attack is not guaranteed to work if $e > n^{1.5}$ and the attack of [1] is effective as long as $e < n^{1.875}$.

3 Proof of Theorems 1 and 2

Proof of Theorem 1. First, we give the proof of Theorem 1. The equalities $ed - k\phi(n) = 1$ and $\phi(n) = n - (p+q) + 1$ yield:

$$ed - 1 = k(n - (p+q) + 1),$$

whence, we obtain:

$$k(n + 1 - (p+q)) + 1 \equiv 0 \pmod{e}.$$

Setting $y_0 = k$ and $x_0 = p + q$, we get:

$$1 + ay_0 - x_0y_0 \equiv 0 \pmod{e}.$$

Suppose that $p < q$. Then $p < \sqrt{n}$. Since p and q have the same size ℓ, we have:

$$2^{\ell-1} + 1 \leq p < q \leq 2^{\ell-1} + \cdots + 1.$$

Thus, we get:

$$q - p \leq 2^{\ell-2} + \cdots + 2 < 2^{\ell-1} + 1 \leq p,$$

whence we obtain $q < 2p$. Therefore, we have:

$$x_0 = p + q < 3\sqrt{n} < 3\sqrt{ce}.$$

Suppose that $y_0 \leq e^{1/4}/6\sqrt{c}$. If $\Delta \geq e^{3/4}$, then we have $x_0y_0 \equiv 1 \,(\mathrm{mod}\, \Delta)$ and

$$|x_0y_0 - 1| < e^{3/4} \leq \Delta.$$

It follows that $x_0y_0 = 1$, whence we get $x_0 = y_0 = 1$ which is a contradiction. Hence $\Delta < e^{3/4}$. Let r_j be the bigger among the remainders which are $< e^{3/4}$. Then, we have $r_{j-1} > e^{3/4}$ and $|t_j| < e/r_{j-1} < e^{1/4}$. Further, we have:

$$t_j(1 + ay_0 - x_0y_0) + s_jey_0 \equiv 0 \,(\mathrm{mod}\, e),$$

whence we get:

$$0 \equiv t_j + (t_ja + s_je)y_0 - t_jx_0y_0 \equiv t_j + r_jy_0 - t_jx_0y_0 \,(\mathrm{mod}\, e).$$

Set $f(x, y) = t_j + r_jy - t_jxy$. Then, we have $e \mid f(x_0, y_0)$ and

$$|f(x_0, y_0)| < e^{1/4} + \frac{e}{6\sqrt{c}} + \frac{e}{2} < e.$$

It follows that $f(x_0, y_0) = 0$, and so, we obtain:

$$t'_j + r'_jy_0 - t'_jx_0y_0 = 0.$$

It follows that $t'_j \mid r'_jy_0$. Since $\gcd(t'_j, r'_j) = 1$, we obtain $t'_j \mid y_0$. Furthermore, the above equality implies that $y_0 \mid t'_j$. Therefore, we have $y_0 = |t'_j|$. Thus, the congruence $1 + ay_0 - x_0y_0 \equiv 0 \pmod{e}$ yields $x_0 = (a + |t'_j|^{-1}) \bmod e$.

Set $z_0 = -y_0 \bmod e$. Suppose that $z_0 \leq e^{1/4}/6\sqrt{c}$. If $\Delta \geq e^{3/4}$, then we deduce $1 + x_0z_0 \equiv 0 \,(\mathrm{mod}\, \Delta)$ and

$$|x_0z_0 + 1| < 1 + \frac{e^{3/4}}{2} < \Delta.$$

It follows that $x_0 z_0 + 1 = 0$ which is a contradiction. Thus, we get $\Delta < e^{3/4}$. Now, working as previously, we have:

$$1 - az_0 + x_0 z_0 \equiv 0 \pmod{e}$$

and we deduce that $z_0 = |t'_j|$. Therefore $e - y_0 = |t'_j|$. It follows that $x_0 = (a + (e - |t'_j|)^{-1}) \bmod e$.

Proof of Theorem 2. The proof of correctness of the algorithm EUCLID-ATTACK is a simple consequence of Theorem 1. We shall compute its time complexity following [8]. The execution of the extended Euclidean algorithm in Step 2 needs $O((\log e)^2)$ bit operations. The computation of δ and t'_j in Step 3 requires $O((\log e)^2)$ bit operations. Similarly, the computation of b_1 and b_2 needs $O((\log e)^2)$ bit operations. Finally, the solution of the quadratic equations in Steps 4 and 5 requires also $O((\log e)^2)$ bit operations. Therefore the time complexity of the algorithm EUCLID-ATTACK is $O((\log e)^2)$ bit operations.

