FBA

Abstract

One of the most difficult challenges in number theory and cryptography is the problem of factoring large integers since factoring large numbers more than 200 digits remain difficult. This difficulty has been useful in proving the strength of the RSA encryption scheme, which uses two large numbers of similar size. In this work, we introduce factorisation by bases algorithms (FBA) and implement them on the RSA-numbers defined by the RSA challenge. We analyse the time spent on factoring to check the speed and efficiency of each algorithm.

Introduction

Integer factorization is the task of computing the divisors of natural numbers. It is a problem with a long and fascinating history, and it is certainly among the most influential in algorithmic number theory. While there is a variety of algorithms significantly faster than the brute-force search for divisors, it is still an open problem to construct a technique that efficiently factors general numbers with hundreds to thousands of digits. The hardness of this problem is fundamental for the security of widely used cryptography schemes, most prominently the RSA cryptosystem. Nevertheless, there is no proof for its hardness besides the fact that decades of efforts have failed to construct a more efficient technique. Quite regularly, there are set new records1 concerning the factorization of numbers of certain size, mostly due to improved implementations of the best available algorithms and advances in the hardware and computing power. In addition, the bound for the deterministic integer factorization problem has been improved multiple times in recent years ([11], [12], [10], [14]). On the other hand, there has only been little progress in the development of new techniques for practical integer factorization since the invention of the Number Field Sieve ([19]) in the 1990s. One of the earlier algorithms with sub-exponential runtime was by Dixon ([7]) in 1981.

Background

Let N be the number we want to factorize. We will always assume that N is odd, composite and not a perfect power of another number. We will also assume that N is made up of 2 prime factors, p and q. We will explain properties of factorisation by bases techniques using one number in base 10.

Lemma: N can be converted into a polynomial given a base.

Example: 667 in base 10 is

Now 667 is built by two prime factors which are 23 and 29. If both of them are converted into base 10 they become, and respectively.

which is different from the first form of N gotten. (A1)

From this derivation we notice that cannot be factorised to give us p and q because the discriminant was imaginary. Furthermore, we also notice that for any converted form of N we get we cannot factorise if also the discriminant is imaginary. From (A1) we noticed that a change of the form happened with h=2 and k=2 (A2) which results in, . This becomes the obtained equation.

If we say

We obtain the discriminant: which reduces to

(A3)

To solve such a problem is quite complex. We want to reduce the equation to 2 variables which can be easily solved either as Diophantine equations or through some ingenious method. There are many ways to do this.

CASE I

Given Here if we insert a value for m for example, 12 we get: which is a conic section. We will refer to this paper written by on solving conic sections.

Note: Since this one is a conic section and to find the values of we will need to factorise N using SIQS or ECM, hence this one will not have an algorithm.

CASE II

Given Let if we say M=34, we get We will now only be left with finding m of which by Diophantine solutions, we know that m is of a specific parametric solution.

Let

Note: This one will have an algorithm. Name it ALGORITHM 1.

Not necessary since predicting correct M is difficult

CASE III

If we are to go back to the original form with only variables, we notice that and . For the discriminant we get,

Making unknowns subjects of the knowns, for example, making subject of B, we get.

And for in base 10, we obtain.

Well, this means that we now have 2 equations of which we can use to find solutions. We now have four variables of which we need a third equation to reduce all the variables to one variable which can easily be solved with.

If we have,

(T1)

(T2)

And then substitute (T1) into (T2) to eliminate m, we have,

(T4)

Which means, if we put a value for we find and conversely.

But since m is the holy grail of this solution, eliminating it makes it harder for us to calculate it, rather let’s eliminate a value such as k, and that must give us a desirable solution.

Given we have (T1) and (T2), the third equation is establishing a relationship between o and m, such of their squares, for example: . If we insert any value for R and then iterate for any value of k, we get deterministic results for all the variables. Examples are shown below.

Let (T1), (T2) and

(explain how we got )

For

For

This is because if we look closely at the situation, we have

(T1)

(T2)

If we put a value for k, for example, k=2, we get:

Hence if we put any value for R we get h and m values which will give us integer solutions. However, most of these solutions will be N of which we want to find p or q, making it difficult.

To conquer this difficulty we substitute the gotten values of and m, into and (T1), meaning we have to just solve:

(T1)

And we get:

and we know that one of the values of R is

The challenge we face here is that of finding the gcd of numerator and the denominator since we should first find x in its simplest form so that we can later equate the product of p generated from x to N. but we know that the gcd must be of the factors of 16, however, even if we predict gcd to be 8, we find ourselves in an interesting situation when trying to solve for the other x, since this will imply that both the numerator and the denominator must be divisible by the gcd.

Because of the failed attempt, we can still solve it using another way. In this case we use, (T2). Because are always strict, we assume that x is varying to find o.

