A NEW ATTACK TO RSA WITH SMALL PRIVATE EXPONENT AND PARTIAL INFORMATION

JORGE JIMÉNEZ URROZ

Abstract. We give a new algorithm to attack RSA with small private exponent, when some partial information of $p+q$ is given. The algorithm is a very simple modification of the original Wiener's attack with continued fractions, and allows us to factor n whenever $d < n^{\frac{1+\delta}{4}}$ if we know a δ-fraction of the most significant bits of $p+q$. The algorithm is unconditional, which is not the case in previous improvements that use Coppersmith method. As an example, our algorithm can be applied to break any cryptosystem with modulus n of 512 bits and $d < n^{0.3}$, given an improvement in the original attack of Wiener.

1. Introduction

Nowadays the organization of society is based on digital communication and over 6000 million people are connected. The efficiency and security needed for encryption or digital signature on most of the protocols like SSH, PGP, OpenPGP, SSL/TLS, software, browsers etc, rely on RSA and since it was created in 1978 many papers have appeared trying to break it in very different scenarios.

To build RSA we need a public key an a secret key. The public key is made by two integers n, e, where n is the product of two unknown primes p, q, and e is coprime to $\varphi(n) = (p-1)(q-1)$ the Euler totient function. The secret key consist in the two unknown primes p, q and the integer d inverse of e modulo $\varphi(n)$. Most of the attacks to RSA are based on this fact and use the identity

$$ed - k\varphi(n) = 1, \tag{1}$$

valid for some integer k, to get information on the secret exponent d and the factors of n.

When the difference in computer power between sender and receiver is high, we need to make the private and public exponents d, e small, to make feasible computation in the lighter device. For example in a communication between a smart card and a computer, the smart card

needs a short d to decrypt, while the computer needs to have a short e, so the smart card can encrypt. Nowadays is known that the smaller are those exponents, the more insecure is the protocol. A simple example comes with the following scenario: suppose we take the public exponent to be $e = 3$ and send the message to three different people with the same exponent. Then we would have $c_1 = m^3 \pmod{n_1}$, $c_2 = m^3 \pmod{n_2}$ and $c_3 = m^3 \pmod{n_3}$ and then we just have to apply the chinese remainder theorem to get $m^3 \equiv c \pmod{n_1 n_2 n_3}$ but $m^3 < n_1 n_2 n_3$ so $m^3 = c$ and $m = c^{1/3}$.

But there is a lot more to study on small exponents. Almost 40 years ago Wiener proved in [27], using continued fractions in a clever way on Equation (1), that small private exponents $d < cn^{1/4}$ for some constant c are insecure, and the attacker can recover the secret key in polynomial time. That paper was seminal and since then, until today, there is a strong interest in discovering attacks to break RSA for exponents d of bigger size. We can find several papers trying to improve even the constant c, up to the current record of $c = \frac{1}{18^{1/4}}$, given by Susilo et al. [22] in 2019. More information on the constant can be found in [1, 3, 20]. The authors of [22] also gave an example of $d \leq \frac{1}{2}n^{1/4} + 1$ in which the method does not break the RSA. This restricts the value of the constant to $c \in (\frac{1}{2.06}, \frac{1}{2}]$ but also proves that the exponent $1/4$ is the best one can hope, using the method as it was proposed by Wiener.

There are other variants of Wiener's attack, [5,9,10,26] that allow to break the RSA cryptosystem even if d is few bits longer than the previous bound. The references [9, 10, 26] are based on an identity of the form $d = rb_{m+1} + sb_m$ where b_m, b_{m+1} are denominators of the convergents of e/n, while the first, [5], uses another integer n' derived from n and construct the convergents of e/n'. Concretely considering the integer $n' = \left\lfloor n - \left(1 + \frac{3}{2\sqrt{2}}\right) n^{1/2} + 1 \right\rfloor$ they get up to $d^2 e < 8n^{3/2}$. Observe that for $e \sim n$ this increases Wiener's bound up to $d < 2\sqrt{2}n^{1/4}$. Very recently Trishing proved in [25] that a variant of the attack also is able to break the RSA for large exponents d, and gives a proportion of $n^{-1/2}$ of weak exponents.

Then, the paper by Boneh-Durfee [3] comes, and strikes the community showing an improvement in the exponent, and gets up to $d < n^{0.292}$, at least in theory. Their method has changed completely from Wiener's and creates another school to attack RSA with small private exponent.

