FACTORIZATION AND MALLEABILITY OF RSA MODULI, AND COUNTING POINTS ON ELLIPTIC CURVES MODULO N

LUIS V. DIEULEFAIT AND JORGE JIMÉNEZ URROZ

Abstract. In this paper we address two different problems related with the factorization of an RSA modulus N. First we can show that factoring is equivalent in deterministic polynomial time to counting points on a pair of twisted Elliptic curves modulo N. The second problem is related with malleability. This notion was introduced in 2006 by Pailler and Villar, and deals with the question of wether or not the factorization of a given number N becomes substantially easier when knowing the factorization of another one N' relatively prime to N. Despite the efforts done until today, a complete answer to this question was unknown. Here we settle the problem affirmatively. To construct a particular N' that helps the factorization of N we use the number of points of a single elliptic curve modulo N. Coppersmith's algorithm allows us to go from the factors of N' to the factors of N in polynomial time.

1. Introduction

There is no need to explain the importance of secure digital communication nowadays. We are using computers for military purposes, politics, electronic payments, voting and even lately taking sharing decisions via blockchain. And the standard tool to provide data security is through Cryptography. Since 1977 it has been proved that asymmetric encryption and, in particular RSA which is the most widely used, is a very convenient mechanism both from security and efficiency point of view.

The security of RSA is based on the hardness of factoring large integers, and there is a massive literature on this subject. Nowadays, even though the most efficient factorization algorithm is the general number field sieve [1], [2], [3], which works in subexponential running time, the future seems to lead us to quantum computation, where the improvement is dramatic. In this setting we find Shor's algorithm which is able to factor integers in polynomial-time in a gate-based quantum computer, and there are other apparently fast algorithms in adiabatic or annealing quantum computers [4], [5], [6], [7].

In spite of the existence of these algorithms, since it is not yet possible to build quantum computers with sufficiently many qbits to factor large integers, the security of cryptosystems relying on the hardness of integer factorization such as RSA is not currently at stake. Therefore, both for practical purposes and for its theoretical

The second author is partially supported by the PID2019-110224RB-I00 grant of the MICINN (Spain).

intrinsic interest, the problem of integer factorization (with classical computers) is highly relevant.

But we have many algorithms to find the factors of a number, and we learn them at the school, so what does it mean that factoring is difficult? Well, it simply means that given

$$\begin{aligned} N = {} & 22112825529529666435281085255026230927612089502470015394413748319128822941402001986512729726569746599085900330031400051170742204560859\\ & 27635795375718595429883895870922923849100670303412462054578456641366454068421436129301769402084639106587591479425143514445819 9 \end{aligned}$$

and knowing that it is the product of two prime factors, nobody in the world knows how to find them using only N and no extra a priori information on his factors. It is so hard that in fact the problem seems completely different each time you try to factor a new number. In other words it seems, at first, that even if you know the factors of any number relatively prime to N, it will not help to find the factors of N. This is what is known in the literature as non-malleability of the factorization problem.

The motivation of this note is twofold. We started studying the problem of malleability of an RSA modulus N and suddenly we came to deal directly with the problem of factorization, finding that in fact it is equivalent to counting the number of points on an elliptic curve modulo N.

But let us start with malleability. It was introduced in 2006 in the paper by Pailler and Villar [8]. This notion, which we give explicitly in Section 3, captures a very basic fact in arithmetic: intuitively, one tends to believe that the problem of factoring a given number N (an RSA modulus) is not made easier if we know how to factor other numbers N' relatively prime to n. If this is true we say that factoring is non-malleable.

Appart from its purely arithmetic nature, the truth is that malleability appeared to the authors while studying the existence of a tradeoff between one-wayness and chosen ciphertext security, already observed back in the eighties for example in [9, 10, 11]. In some sense, one cannot achieve one-way encryption with a level of security equivalent to solve certain difficult problem, at the same time as the cryptosystem being IND-CCA secure with respect to it.