If we have that , we get this sample of data between x and o.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| x | 0 | 10 | 24 | 42 | 64 | 90 |
| o | 32 | 48 | 64 | 80 | 96 | 112 |

And if we have , we also get.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| x | 0 | 2 | 10 | 16 | 32 | 42 |
| 0 | 16 | 32 | 64 | 80 | 112 | 128 |

This means that we can use the values of x to obtain the required solutions for . to test our hypotheses, we will use the first scenario in which m was 12.

(T8)

(T9)

(T9) – (T8) gives:

(T10)

Meaning that we just need to guess and then we can solve for h,k. of which we know that must be a factor of 80. Testing for it gave us these solutions:

Note: ALGORITHM 2, a breaker, which might give us deterministic solutions. Contains the ability to prove that P=NP.

CASE IV

Let’s have then it shows that for the discriminant we have

Here we can insert any value for b and m and get deterministic results.

For example, for N = 667, x = 10, we have:

If and we get: . The solution set for 6 values is shown below.

Such solutions are a closed set in which x and the complement of x are all solutions.

If we substitute x with 10 for all the solutions, we will find none equating to 667 and its multiples. However, if we try to find solutions for x such that we get the prime solutions, for example.

For

Using , for the solutions of x to find p and q, we get:

But finding solutions such that takes time especially when resolving large numbers, since current methods require us to factorise N first.

The solution to this problem we have two options.

OPTION 1

We assume that the reason why some of the values did not give us deterministic results was due to the fact that both andmust have been fractions, hence by Diophantine solving were skipped. If we take that both and are of the form and since we want to get complete integer solutions we get the general solution:

In which we continuously iterate theta until we find

Note: ALGORITHM 3

OPTION 2

Here we put C to be C-N for f(x).

Using N=667, x=10, we get.

Let and , we get:

As for the solution sets, we have:

And if we take all the solutions of we have:

We had no or its multiples.

Assuming S is a group and using the preliminary laws of algebra, a group is said to be associative, commutative and contains and identity element as well as an inverse. In our case, let the identity be the prime factor we want to find. Then we have whereby \* is addition.

Examples:

Therefore ALGORITHM 4

If for was found, then we can get the same solutions for .

Let’s take the situation in (A9) whereby.

If and we get: .

Then for where \* is addition we have:

Note: ALGORITHM 5

FURTHER CLUES

Even if we insert any pair of to find , it’s still not enough to give us some desired solutions. Some solutions will be omitted from the mix. So to find the whole solutions, firstly we insert the pair such that they are small enough. After this we obtain the pairs of Then using any pair of , we substitute into the original equation so that we find the pairs of We can then combine them to find the pairs of

Example:

Let

Using

CASE V

Here we try to solve in a different manner from the rest. Previously, the solutions were based on the derivative of a function but here we solve using integration instead. Rather than using roots to solve a function, we propose that we can find similar solutions even if we use the area under a cave, as long as it is bound within some roots.

Let’s take for example, four situations in which gave us prime solutions.

(V1)

(V2)

(V3)

(V4)

From previous examples, we noticed that for most functions in which we got p,q = 1 or N, if we added or subtracted such functions from that in which we get p,q not equal to N or 1, we would derive deterministic prime solutions. Let’s now apply this for the integration, ignoring the constant. Let be the area between one of the deterministic roots against that when x is the base value.

Using V1

Meaning the area can be between or

For the area bound between we get:

For the area bound between we get:

We notice that and

And using V3

Meaning the area can be between or

For the area bound between we get:

For the area bound between we get:

We notice that and

And using V4

Meaning the area can be between or

For the area bound between we get:

For the area bound between we get:

We notice that and

Using the above scenarios, we notice that the area bound between x=base and x=root will give a prime solution which is factor of N.

In this situation we try to solve using the simplest possible solutions. One way is to input the value of then iterate. In one situation we found for V1, hence we can assume that such a value must exist for some .

Let

and if equal to zero we obtain.

and to remove the fractions, we get.

then as for the discriminant:

of which m was found to be zero.

This implies that: of which solving it like this is more difficult as it resorts us to the first case scenario (CASE I). Hence, we combine this solution with the discriminant solution of to reach a consensus.

(T16)

(T17)

results in:

where

when

when

when

when

Using the above cases, we notice that the value of h remained constant regardless of the value of m. And when letting , we got which gave us a number of parametric solutions sets. One of them was .

In the next scenario we notice an anomaly in the sense that if we get some value of k, m and h then substitute them in we get an which does not satisfy some of the requirements we stated should happen for a full factorization, but even so, if we continue with the formulas stated we return with the correct value for p.

Example:

Given .

When

It results in such that

And such that

Also if we use we get and expanding the number, we find out that the which confirms our proposition.