Their tool is the powerful theorem of Coppersmith [6] to find small solutions to polynomial equations. Namely they consider $\varphi(n) = A+y$ for $A = (n+1)/2$ known, and $y = -(p+q)/2 < n^{1/2}$ unknown, in Equation (1), and apply Coppersmith method to

$$f(x,y) = x(A+y) - 1 \equiv 0 \pmod{e},$$

getting a solution for $|x| < n^{\delta}$ where $\delta = 1 - \frac{\sqrt{2}}{2} \approx 0.292$.

Their result is asymptotic and to reach the previous value the dimension of the lattice needed makes the attack infeasible. In practice, they get only to exponents of order 0.28 for moduli of 1000 bits, while 0.255 for moduli of 10000 bits. There are other improvements of the method, like [12, 13, 16, 23], getting better results in practice, but not reaching $d < n^{0.292}$ which remains the state of the art nowadays despite all the efforts.

But there are other common ways of attackng RSA, assuming that the attacker could have an extra information on the secret key. This information normally comes from certain side channel attacks, in the presence of convert channels, [7, 15]. The first of such instances was in [4] where the authors, also with lattices, recover the secret key knowing only $\log d/4$ of the least significant bits of d. In [21] the authors prove that the running time in [4] depends exponentially on the least significant bits, LSBS for short, shared by p and q. Other works as [19], guessing some of the most significant bits, MSBs for short, of one factor of n reduce the information needed on d, while [11] improves the bounds by some refinement of the lattices. Other works using Coppersmith method and partial information on the secret key can be found in [18, 24, 28]. In [28] the authors go a bit further and use Coppersmith method somehow combined with continued fractions, to get a general exponent depending on the information we know about e, d and the MSBs of $p+q$. Concretely if $e < N^{\alpha}$ and we know a proportion of $1/2-\gamma$ of the MSBs of $p+q$ one can recover any key $d < 1-\alpha/3-\gamma/2$. A generalization of the method to equations of the form $ed - 1 \equiv 0 \pmod{(p^a-1)(q^a-1)}$ appeared very recently in [29].

On the other hand, using only continued fractions, De Weger's [8] uses the most significant bits, MSBS for short, to prove that for primes with $|p-q| < n^{\beta}$, Wiener's attack will go up to $d < n^{3/4-\beta}$ and [2] combine Wiener's attack with Coppersmith's original theorem to factor any modulus such that there is a linear relation of the form $ex + y \equiv 0 \pmod{\varphi(n)}$ with x and y small. The authors in [17] knowing and

approximation of p of order $n^{\alpha}/8$ MSB's get up to $d < n^{(1-\alpha)/2}$ on Wiener's attack, by using convergents of different integers as in [5].

It is of fundamental importance to remark now that even though the results using Coppersmith method are stronger, all of them depend on a hypothesis which works in practice, but never has been proved. Concretely the algebraic independence of the solutions given by the reduction algorithm in the lattice. On the other hand, results based on continued fractions are unconditional.

Our contribution. The paper is dedicated to attack RSA, find the private key and factor the modulus, when extra information on the factors is leaked. We will only use continued fractions and hence all the results are unconditional. In our main theorem we prove that knowing an approximation S of $p+q$, so that $p+q-S < n^{(1-\delta)/2}$, is enough to get the private exponent whenever $d < n^{\frac{1+\delta}{4}}$.

We dedicate a section to prove that an approximation of $p+q$ is equivalent to knowing an approximation of p and q separately, and it is in fact another way of expressing the knowledge of the MSBs of the corresponding integer. In this way we can recover the results in [17], in terms of approximations of $p+q$. We get the same result as they when $e \approx n$, but removing also the constant $1/8$ from that paper, and generalize it to any range of e. Also, we need to emphasize that our method is extremely simple in comparison with them, which makes computation much faster than [17], since we only consider the computation of the convergents of a precise fraction.

The estimation in our main result is worse than in [28]. However, we should mention that their results, apart from being conditional to finding algebraically independent solutions on the lattice reduction, also use an extra hypothesis which seems unlikely to occur in certain range of the parameters that they consider. Concretely their results are based on the size of the inverse of a particular integer q_r modulo q_{r-1}, which normally will be of the same size as the modulus.