4 A Toy Example

In this section we give an example of application of our algorithm. Let

$$p = 9223372036854777017 \quad \text{and} \quad q = 9224497936761618437$$

be two 64-bits primes. Their product is the number

$$n = 850809763239516967196355785796710624 29.$$

We compute:

$$\phi(n) = (p-1)(q-1) = 850809763239516967011877086060546 66976.$$

We select:

$$d = \phi(n) - 2^{22} - 2^{14} - 2^6 - 2^3 - 1 = 850809763239516967011877086060504562 15$$

and compute:

$$e = d^{-1} \bmod \phi(n) = 611005594062514632567097160703021510 15.$$

Thus (n, e) and d is the public and private key for a RSA scheme. We shall use the algorithm EUCLID-ATTACK in order to compute the factorization of n.

First, we compute

$$a = (n+1) \bmod e = 239804169177002334629258625093689114 15.$$

We apply the Euclidean algorithm for $r_0 = e$ and $r_1 = a$, and we compute the remainders $r_2, r_3, \ldots$. The bigger remainder which is smaller than $e^{3/4}$ is

$$r_{13} = 55785270375887536485564215.$$

The corresponding pair (s_{13}, t_{13}) is the pair $(-1186820, 3023941)$. Further, we have $\gcd(r_{13}, t_{13}) = 1$. Following the steps of the algorithm, we compute:

$$b_1 = a + t_{13}^{-1} \bmod e = 479608338354004669074038550451214 27376.$$

We solve the equation $x^2 - b_1 x + n = 0$ and we see that their solutions are not integers. Next, we compute

$$b_2 = a + (e - t_{13})^{-1} \mod e = 18447869973616395454.$$

The solutions of the equation $x^2 - b_2 x + n = 0$ are the primes p and q. Note that $2e > n$ and so, $c = 2$. Furthermore, we have $n - k < e^{1/4}/6\sqrt{2}$.

References

[1] J. Blömer and A. May, Low secret exponent RSA revisited. In: Cryptography and lattice. Proceedings of CaLC 2001. Lecture Notes in Computer Science, vol. 2146, (2001), 4-19

[2] D. Boneh and G. Durfee, Cryptanalysis of RSA with Private Key d Less Than $N^{0.292}$, *IEEE Trasanctions on Information Theory,* 46, 4, (2000), 1339-1349.

[3] A. Dujella, Continued fractions and RSA with small secret exponent, *Tatra Mt. Math. Publ.* 29 (2004), 101-112.

[4] A. Dujella, A variant of Wiener's attack on RSA, *Computing* 85 (2009), 77-83.

[5] L. Hernádez Encinas, J. Munoz Masqué and A. Queiruga Dios, Large Decryption Exponents in RSA, *Applied Mathematics Letters,* 16 (2003) 293-295.

[6] M. J. Hinek, (Very) large RSA private exponent vulnerabilities. CACR Technical Report CACR 2004-01, Centre for Applied Cryptographic Research, University of Waterloo, 2004. [http://www.cacr.math.uwaterloo.ca/]

[7] A. May, Using LLL-Reduction for Solving RSA and Factorization Problems, P.Q. Nguyen and B. Vallée (eds.), The LLL Algorithm, Information Security and Cryptography, Springer-Verlag, Berlin Heidelberg 2010, pp. 315-348.

[8] V. Shoup, *A Computational Introduction to Number Theory and Algebra,* Second Edition, Cambridge University Press 2008.

[9] H. M. Sun, M. E. Wu and Y. H. Chen, Estimating the prime-factors of an RSA modulus and an extension of the Wiener attack. In: Applied cryptography and network security. Lecture Notes in Computer Science, vol 4521, (2007), 116-128

[10] E. R. Verheul and H. C. A. van Tilborg, Cryptanalysis of 'less short' RSA secret exponents, *Appl. Algebra Eng. Comm. Comput.,* 8, (1997), 425-435.

[11] M. J. Wiener, Cryptanalysis of short RSA secret exponents, *IEEE Trasanctions on Information Theory,* 36 (1990), 553-558.