Even though this paradox has been observed, it has not been formally proved except in the case of factoring-based cryptosystems in which Pailler and Villar [8] clarified the question reformulating the paradox in terms of key preserving black-box reductions and proved that if factoring can be reduced in the standard model

to breaking one-wayness of the cryptosystem then it is impossible to achieve chosen-cyphertext security.

After this they introduce the notion of malleability of a key generator and, with it, they are able to extend the result from key preserving black box reductions to the case of arbitrary black box reductions.

Given the importance of both, one-wayness and CCA security, authors started to search for ways to overcome this uninstantiability and two relevant papers appeared. In [12] the authors propose a new public-key encryption scheme that is based on Rabin's trapdoor one-way permutation with equivalence between CCA security and the factoring problem. Being aware of the work of [8], they had to modify the setting to achieve their result. Then, in [13] the authors show that the widely deployed RSA-OAEP encryption scheme of Bellare and Rogaway [14], which combines RSA with two rounds of an underlying Feistel network whose hash (i.e., round) functions are modeled as random oracles, meets indistinguishability under chosen-plaintext attack (IND-CPA).

Both papers introduced modifications trying to avoid the paradox in the more general setting of arbitrary black-box reductions, proved via non-malleability. So, as the authors themselves stress in [8], it is very important to study non-malleability of key generators and, in fact, they conjecture that most instance generators are non-malleable, although no arguments are given to support this belief.

At this point let us give the more precise definition of malleability. It rests in measuring the difference between suitable Game 0 and Game 1, as defined in [8]. In Game 0 we factor a given number N with an oracle which can solve any problem that can be reduced to factoring. On the other hand, in Game 1 the oracle has the extra ability of factoring numbers which are relatively prime to N. If the probability of factoring N increases significantly in Game 1, then we say that factoring is malleable. (see [8], Section 4.1, for more details)

In [15] we address this question and notice that the freedom of selecting the new number N' breaks the independent behaviour of prime numbers, and hence we produce an explicit N' which makes factorization malleable. In other words, given any RSA modulus N we prove the existence of a polynomial time reduction algorithm from factoring N to factoring certain explicit numbers N', all relatively prime to N.

The numbers given in [15] are very simple: given the RSA modulus $n = pq$, the factorization of $N' = m^N - 1$, where m is a primitive root of the smallest prime dividing N, allows us to factor N in polynomial time. However, this does not give a complete satisfactory answer for several reasons. First one could think of N' to be of exponential size and then out of the scope of the question. However, as we

mention in [15], one can think of N' as a collection of exactly N ones when it is written in m-ary, and we just need the factors of N' modulo N, a data that has the same size as the given number. In any case, it still persists the restlessness of not knowing whether or not in a small interval centered in N we can find an explicit N' which can help to factor N.

In this paper we address precisely this question, and give an affirmative answer to the malleability of the problem of factoring by showing a number of the same size of N whose factorization allows us to factor N with a algorithm that runs in polynomial time.

To achieve this goal we will use very basic facts from the theory of elliptic curves. Concretely we will prove that given a random elliptic curve E defined modulo N, where N is an RSA modulus, and assuming that its number of points $|E(\mathbb{Z}/N\mathbb{Z})|$ is known, by further knowing the factorization of $|E(\mathbb{Z}/N\mathbb{Z})|$ we can produce a deterministic polynomial time algorithm that factors N. The key tool in our proof will be the result of Coppersmith (see [16]) that allows to factor an integer by knowing only certain bits of one of its prime factors.

This settles the question and proves that factoring is a malleable task. The first consequence of the result is obviously that the impossibility results gotten in [8] for key-preserving reductions, cannot be extended to arbitrary reductions, leaving open whether a cryptosystem could be constructed such as its one-way security is equivalent to factoring, and CCA at the same time. In particular, for example it is known that onewayness of Rabin encription is equivalent to factoring, and it remains unknown the existence of an instantiation in the standard model chosen-cyphertext secure under the factoring assumption.

While proving the previous statement on malleability another interesting problem treated widely in the literature, (see [17] and [18] for related results) showed up in a natural way:

Problem Is factoring N equivalent to counting the number of points of elliptic curves modulo N.?