We should mention that the authors themselves mention that they find a gap between their experiments and their theoretical results, without explaining why.

In Section 2 we include the necesary preliminaries on continued fractions. Then in Section 3 we state and prove the results. Section 4 is dedicated to analyze the relation between approximation of an integer and thier MSBs, for $p+q$ and p and q. Finally we include some experiments in Section 5. Throughout the paper, given two integers a, b $[a, b]$ will denote thier least common multiple while (a, b) will be their greatest common divisor.

2. Preliminaries

To attack RSA with small d we will use continued fractions.

Definition 2.1. *Given a real number α the continued fraction of α is an expression of the form*

$$\alpha = q_0 + \cfrac{1}{q_1 + \cfrac{1}{q_2 + \cfrac{1}{q_3 + \cfrac{1}{q_4 + \cfrac{1}{q_5 + \cfrac{1}{q_6 + \cfrac{1}{q_7 + \cfrac{1}{q_8 + \cdots}}}}}}}}$$

where $q_0 = [\alpha]$, $r_0 = \{\alpha\}$, $q_i = [1/r_{i-1}]$, $r_i = \{1/r_{i-1}\}$ for $i \geq 1$, until $r_m = 0$, which will occur exactly when $\alpha \in \mathbb{Q}$. In this case the continued fraction is finite and we denote it as $\alpha = \langle q_0, q_1, \ldots, q_m \rangle$

Remark 2.2. *Note that $q_m \geq 2$ since otherwise $r_{m-1} = 1$ which is impossible.*

We can construct the continued fraction in the following way. Let $\frac{n_i}{d_i} = \langle q_0, \ldots, q_i \rangle$. Then one can prove [14, Theorem 1] that

$$\begin{array}{lll} n_0 = q_0, & d_0 = 1 & \\ n_1 = q_0 q_1 + 1, & d_1 = q_1 & \\ n_i = q_i n_{i-1} + n_{i-2}, & d_i = q_i d_{i-1} + d_{i-2}, & \text{for any } 2 \leq i \leq m, \end{array} \tag{2}$$

and [14, Theorem 2]

$$n_i d_{i-1} - d_i n_{i-1} = (-1)^{i+1} \quad \text{for any } 1 \leq i \leq m. \tag{3}$$

This in particular implies

$$(n_i, d_i) = 1, \tag{4}$$

for all $0 \leq i \leq m$.

We will also need to control the growth of the convergents. Concretely we have

Theorem 2.3. *Let $\alpha = \langle q_0, \ldots, q_m \rangle$ and let $\frac{n_i}{d_i}$ its convergents for $i = 0, \ldots, m$. Then for any $i \geq 4$,*

$$n_i > 2^{\frac{i-1}{2}}.$$

Proof. We will prove it by induction. We know that for $i \geq 1$, $q_i \geq 1$, and $n_i \geq 1$. Hence, noting that $n_3 \geq n_2 + n_1 \geq 2$, we get the first two cases $n_4 \geq n_3 + n_2 \geq 3 > 2^{3/2}$ and $n_5 \geq n_4 + n_3 \geq 5 > 2^{4/2}$. Now, suppose $n_{i-2} > 2^{\frac{i-3}{2}}$. Then, by (2) for $i \geq 6$

$$n_i = q_i n_{i-1} + n_{i-2} \geq n_{i-1} + n_{i-2} \geq n_{i-2} + n_{i-3} + n_{i-2} > 2n_{i-2} > 2^{(i-1)/2}.$$

□

Remark 2.4. *See [14, Theorem 12] where the inequality is proved for the denominators of the convergents. We did not find a reference including the previous inequality for the numerators.*

The key fact on Wiener's method to attack RSA comes with the following lemma.

Lemma 2.5 (Wiener)**.** *Let $\alpha = \langle q_0, \ldots, q_m \rangle$, and $\alpha' \in \mathbb{Q}$ such that $\alpha = \alpha'(1 - \delta)$ for $\delta < \frac{2}{3n_m d_m}$, as in (2). Then one of the convergents of α is α'.*

Proof. See [27, p. 555, (22)]. □

We end this section with a simple lemma on the Euler function. We include the proof for convenience.

