

## Re: JSH: Nearly done

**Source:** <http://sci.tech-archive.net/Archive/sci.math/2005-01/7886.html>

---

*jstevh\_at\_msn.com*

**Date:** 01/27/05

Date: 26 Jan 2005 16:13:05 -0800

Mark Nudelman wrote:

> *jstevh@msn.com* wrote:

>> *If any of you have an ounce of mathematical ability then meet me with*

>> *mathematics.*

>>

>> *I dare you.*

>

> *I wish you \_would\_ talk about the mathematics more, rather than all the long*

> *postings about how people are mistreating you.*

>

> *Specifically I would like you to answer two questions:*

>

> *1. Why is it easier to factor T rather than M? In another post I think you*

> *said it's \_not\_ because it may have small factors, as Nora proposed.*

No.

$$T = M^2 - j^2 = (M-j)(M+j)$$

and j can be chosen such that M+j has any given prime you wish as a factor.

So you can guarantee that the numbers you need to factor are smaller than M.

My prototype program already does that to a certain extent, so you could put in some huge number into it, but it'd probably take it a few days to process through and more than likely, it wouldn't factor.

But it would process through in a few days, even with an RSA challenge number.

That's how fast it is.

- > 2. *What's the basis for your statement that the whole process will execute*
- > *in polynomial time?*
- >

I'm focusing on building a full method that relies only on my work, so it has to call itself recursively to factor, and such a method can potentially chew through even an RSA challenge type number in minutes.

Then it's just a matter of iterating through the various combinations, which I've figured out are at about 150 million for the number generated by the first 1000 primes.

Potentially this method can factor an RSA challenge sized number in seconds.

I call that polynomial time.

Now, when I talk about development, I'm talking about using the method fully, so that it acts recursively.

NOW you can use some other method to factor T, like elliptic curve, and then use that factorization, as if you can get rational x's, then you will have about a 50% probability per.

I call that an inelegant solution and rely on my own algorithm for my research.

However, if rational x's can be consistently found, then the mathematics says that the method will more than likely factor a number without regard to size as long as you have T factored, as only a few rational x's are needed with the 50% success rate per.

Now, all the math at the lower level has been worked out, while I haven't yet proven conclusively that you can get rational x's at will, and even for very large numbers.

Maybe for really big numbers rational x's become difficult to find.

But at this point, I don't see a mathematical reason why they should.

I think it would be quite useful if one of you could settle the question.

James Harris