In this paper we give a definite answer to this question by proving the following theorem:

Theorem 1. *Given N and the number of affine or projective points, $M \neq N$, of any elliptic curve E and of one of its twists E_d modulo N, with $(d, N) = 1$, we can factor N in deterministic polynomial time.*

The proof of this result relies in proving a rather elementary new lemma, Lemma 3, that even though it is remarkably simple it was not in the literature so far.

Remark 2. *As we have already remarked, the previous problem has been addressed in* [18]*. We should stress that the results in that paper are based in an assumption on the distribution of the number of points on elliptic curves over finite fields which is not accurate. Also, the reduction algorithm from counting the number of points of the elliptic curve modulo N to factoring N in their case is probabilistic while here it is proved to be deterministic. Moreover, in terms of malleability, what we do in Section 3 involves taking a single elliptic curve, and succeeds with probability* 1*, while the results in* [18] *require considering many elliptic curves to have positive probability to factor N. Finally the method used in that paper only works for the number of projective points on the elliptic curve, not covering the affine case as we do.*

The structure of the paper goes as follows: In Section 2 we prove Theorem 1, while Section 3 is dedicated to the problem of malleability of factoring.

2. Factorization

Let $N \in \mathbb{Z}$. Given an elliptic curve $E := \{y^2 = x^3 + ax + b\}$ over $\mathbb{Z}/N\mathbb{Z}$, we will denote by E_d its quadratic twist $E_d := \{dy^2 = x^3 + ax + b\}$. $E(\mathbb{Z}/N\mathbb{Z})$ will be the group of $\mathbb{Z}/N\mathbb{Z}$ points, and $E'(\mathbb{Z}/N\mathbb{Z})$ the set of affine points of the curve. In any event, if C is a set we will let $|C|$ to be its cardinal.

In the case when $N = l$ is a prime number, then $|E(\mathbb{Z}/l\mathbb{Z})| = l + 1 - a_l$, where a_l is the trace of the Frobenius endomorphism of the curve E modulo l, $|a_l \leq 2\sqrt{l}|$ and the curve has only one point at infinity. We will denote $I_l = \{l+1-2\sqrt{l}, l+1+2\sqrt{l}\}$ the Hasse interval. Also, it is well known that if d is an integer coprime to l then $|E_d(\mathbb{Z}/l\mathbb{Z})| = l + 1 - \left(\frac{d}{l}\right) a_l$.

In the case when $N = pq$, a product of two prime numbers, then we know that $E(\mathbb{Z}/N\mathbb{Z}) = E(\mathbb{Z}/p\mathbb{Z}) \times E(\mathbb{Z}/q\mathbb{Z})$ and hence

$$|E(\mathbb{Z}/N\mathbb{Z})| = (P - a_p)(Q - a_q) = PQ - Pa_q - Qa_p + a_p a_q, \tag{1}$$

where $P = p + 1, Q = q + 1$ in the projective case and $P = p, Q = q$ in the affine case.

Now consider and RSA modulus N and $(d, N) = 1$. There are three options for $E_d(\mathbb{Z}/N\mathbb{Z})$ (or $E'_d(\mathbb{Z}/N\mathbb{Z})$), depending on the Legendre simbols $\left(\frac{d}{p}\right)$ and $\left(\frac{d}{q}\right)$. Let us denote, by abuse of notation, $E = |E(\mathbb{Z}/N\mathbb{Z})|$, and $\hat{E}, \tilde{E}, \bar{E}$ to the following

integers

$$\hat{E} = (P + a_p)(Q + a_q) = PQ + Pa_q + Qa_p + a_p a_q,$$
$$\tilde{E} = (P - a_p)(Q + a_q) = PQ + Pa_q - Qa_p - a_p a_q,$$
$$\bar{E} = (P + a_p)(Q - a_q) = PQ - Pa_q + Qa_p - a_p a_q.$$

Then,

$$E + \hat{E} + \tilde{E} + \bar{E} = 4PQ, \tag{2}$$

while

$$4PQ = \frac{(E + \tilde{E})(E + \bar{E})}{E} = E + \tilde{E} + \bar{E} + \frac{(\tilde{E}\bar{E})}{E} = 4PQ - \hat{E} + \frac{(\tilde{E}\bar{E})}{E},$$

so

$$E\hat{E} = \tilde{E}\bar{E}. \tag{3}$$

Lemma 3. *Knowing two among $\tilde{E}, \hat{E}, \bar{E}$ and E, we know the four of them.*

Proof. We split the proof in 2 cases.