Lemma 2.6. *Let $n = pq$ with $q < p < 2q$. Then*

$$n + 1 - \frac{3}{\sqrt{2}}\sqrt{n} < \varphi(n) < n + 1 - 2\sqrt{n} \tag{5}$$

Proof. The function $f(p) = p + \frac{n}{p}$ is increasing for $p > \sqrt{n}$, so under the hypothesis we get the lower bound $\sqrt{n} \leq p$ and, multiplying by p in $p < 2q$, the upper bound $p^2 < 2n$, so $2\sqrt{n} \leq f(p) \leq \frac{3}{\sqrt{2}}\sqrt{n}$. The result follows by noting that $\varphi(n) = n + 1 - f(p)$. □

3. Wiener's approach revisited

Theorem 3.1. *Let n be an RSA modulus $n = pq$, with prime factors $q < p < 2q$, and suppose $p + q - 1 = x + R$ where x is known. Let e be an integer coprime with $\varphi(n)$ and $d = e^{-1} \pmod{\varphi(n)}$, verifying (1). Then, if $d < (\sqrt{2/3}n - \sqrt{3n})(eR)^{-1/2}$ we can factor n in polynomial time in $\log(en)$.*

Proof. Dividing by $d(n-x)$ in (1) we get

$$\begin{aligned} \frac{e}{n-x} &= \frac{1}{d(n-x)} + \frac{k}{d}\frac{(p-1)(q-1)}{(n-x)} = \frac{k}{d}\left(\frac{n-p-q+1+\frac{1}{k}}{n-x}\right) \\ &= \frac{k}{d}\left(1 - \frac{R-\frac{1}{k}}{n-x}\right). \end{aligned} \tag{6}$$

Hence, by Lemma 2.5 if $\frac{R-\frac{1}{k}}{n-x} < \frac{1}{3/2kd}$, or equivalently

$$kd < \frac{2}{3}\frac{n-x}{R-1/k}, \tag{7}$$

one of the convergents of $\frac{e}{n-x}$ is $\frac{k}{d}$. Again Equation (1) gives trivially $ed^2 = kd\varphi(n) + d > kd\varphi(n)$, or $kd < \frac{ed^2}{\varphi(n)}$, so in particular if we have $d < (\sqrt{\frac{2}{3}}n - \sqrt{3n})(eR)^{-1/2}$, then using Lemma 2.6 we get the inequality $d < \sqrt{2/3}\varphi(n)(eR)^{-1/2}$ so

$$kd < \frac{ed^2}{\varphi(n)} < \frac{2}{3}\frac{\varphi(n)}{R} < \frac{2}{3}\frac{n-x}{R-1/k},$$

and hence one of the convergents of $\frac{e}{n-x}$, say $\frac{n_i}{d_i}$ is $\frac{k}{d}$. Noting that $(n_i, d_i) = (k, d) = 1$ by (4) and (1), we get $k = n_i$ and $d = d_i$. Then, we compute $\varphi(n)$ by (1), and from there the factors p, q.

By Theorem 2.3, $e = n_m > 2^{(m-1)/2}$, so we need at most $O(\log e)$ steps to find $\frac{k}{d}$, and on each step we perform one multiplication and one addition by 2 with integers bounded by $e, n-x$, since the sequence of numerators and denominators are increasing with limit e, and $n-x$ respectively. □

(8) **Algorithm**

Input: e, n, x
Output: d, p, q

for $i = 1$ until m we:

Compute n_i, d_i the convergents of $\frac{e}{n-x}$.
Compute $L = \frac{ed_i - 1}{n_i}$. This is our candidate of $\varphi(n)$
Solve $X^2 - (n - L + 1)X + n = 0$ to get the roots x_1, x_2
If $x_1, x_2 \in \mathbb{N}$ and $x_1 x_2 = n$
  return (d_i, x_1, x_2)

Remark 3.2. *We should emphisize that the extra information needed on $p+q$ would be unnecessary if the size of the modulus is not too big. In particular since the algorithm runs in $O(\log en)$, one could repeat it $O(\sqrt{n}/2^l)$ times with a guess of $x = 2^l A$ for $2\sqrt{n}/2^l \leq A < 3/\sqrt{2}\sqrt{n}/2^l$ and increasing A by 1 each time. This would give an algorithm running on $O(\sqrt{n}/2^l \log(en))$, which still is feasible for l large enough. For example if n is a modulus of 512-bits, then $l = 200$ would mean that we repeat the algorithm 2^{56} times, and we capture 56 bits of $p+q$ which is more than $1/10$ of the bits of n, and we could factor n whenever $d < 0.3$ with no previous information at all on the primes factors of n.*

4. δ-APPROXIMATIONS

In this section we establish, as we mentioned in the introduction, that knowing an approximation of p and q is equivalent to knowing an approximation of $p+q$. Before proving it, we will make a concrete definition for convenience in the exposition. Along the whole section we will denote all the constants by c, or C, but on each step of the argument could have different values.

