

Re: Integer factorization, basics

Source: <http://sci.tech-archive.net/Archive/sci.math/2005-01/0517.html>

mensanator_at_aol.compost

Date: 01/01/05

Date: 1 Jan 2005 09:41:07 -0800

jstevh@msn.com wrote:

- > *Just in case you don't know a lot about integer factorizations, I*
- > *thought I'd give a quick informative post to go over the basics, and*
- > *why my current approach that I call surrogate factoring raises some*
- > *serious concern.*
- >
- > *Basically if you have some non-zero integer M , you can consider*
- >
- > $ab = M$
- >
- > *where let's say you want 'a' and 'b' to be integers as well.*
- >
- > *There is always a finite set of possible values for 'a' and 'b', for*
- > *any non-zero integer M .*
- >
- > *The factoring problem concerns itself with finding non-trivial values*
- > *for 'a' and 'b', where trivial values are ones like $a=M$, and $b=1$, or*
- > *$a=-1$, and $b=-M$.*
- >
- > *Now let $a=b+c$, and you get*
- >
- > $(b+c)b = M$, so $b^2 + bc = M$, so
- >
- > $b^2 + bc - M = 0$
- >
- > *and you can solve to get*
- >
- > $b = (-c \pm \sqrt{c^2 + 4M})/2$
- >
- > *where you have the congruence of squares method, and if you find some*
- > *square that adds to $4M$ to get a square, then you have a value of 'b'*
- > *which MUST be a factor of M , though it may be one of those trivial*
- > *factors.*
- >
- > *Now let's peer into that result more deeply, by considering factors*
- > f_1
- > *and f_2 , where $f_1 f_2 = M$.*

sci.math: Re: Integer factorization, basics

>
> Since $a = b+c$, then $c = a-b$, and if you have non-trivial factors then
>
> $c = f_1 - f_2$
>
> is a solution that works.
>
> Notice then that the square root becomes trivial, as I have
>
> $\text{sqrt}((f_1 - f_2)^2 + 4f_1 f_2)$
>
> which is
>
> $\text{sqrt}(f_1^2 - 2f_1 f_2 + f_2^2 + 4f_1 f_2)$
>
> which is
>
> $\text{sqrt}(f_1^2 + 2f_1 f_2 + f_2^2)$
>
> which is
>
> $\text{sqrt}((f_1 + f_2)^2) = f_1 + f_2$.
>
> Then I have
>
> $b = (-f_1 + f_2 + (f_1 + f_2))/2$
>
> which gives $-f_1$ or f_2 .
>
> So the congruence of squares method works by having the square root
be
> a sum of factors of your target number M .
>
> Simple. But it requires searching for some square that adds to $4M$ to
> give you a square.
>
> Here's an example: Let $M=15$, so $4M = 60$, and notice that you can
just
> *see* that adding 4, gives you 64, so you have $8 = f_1 + f_2$, and f_1
> $=5$, $f_2=3$, works.
>
> But you have to go looking for a square.
>
> I took a completely different approach to the factoring problem!
>
> Instead of doing something like using $ab=M$, with $a=b+c$, I do
something
> like
>
> $a_1 x + b_1 = f_1$
>

> $a_2 x + b_2 = f_2$
 >
 > and $A = a_1 b_2 + a_2 b_1$
 >
 > and taking
 >
 > $(a_1 x + b_1)(a_2 x + b_2) = f_1 f_2$
 >
 > and multiplying out I have
 >
 > $a_1 a_2 x^2 + Ax + b_1 b_2 = f_1 f_2$
 >
 > and I have the target $M = f_1 f_2 - b_1 b_2$
 >
 > so there's a LOT more upfront complexity than you saw with the
 previous
 > example.
 >
 > But I end up with a dependency on the factorizations of numbers other
 > than my target, as I can solve to get
 >
 > $x = (b_2 f_1 - b_1 f_2 - 2b_1 b_2)/A$
 >
 > and notice that
 >
 > $a_1 a_2 x^2 + Ax + b_1 b_2 - f_1 f_2 = 0$
 >
 > is
 >
 > $x(a_1 a_2 x + A) + b_1 b_2 - f_1 f_2 = 0$
 >
 > so I have a back route to x being a factor to $b_1 b_2 - f_1 f_2$.
 >
 > One difference from before is that x might be a fraction, so you
 focus
 > on its numerator.
 >
 > Trouble is, I have all these variables!
 >
 > How do I get a_1, a_2, b_1, b_2, f_1 and f_2 ?
 >
 > Well, it turns out that I pick $b_1 b_2$ and $f_1 f_2$, as well as A ,
 which
 > determines the rest in a very basic way using elementary methods
 which
 > can be seen in my paper at
 >
 > <http://groups.yahoo.com/group/simplefact/files/>
 >
 > and the result there shows a square root with a dependency on the
 > factors of A and a number I call T in the paper, which is just f_1

f_2.