Case 1. We suppose E and $\hat{E}$ are known. The case in which $\tilde{E}$ and $\bar{E}$ are known is analogous. Then we compute its product, $M = E\hat{E}$ and its sum $L = E + \hat{E}$, and we have

$$\tilde{E}\bar{E} = M$$
$$\tilde{E} + \bar{E} = 4PQ - L$$

so $\tilde{E}$ and $\bar{E}$ are the solutions of the quadratic polynomial $X^2 - (4PQ - L)X + M$.

$$\tilde{E} = \frac{4PQ - L + \sqrt{(4PQ - L)^2 - 4M}}{2}, \qquad \bar{E} = \frac{4PQ - L - \sqrt{(4PQ - L)^2 - 4M}}{2}$$

Case 2. Suppose E and $\tilde{E}$ are known. The cases in which the pairs $(E, \bar{E})$, $(\hat{E}, \tilde{E})$ and $(\hat{E}, \bar{E})$ are known, are analogous. Then, compute the quotient $\frac{E}{\tilde{E}} = M$ and the sum $E + \tilde{E} = L$. Hence, $\frac{\bar{E}}{\hat{E}} = M$, or $\bar{E} = M\hat{E}$, and by (2) $(M+1)\hat{E} = 4PQ - L$, or

$$\hat{E} = \frac{4PQ - L}{(M+1)}, \qquad \bar{E} = \frac{M(4PQ - L)}{(M+1)}.$$

Theorem 4. *Knowing either $E(\mathbb{Z}/N\mathbb{Z})$ and $E_d(\mathbb{Z}/N\mathbb{Z})$ or $E'(\mathbb{Z}/N\mathbb{Z})$ and $E'_d(\mathbb{Z}/N\mathbb{Z})$ for $\left(\frac{d}{p}\right) = -1$, we can factor N in polynomial time.*

In the projective case, we compute the four integers $E, \hat{E}, \tilde{E}, \bar{E}$ by Lemma 3 and then its sum to compute PQ. With PQ and N we factor N.

In the affine case, we again compute the four integers $E, \hat{E}, \tilde{E}, \bar{E}$ and then note that $E + \tilde{E} = 2q(p - a_p)$, has q as a common factor with N, so if $a_p \neq 0$, computing the gcd with N, we factor N. On the other hand, if $a_q \neq 0$, we do the same with $E + \bar{E} = 2p(q - a_q)$.

In both cases observe that, in principle, we do not know which one is $E_d(\mathbb{Z}/N\mathbb{Z})$, so we will have to make two computations.

Remark. The theorem obviously does not apply in the affine case if $a_p = a_q = 0$, since then the number of affine points of the elliptic curve and its twists is simply N, so we do not get new information. In this case, we just have to select another curve.

Theorem 5. *Under ERH factoring an RSA modulus $N = pq$ is polynomial time equivalent to counting the number of points, affine or projective, of any elliptic curve E modulo N, non-supersingular for both primes p and q.*

Proof. Let E be an elliptic curve. Then, knowing the factorization of N, we can compute $|E(\mathbb{Z}/N\mathbb{Z})|$ by Schoof's algorithm [19].

Now suppose we know $|E(\mathbb{Z}/N\mathbb{Z})|$. Then, from [20], we know that under ERH the smallest quadratic nonresidue modulo p, call it d, is of size $O((\log p)^2)$. Hence, apply the previous Theorem 4 to the pair E, E_d for every d up to this bound.