Definition 4.1. *Let m be an unknown integer and $\delta > 0$. We say that the integers K, U are a δ-approximation of m if $m = K + U$, K is known and $|U| < cm^{1-\delta}$ for some constant c.*

We include two straighforward lemmas for convenience.

Lemma 4.2. *Having a δ-approximation of an integer m is equivalent to knowning the MSBs of m.*

Proof. Concretely, if we know the L MSBs of an integer m, then we can write $m = 2^l A + B$, where $l = [\log_2 m] - L$, A is a known integer and $0 \leq B < 2^l < m^{1 - \frac{L}{\log_2 m}}$, so we have a $\frac{L}{\log_2 m}$-approximation of m. On the other hand, if $m = K + U$ with $|U| < cm^{1-\delta} < 2^l$, then we have the inequalities $-1 + \frac{K}{2^l} < \frac{m}{2^l} < \frac{K}{2^l} + 1$, so

$$\left[\frac{m}{2^l}\right] = \left[\frac{K}{2^l}\right] - \varepsilon,$$

where $\varepsilon = 1$ if $U < 0$, while $\varepsilon = 0$ if $U > 0$. Hence,

$$m = \sum_{j=0}^{[\log_2 m]} m_j 2^j = 2^l \left[\frac{m}{2^l}\right] + \sum_{j=0}^{l} m_j 2^j = 2^l \left(\left[\frac{K}{2^l}\right] - \varepsilon\right) + B,$$

where $0 \leq B < 2^l$, and we know the $[\log_2 m] - l$ MSBs of m. □

Lemma 4.3. *Let the public integer $n = pq$ be the product of two unknown prime numbers $q < p < 2q$ large enough. Having a δ-approximation of either p or q is equivalent to having a δ-approximation of both p and q.*

Proof. Obviously we just have to prove one implicaiton. Now, note that $n = pq < \min\{p^2, 2q^2\}$. Then If $p = K_p + U_p$ for $|U_p| < Cp^{1-\delta}$, then $|U_p| < cp$, for some $c < 1$ and p large enough so $K_p > (1-c)p$ and

$$\left|\frac{n}{K_p} - \frac{n}{p}\right| = n\frac{|p - K_p|}{pK_p} < \frac{C}{1-c} qp^{-\delta} < \frac{C}{1-c} q^{1-\delta},$$

so $q = K_q + U_q$ where $K_q = \frac{n}{K_p}$ and $|U_q| < \frac{C}{1-c} q^{1-\delta}$. On the opposite, If we start knowing $q = K_q + U_q$ for $|U_q| < Cq^{1-\delta}$, the reasoning is analogous.

□

We now can prove our main result of apprximations.

Proposition 4.4. *Let the public integer $n = pq$ be the product of two unknown prime numbers $q < p < 2q$. Suppose $p + q = K + U$, is a δ-approximation of $p+q$ with $K \geq \kappa\sqrt{n}$ for some $\kappa > 2$. Then one can find a δ-approximation of p and a δ-approximation of q in polynomial time, and viceversa.*

Proof. Suppose we know a δ-approximation of each p and q, say $p = K_p + U_p$ for $|U_p| < c_p p^{1-\delta}$ and $q = K_q + U_q$ for $|U_q| < c_q q^{1-\delta}$, and denote $C = \max\{c_p, c_q\}$. Then we have $p + q = K_{p+q} + U_{p+q}$ where $K_{p+q} = K_p + K_q$ and

$$|U_{p+q}| = |U_p + U_q| \leq |U_p| + |U_q| < C(p^{1-\delta} + q^{1-\delta}) < c\,(p+q)^{1-\delta},$$

for some constant c, using $p < 2(p+q)/3$, $q < (p+q)/2$ by the conditions on the primes.