>
 > Now then, I want to directly compare what I do, with how others have
 > usually approached the factoring problem, as using congruence of
 > square, you have
 >
 > $b = (-f_1 + f_2 + \sqrt{(f_1 + f_2)^2 - 4f_1 f_2})/2$
 >
 > with $b^2 + bc = M$, where the f 's are factors of M .
 >
 > That approach requires that you find some square $(f_1 - f_2)^2$, and
 > factoring is a hard problem with it.
 >
 > My approach gives
 >
 > $x = (b_2 f_1 - b_1 f_2 - 2b_1 b_2)/A$
 >
 > from
 >
 > $a_1 x + b_1 = f_1$
 >
 > $a_2 x + b_2 = f_2$
 >
 > and $A = a_1 b_2 + a_2 b_1$
 >
 > where the target $M = f_1 f_2 - b_1 b_2$,
 >
 > and the f 's are factors of a number I call T , which *you* can pick.
 >
 > Well, clearly I've just shifted factoring one number to factoring
 > another!
 >
 > And what if the other number is hard to factor as well?
 >
 > The problem with that reasoning is that in actual encryption
 > techniques, like those used by RSA, two very large primes are picked
 > to
 > get your hard number.
 >
 > Being large primes, they are rare.

How rare? For RSA640, you'll need 96 digit primes. Oh, that's right, your prime counting function can't handle 10^{96} , so you have no clue how rare. Luckily, nobody needs your prime counting function:

```
>>> import math
>>> n96 = 10**96
>>> n95 = 10**95
>>> pi_n96 = n96/math.log(n96)
>>> pi_n95 = n95/math.log(n95)
>>> prime_96 = pi_n96 - pi_n95
```

```
>>> prime_96
4.0667487669449242e+093
```

Rather a lot, actually.

>
> *It is improbable that randomly picking a large number will give you a
> number with only two large primes as factors, so for a reason
> previously seen as a strength of RSA's method, you now get a major
> weakness, with my approach, as more than likely you'll get a much
> easier surrogate number to factor.*
>
> *And so what if your first choice were still hard? You can pick
> another. And if that were hard you could pick another.*
>
> *A dedicated computer could go through thousands of possibles in
> milliseconds.*

And how many millennia would it take to check $4 \cdot 10^{93}$ possibilities?

Christ, Harris, you're really slipping when you post something so blatantly wrong that even I can see it.

>
> *Then once you have a surrogate fully factored, you just have to pick
> every combination of its factors, and eventually one of them--if this
> method actually does work--will factor the number.*
>
> *I say "if" because I have a lot of theory at this point, and no
working
> model.*
>
> *The paper I have doesn't solve all of the engineering problems behind
> the approach. And you still have to be able to factor a fairly huge
> number.*
>
> *But I think it's worth talking out, and raising a warning, as someone
> might be able to solve all the engineering details VERY quickly.*
>
> *Some of you have made a career of lying about my work, and trying to
> convince others that it is not worth considering, that it is "junk",
> and you may choose to do so now, for whatever reasons are motivating
> you miserable people.*
>
> *However, if this approach does work, then far more clearly than with
my
> past research, your tactics will be detrimental to the world at
large,
> immediately.*
>
> *Time is a factor. Desperate people may be the only ones willing to*

try

> *out some wild idea of a known "crank", and those desperate people may*
> *then be the only ones who have a technique to break Internet*
> *encryption--if this works.*

>

> *If so, then those desperate people will--partly because of you*
> *people--have the key to the world handed to them on a platter with no*
> *one watching out for them, no one looking to protect, no one acting*

to

> *defend until they've had their way, possibly for some time.*

>

> *If that happens, then you people probably will have done a lot to*
> *destroy the world as it is currently known, for whatever reasons*
> *motivate you to lie about mathematics.*

> *It will be on your heads.*

>

>

> *James Harris*