Recall that, as of today, we can compute the number of points modulo N by baby step giant step, since $E \pmod N$ has group structure, in $O(N^{1/4+\varepsilon})$ which is exponential.

3. Malleability

As in previous sections, let $N = pq$ be an RSA modulus. We recall that in order to prove that factoring is malleable we need to find a number relatively prime to N and of the same size, which factorization will allow us to factor N in deterministic polynomial time. We consider a random elliptic curve $E \pmod N$, and we let $|E'(\mathbb{Z}/N\mathbb{Z})|$ be the number of affine points, while $|E(\mathbb{Z}/N\mathbb{Z})|$ will be the number of points including the points at infinity. We can assume that we have at our disposal an Oracle that computes any of these two numbers. Since this computation can be reduced to the factorization of N thanks to Schoof's algorithm, this corresponds to Game 0 in the setup described in the introduction while defining malleability. Note also that, as we have already observed, there is no known polynomial time algorithm that can factor N using this information.

Assume now that we have access to an auxiliary Oracle that can factor any number relatively prime to N. Using it, we factor the number $|E(\mathbb{Z}/N\mathbb{Z})|$ (or $|E'(\mathbb{Z}/N\mathbb{Z})|$) and we will show in the following theorem that from this we can factor N in polynomial time, thus concluding that Game 1 has solved the factorization problem that was not achieved by Game 0, which shows that factorization of RSA modulus is malleable.

Theorem 6. *Given $N = pq$ where p, q are prime numbers, and an elliptic curve $E \pmod N$, uniformly at random, there exists a polynomial time algorithm in $\log N$ such that with input N and the factorization of $|E(\mathbb{Z}/N\mathbb{Z})|$ or $|E'(\mathbb{Z}/N\mathbb{Z})|$, it outputs the factors of N, p and q with probability one.*

Proof. As we mentioned in the introduction, we will use a well known result of Coppersmith, which allows us to find a factor of an integer by just knowing certain part of its highest bits. For convenience we include this result now

Theorem 7. *(Coppersmith) If we know an integer $N = pq$ and we know the high order $(1/4)(\log_2 N)$ bits of p, then in polynomial time in $\log N$ we can discover p and q.*

Observe that it would be sufficient by knowing the $(1/4)(\log_2 N) - O(\log\log N)$ highest order bits, since we could try the rest up to $(1/4)(\log_2 N)$ one by one in polynomial time.

Now, recall again that, by Hasse's theorem the factor found $q-a_q+1$ is at distance $|a_q-1| \leq 2\sqrt{q}+1 \leq 2N^{1/4}+1$ of q which is a factor of N. By bounding the distance, we know certain of the highest bits of q from those of $q - a_q + 1$.

In particular, let us suppose that two integers $x < y$ are at distance $y-x = 2^t+R$ where $R < 2^t$. We can write $x = M_x 2^t + R_x$, $y = M_y 2^t + R_y$ with $R_x < 2^t$, $R_y < 2^t$ and $-2^t < R_y - R_x < 2^t$. Then, $y - x = (M_y - M_x)2^t + R_y - R_x$, which gives $M_y = M_x + 1$ or $M_y = M_x + 2$ and, hence, from the highest bits of y up to t of x we know those of y and viceversa.

In our case, the distance is bounded by $2q^{1/2}$ so we know up to $t = [\log_2 q/2] + 1$ of the highest bits of q. By division, we also know up to t of the highest bits of p. But $\sqrt{N} \leq p = M_p 2^t + R_p \leq (M_p + 1)2^t$, and so $M_p \geq \sqrt{N}/(2^t + 1) \geq N^{1/4}$ and, hence, we can apply Coppersmith algorithm to find p, thus factoring N.

The previous algorithm would not work if the number of factors of $|E(\mathbb{Z}/N\mathbb{Z})|$ would not be logarithmic on the size of N. However, this event is negligible. Indeed, Let C_N be the set of elliptic curves modulo N, where two curves will be the same if their reduction modulo p and q are isomorphic. For any set $C \subset C_N$ we will consider the probability $p(C) = \frac{|C|}{|C_N|}$. Recall that $\frac{|C_N|}{4N} = o(1)$, as can be seen in section (1.4) of [21] by using the Chinese Remainder Theorem.