Now suppose $p+q = K_{p+q} + U_{p+q}$ where K_{p+q} is a known integer with $K_{p+q} > \kappa\sqrt{n}$, for some constant $\kappa > 2$ and $|U_{p+q}| < C(p+q)^{1-\delta}$ for some constant C. By Lemma 4.3 it is enough to have a δ-approximation of

one of the primes, say p. Then solving the equation $x^2-(p+q)x+n=0$ with solutions p, q, we get

$$\begin{aligned} p &= \frac{K_{p+q}+U_{p+q}+\sqrt{(K_{p+q}+U_{p+q})^2-4n}}{2} \\ &= K_p + U_p \end{aligned}$$

where

$$K_p = \frac{K_{p+q}}{2} + \frac{\sqrt{K_{p+q}^2-4n}}{2},$$

and

$$U_p = \frac{U_{p+q}^2+2K_{p+q}U_{p+q}}{2\left(\sqrt{(K_{p+q}+U_{p+q})^2-4n}+\sqrt{K_{p+q}^2-4n}\right)} + \frac{U_{p+q}}{2}$$

Now, by the conditions on the primes, $C_1 p \leq K_{p+q} < C_2 p$, for some constants C_1, C_2, , so $U_{p+q}^2 < |K_{p+q}U_{p+q}| < Cp^{2-\delta}$, for some constant C, while

$$\begin{aligned} &\sqrt{(K_{p+q}+U_{p+q})^2-4n}+\sqrt{K_{p+q}^2-4n} > \sqrt{K_{p+q}^2-4n} \\ &> (\sqrt{c^2-4})n^{1/2} > Cp, \end{aligned}$$

for some constant C, so

$$U_p < C(p^{1-\delta}+(p+q)^{1-\delta}) < cp^{1-\delta}.$$

for some constants C, c. The δ-approximation then will be $p = K + U$ where $K = [K_p]$ and $U = K_p - K + U_p$. □

Remark 4.5. *If the δ-approximation of $p+q$ is with any K with no lower bound, then the same reasoging gives a $\delta/2$-approximation of p and q. In this case we should use*

$$\begin{aligned} p &= \frac{K_{p+q}+U_{p+q}+\sqrt{(K_{p+q}+U_{p+q})^2-4n}}{2} \\ &= K_p + U_p \end{aligned}$$

where

$$K_p = \frac{K_{p+q}}{2} + \frac{\sqrt{K_{p+q}^2-4n}}{2},$$

and

$$\begin{aligned} U_p &= \frac{U_{p+q}^2 + 2K_{p+q}U_{p+q}}{2\left(\sqrt{(K_{p+q}+U_{p+q})^2 - 4n} + \sqrt{K_{p+q}^2 - 4n}\right)} + \frac{U_{p+q}}{2} < \\ &< \sqrt{U_{p+q}^2 + 2K_{p+q}U_{p+q}} < cp^{1-\delta/2}. \end{aligned}$$

for some constant c.

5. Experiments

All the experiments have been done in python using a personal computer, concretely a Mac book pro with 2GHz intel core i5 of 4 kernels and 16GB of memory. The program used is included in the Appendix 7

Example 5.1. *Let* $n = 123543221$*, and* $e = 3782461$*.* $n^{1/4} = 105.4277$*, so assume* $p + q = 22130 + R$*, where* $R < 106$*. So we consider the continued fraction of* $\frac{e}{n-x} = \frac{3782461}{123521091}$ *to get as its list of convergents*

$$\begin{aligned} &[(0,1),(1,32),(1,33),(2,65),(3,98),(32,1045),\\ &(1187,38763),(4780,156097),(10747,350957),\\ &(15527,507054),(57328,1872119)] \end{aligned}$$

When reaching the 5*-th convergent, multiply by* e *and take the integer part, we get* 123520992 *and quotient* 1*. Letting*

$$123543221 - 123520992 = 22229 = a,$$

and solving the equation

$$x^2 - 22230x + n,$$

we get $p = 11117, q = 11113$*. In this case* $d > n^{0.373}$*.*

Example 5.2. *The next example is used by the authors of [28]. They need more than* 9 *seconds to factor* n*. This algorithm takes miliseconds.*

$$\begin{aligned} n &= 49744727738813841180146477957115475164782599566456139376718994215049702937974938606611020238731850953950803408792425499240178917020631382829111999892519 77 \\ e &= 33621352126461953737326989940596918224914003196444922846372466250218118962134587236033368546938732470674860097552863597697728273979959783625450378784644 33 \\ x &= 142791762276831471380447927282999667498 21 \times 10^{37} \end{aligned}$$

returns

$$43872548878645166167257647442781990528234 88,$$
$$64911964009273407289381552277793047324037 93,$$
$$86$$

whose second coordinate is the secret exponent on the 86 *convergent, in miliseconds. In this case the exponent is* $d \approx n^{0.2785}$.