Now consider for each divisor l of N the set $S_l = \{E \in C_N : d(|E(\mathbb{Z}/l\mathbb{Z}|) > (\log l)^4\}$, where, as usual, $d(n)$ denotes the number of divisor of the integer n. Note that if $N = pq$, then by (1) if $E \in S_N$ then either $E \in S_p$ or $E \in S_q$ and so $p(S_N) \leq p(S_p) + p(S_q)$. Now, using Proposition 1.9 in [21] we see that for $l = p$ or $l = q$

$$p(S_l) \leq c|S_l|\frac{(\log l)^2}{\sqrt{l}}.$$

On the other hand from the average of the divisor function

$$\sum_{n\leq x} d(n) = x\log x + (2\gamma - 1)x + o(x),$$

we deduce that,

$$|S_p|(\log p)^4 < \sum_{m\in S_p\subset I_p} d(m) \leq \sum_{m\in I_p} d(m) = 4\sqrt{p}\log p + O(\sqrt{p}) \ll,$$

and so

$$|S_p| \ll \frac{\sqrt{p}}{(\log p)^3},$$

which in particular gives

$$p(S_N) = o\left(\frac{1}{\log N}\right)$$

which tends to zero as N goes to infinity. So with probability basically 1, the number of factors of $|E(\mathbb{Z}/N\mathbb{Z})|$ is of order a power of the logarithm of N, and in particular polynomial in N and, hence, once the auxiliary Oracle gives the factors of $|E(\mathbb{Z}/N\mathbb{Z})|$, we will apply Coppersmith one by one finding q in polynomial time in $\log N$.

Remark 8. *The case in which $|E'(\mathbb{Z}/N\mathbb{Z})|$ is given is similar and we leave the details to the reader.*

3.1. **Small difference.** Even though malleability is fully proved in the previous section, we include this section as a small remark in the negligible case in which $|E(\mathbb{Z}/N\mathbb{Z})|$ and $|E'(\mathbb{Z}/N\mathbb{Z})|$ have an exponential number of divisors, but the two prime factors p, q are not too far from each other.

In order to construct an RSA modulus we tipicaly search for a couple of prime factors of the same number of bits, i.e. $q < p < 2q$. However, if the two primes are very close to each other, the scheme is easy to break since the modulus can be factored in polynomial time. Indeed, It is well known that if $\Delta = |p - q| < N^{1/4}$ Fermat's factorization algorithm enables to find both factors of N in polynomial time and there has been an effort of the community to improve the exponent $1/4$ in Δ for the factorization of N. It is worth to mention that if the objective is breaking the RSA scheme, rather than factoring the modulus, then the exponent can be increased all the way up to basically 1 by means of an improved version of the attacks done by Wiener or Boneh and Durfee, (see [22]). However, for the factorization of N not too much more is known. In [23] the authors claim in an apparently unpublished work that are able to factor an RSA modulus $N = pq$ even when the difference is of order $|p - q| < N^{1/3}$.

We devote this section to recover $\Delta < N^{1/3}$ using malleability techniques: in particular the factorization of the number of points of a random elliptic curve modulo N, together with a simple application of an argument of elementary geometry attributed to Heron of Alexandria which says that in any triangle, the product of the length of its three sides equals four times the area times the radius of the circumscribed circle. We will assume from now that $\Delta = |p - q| < c'N^{1/3}$ for some suitable constant c'.

In our case given three points $(x_0, y_0), (x_1, y_1), (x_2, y_2)$ of integer coordinates in the hyperbola $xy = |E'(\mathbb{Z}/N\mathbb{Z})|$, we see that the radius of the circumscribed circle is

$$R = \frac{((x_0x_1)^2 + (|E'(\mathbb{Z}/N\mathbb{Z})|)^2)((x_0x_2)^2 + (|E'(\mathbb{Z}/N\mathbb{Z})|)^2)((x_2x_1)^2 + (|E'(\mathbb{Z}/N\mathbb{Z})|)^2)}{4(|E'(\mathbb{Z}/N\mathbb{Z})|)^2(x_0x_1x_2)^2},$$

and taking $\max\{x_0, x_1, x_2\} \leq \sqrt{|E'(\mathbb{Z}/N\mathbb{Z})|}$, we get

$$R \geq \frac{|E'(\mathbb{Z}/N\mathbb{Z})|}{4}.$$

On the other hand, by Hasse's theorem

$$|E'(\mathbb{Z}/N\mathbb{Z})| \geq (\sqrt{p} - 1)^2(\sqrt{q} - 1)^2$$

and

$$(\sqrt{p} - 1)(\sqrt{q} - 1) = \sqrt{N} - \sqrt{p} - \sqrt{q} + 1 \geq \sqrt{N}/4,$$

for N sufficiently large, and so

$$R \geq \frac{N}{64}.$$

Hence, by Heron of Alexandria's theorem, in an arc of the hyperbola $xy = |E'(\mathbb{Z}/N\mathbb{Z})|$ of lenght less than $(N/32)^{1/3}$ we can only have two points of integer coordinates.

Now recall that the lenght L of an arc of the hyperbola $xy = T$ with $a \leq x \leq b$ is given by

$$\begin{aligned} L &= \int_a^b \sqrt{1 + \frac{T^2}{t^4}}dt \leq \int_a^b 1 + \frac{T^2}{t^4}dt = (b - a) + \frac{T^2}{3}\left(\frac{1}{a^3} - \frac{1}{b^3}\right) = \\ &= (b - a) + \frac{(b - a)T^2}{3}\left(\frac{b^2 + ab + a^2}{(ab)^3}\right) \leq (b - a) + \frac{(b - a)T^2}{a^3b}. \end{aligned}$$

Consider $T = |E'(\mathbb{Z}/N\mathbb{Z})|$, $b \geq N^{1/2}$ and $b - a \leq cN^{1/3}$ for suitable c. Hence, by noting that

$$|E'(\mathbb{Z}/N\mathbb{Z})| \leq (\sqrt{p} + 1)^2(\sqrt{q} + 1)^2 = (\sqrt{N} + \sqrt{p} + \sqrt{q} + 1) \leq N + 7N^{3/4},$$

since the primes are very close, we get from a simple computation that the arc on the hyperbola has lenght

$$L \leq 3cN^{1/3},$$

In particular, we can select c small enough so $L \leq (N/32)^{1/3}$ and hence it can only have at most two points of integral coordinates. Hence we can ask the auxiliary Oracle to factor E_N^* and output the at most two factors of it lying in the interval $[a, b]$. This Oracle will give back the factor $a \leq N^{1/2} - c'N^{1/3} \leq q - a_q + 1 \leq N^{1/2} \leq b$, for $a_q \geq 0$ or $a \leq N^{1/2} \leq q - a_q + 1 \leq N^{1/2} + c'N^{1/3} \leq b$ if $a_q \leq 0$ for some $c' \leq c$. In practice $c = \frac{1}{3(32)^{1/3}}$ and $c' = \frac{1}{6(32)^{1/3}}$ are enough. Using it we can factor N with Coppersmith's algorithm, as we did in the previous subsection.

Remark 9. *Again the case in which $|E(\mathbb{Z}/N\mathbb{Z})|$ is given is similar and we leave the details to the reader.*

References

[1] Buhler JP, Lenstra HW Jr, Pomerance C, Factoring integers with the number field sieve. In: The Development of the Number Field Sieve LNM 1554, Springer-Verlag 1993.

[2] Kleinjung, T., On polynomial selection for the general number field sieve, Math. Comp.,75.256, 2037-2047, 2006.

[3] Aoki K, Franke J, Kleinjung T, Lenstra AK, Osvik DA, A kilobit special number field sieve factorization, ASIACRYPT, LNCS 4833, 1-12, 2008.

[4] Shor, P.W., Algorithms for quantum computation: discrete logarithms and factoring, Proceedings 35th Annual Symposium on Foundations of Computer Science. IEEE Comput. Soc. Press: 124-134, 1994.