Taking

$$\begin{aligned} e \;=\; & 44034070168020900473938734992781610875178561779832070 67 \\ & 72482977453082570303456717221333307551794116401184549527 69 \\ & 35054951436828450021261154418374318704225 \end{aligned}$$

the program returns

$$57460119048920727635866412899612690479140804992453674733 4,$$
$$64911964009273407289381552277793047324037939485723459871 3,$$
$$110$$

whose second coordinate is the secret exponent on the 110 *convergent, also miliseconds. In this case the exponent is* $d \approx n^{0.369}$.

Example 5.3. *This is also an example used by the authors of [28]. While they need* 17 *digits of the sum of the two primes, to get the exponent* $d \approx n^{0.317}$*, we get the same with* 21 *digits. Observe that the* 4 *digits of difference can be obtained by repeating the algorithm* 10^4 *times.*

Now take

$$\begin{aligned} n \;=\; & 66788128980215403882362868061483689535214707015713535 9 \\ & 51737078614658556966380576759897646994836110233167560 0 \\ & 61753051495205720784884950001868448113996412 03 \\ e \;=\; & 48193961975298196910970934752997240112221576460690017 4 \\ & 83451762610740269784984650141656921535710014388796307 3 \\ & 43880034526705888094113389016105675715221750 99 \\ x \;=\; & 10^{25}y \text{ where} \\ y \;=\; & 16637133402080618303507029737134206559438091398368285 \end{aligned}$$

6. Conclusion

The main objective of the paper is to establish an unconditional improvement on the size of the secret exponent which is vulnerable to Wiener's attack with continued fractions, knowing partial information

on $p+q$. We also prove that knowing this information is equivalent to knowing partial information on p and q. We observe that up to 60 bits, this information could be found by repeating the algorithm until succeed, which would happen in polynomial time. In particular if n is a modulus of 512-bits, then running the algorithm 2^{56} times , with consecutive brute force guesses of x, we could factor n whenever $d < 0.3$.

7. Appendix

The program first compute the gcd of two integer numbers, collecting the quotients on each step with the following instructions in python:

Computing the gcd

```
def mcd(a,b):
a,b=max(a,b),min(a,b)
q=[]
while b!=0:
q.append(a//b)
c=a%b
a=b
b=c
return q
```

Then, given the quotients, we compute the sequence of numerators and denominators of the convergents by the formulas in (2). For each convergent, we try to factor n by guessing k, d as numerator and denominator of the convergent, with the Algorithm 8, with the following instructions:

```
q=mcd(n-x,e)
num,den=[q[0],q[0]*q[1]+1],[1,q[1]]
for j in range(2,len(q)):
num.append(q[j]*num[j-1]+num[j-2])
den.append(q[j]*den[j-1]+den[j-2])
for l in range(len(num)):
  L = (e * den[l] - 1)/num[l].
  import sympy as sym
x=sym.symbols('x')
eq = sym.Eq(x**2 - (n - L + 1) * x + n, 0)
roots = sym.solve(eq, x)
if roots[0].is_integer:
print(den[i],roots)
```

References

[1] D. Bleichenbacher and A. May. New attacks on rsa with small secret crtexponents. *PKC 2006. LNCS*, 3958:1–13, 2006.

[2] J. Blomer and A. May. A generalized wiener attack on rsa. *PKC 2004, LNCS*, 2947:1–13, 2004.

[3] D. Boneh and G. Durfee. Cryptanalysis of rsa with private key d less than N^{0292}. *IEEE Trans. Inf. Theor.*, 46:1339–1349, 2000.

[4] D. Boneh, G. Durfee, and Y. Frankel. An attack on rsa given a small fraction of the private key bits. *ASIACRYPT'98, LNCS 1514.*, 1514:25–34, 1998.

[5] M. Bunder and J. Tonien. A new attack on the rsa cryptosystem based on continued fractions. *Malays. J. Math. Sci.*, 11:45–57, 2017.