[5] Beckman, D.; Chari, A.N.; Devabhaktuni, S.; Preskill, J. Efficient Networks for Quantum Factoring. Physical Review A. 54.2, 1034-1063, 1996.

[6] Xinhua P., Zeyang L., Nanyang X., Gan Q., Xianyi Z.,. Dieter S., and Jiangfeng D.,Quantum Adiabatic Algorithm for Factorization and Its Experimental Implementation, Phys. Rev. Lett. 101, 220-405 2008.

[7] Shuxian J., Keith A. B., Alexander J. M., Travis S. H. and Sabre K., Quantum Annealing for Prime Factorization, Scientific Reports volume 8, Article number: 17667 2018.

[8] Paillier, P. and Villar, J. L., Trading one-wayness against chosen-ciphertext security in factoring-based encryption, Lecture Notes in Comput. Sci., 4284, (ASIACRYPT 2006), 252-266, 2006.

[9] Rabin, M. O., Digital signatures and public key functions as intractable as factorization, Technical Report MIT/LCS/TR-212, 1979.

[10] Williams, H. C., A modification of the RSA public-key encryption procedure, IEEE Trans. Inform. Theory, 26.6, 726-729, 1980.

[11] Goldwasser, S. and Micali, S. and Rivest, R. L., A digital signature scheme secure against adaptive chosen-message attacks, SIAM J. Comput., 17.2, 281-308, 1988.

[12] Hofheinz, D.and Kiltz, E., Practical Chosen Ciphertext Secure Encryption from Factoring, EUROCRYPT 2009, 312-332, 2009.

[13] Kiltz, E.; O'Neill, A. and Smith, A., Instantiability of RSA-OAEP under chosen-plaintext attack, J. Cryptology, 30.3, 889-919, 2017.

[14] Bellare, M. and Rogaway, P., Optimal asymmetric encryption, EUROCRYPT, LNCS, 950, 92-111, 1995.

[15] Dieulefait, L. and Jiménez Urroz, J., Small primitive roots and malleability of RSA moduli, J. Comb. Number Theory, 2.2, 171-179, 2010.

[16] Coppersmith, D., Finding a Small Root of a Bivariate Integer Equation; Factoring with High Bits Known, Advances in cryptology, EUROCRYPT '96, Lecture Notes in Comput. Sci., 1070, 178-189, 1996.

[17] Martín, S. and Morillo, P. and Villar, J. L., Computing the order of points on an elliptic curve modulo N is as difficult as factoring N, Appl. Math. Lett. 14.3, 341-346, 2001.
[18] Kunihiro, N. and Koyama, K., Equivalence of counting the number of points on elliptic curve over the ring $\mathbb{Z}_n$ and factoring n, Lecture Notes in Comput. Sci., 1043, 47-58, 1998
[19] Schoof, R., Elliptic Curves Over Finite Fields and the Computation of Square Roots modulo p, Mathematics of Computation, 44.170, 483-494, 1985.
[20] Ankeny, N. C., The least quadratic non residue, Ann. of Math. (2), 55, 65-72, 1952.
[21] Lenstra, Jr., H. W.,Factoring integers with elliptic curves, Ann. of Math. (2), 126.3, 649-673, 1987.
[22] De Weger, B., Cryptanalysis of RSA with small prime difference, Appl. Algebra Engrg. Comm. Comput. 13.1, 17-28,2002.
[23] Erra, R and Grenier, C, The Fermat factorization method revisited, https://eprint.iacr.org/2009/318.pdf, 2009.

Dept. mat. i informàtica, Universitat de Barcelona, Gran Via de les Corts Catalanes 585;, 08007 - Barcelona; Spain.
Email address: ldieulefait@ub.edu

Departamento de Matemáticas, Universitat Politécnica Catalunya, Edificio C3 - Campus Nord UPC, Carrer de Jordi Girona 1-3, 08034 Barcelona, Spain
Email address: jorge.urroz@upc.edu