[6] D. Coppersmith. Small solutions to polynomial equations, and low exponent rsa vulnerabilities. *J. Cryptology*, 10(233-266), 1997.

[7] G. De Micheli and N. Heninger. Survey: Recovering cryptographic keys from partial information, by example. *IACR Communications in Cryptology ISSN 3006-5496*, 1(1):48 pages.

[8] B. de Weger. Cryptanalysis of rsa with small prime difference. *AAECC*, 13:17–278, 2002.

[9] A. Dujella. Continued fractions and rsa with small secret exponent. *Tatra Mt. Math. Publ.*, 29:101–112, 2004.

[10] A. Dujella. A variant of wiener's attack on rsa. *Computing 85, 77–83 (2009)*, 85:77–83, 2009.

[11] M. Ernst, E. Jochemsz, A. May, and B. de Weger. Partial key exposure attacks on rsa up to full size exponents. *EUROCRYPT 2005. LNCS*, 3494:371–386, 2005.

[12] Y. Feng, A. Liu, Z. nad Nitaj, and Y. Pan. Practical small private exponent attacks against rsa. *IACR Cryptol. ePrint Arch. 1331*, 2024.

[13] M. Herrmann and A. May. Maximizing small root bounds by linearization and applications to small secret exponent rsa. *PKC 2010. LNCS*, 6056:53–69, 2010.

[14] A. Y. Khinchin. *Continued Fractions*. University of Chicago Press, 1964.

[15] P. Kocher. Timing attacks on implementations of diffie-hellman, rsa, dss, and other systems. *Proc. of Cyrpto 96,*, (104–113), 1996.

[16] Q. Li, Q. Zheng, and W. Qi. Practical attacks on small private exponent rsa: New records and new insights. *Des. Codes Cryptogr.*, 91(12):4107–4142, 2023.

[17] D. I. Nassr, H. M. Bahig, A. Bhery, and S. S. Daoud. A new rsa vulnerability using continued fractions. *2008 IEEE/ACS*, pages 694–701, 2008.

[18] L. Peng, L. Hu, Z. Huang, and J. Xu. Partial prime factor exposure attacks on rsa and its takagi's variant. *ISPEC 2015, LNCS*, 9065:96–108, 2015.

[19] S. Sarkar and S. Maitra. Improved partial key exposure attacks on rsa by guessing a few bits of one prime factor. *ICISC 2008, LNCS*, 5461:37–51, 2008.

[20] R. Steinfeld, S. Contini, H. Wang, and J. Pieprzyk. Converse results to the wiener attack on rsa. *PKC 2005. LNCS*, 3386:184–198, 2005.

[21] R. Steinfeld and Y. Zheng. An advantage of low-exponent rsa with modulus primes sharing least significant bits. *Topics in Cryptology — CT-RSA 2001., LNCS*, 2020, 2001.

[22] W. Susilo, J. Tonien, and G. Yang. The wiener attack on rsa revisited: A quest for the exact bound. *Information Security and Privacy: 24th Australasian Conference, Proceedings. 11547.*, pages 381–398, 2019.

[23] A. Takayasu and N. Kunihiro. How to generalize rsa cryptanalyses. *PKC 2016, LNCS*, 9615:67–97, 2016.

[24] A. Takayasu and N. Kunihiro. A tool kit for partial key exposure attacks on rsa. *Topics in Cryptology – CT-RSA 2017.*, 2017.

[25] A. E. Trishin. Wiener attack and weak keys of the rsa cryptosystem. *Discrete Mathematics and Applications.*, 35(5):331–338, 2025.

[26] E. Verheul and H van Tilborg. Cryptanalysis of 'less short' rsa secret exponents. *Appl. Algebra Eng. Commun. Comput.*, 8:425–435, 1997.

[27] M. Wiener. Cryptanalysis of short rsa secret exponents. *IEEE Trans. Inform. Theory*, 36:553–558, 1990.

[28] M. Zheng, Y. Feng, A. Nitaj, and Y. Pan. Improving rsa cryptanalysis: Combining continued fractions and coppersmith's techniques. *ACISP 2025. LNCS*, 15660, 2025.

[29] M. Zheng, A. Nitaj, M. Boudabra, M. Seck, O. Niang, and D. Sow. A generalized attack on rsa and its variants. *IACR Cryptol. ePrint Arch.*, 172, 2026.