

Re: Public Key, Symbolic Calculation

Source: <http://www.derkeiler.com/Newsgroups/sci.crypt/2005-02/1017.html>

tomstdenis_at_gmail.com

Date: 02/04/05

Date: 4 Feb 2005 09:31:14 -0800

Kiuhnm wrote:

> *tomstdenis@gmail.com* wrote:

>> *The problem with this scheme is the set R requires high enough*

>> *precision or you will have a lot of messages you can't decrypt.*

You

>> *can mitigate this by limiting the precision or using finite fields*

>> *[e.g. $GF(p)$] but that's about it.*

>

> *No, the precision is *absolute* by using symbolic calculation (like*

in

> *Mathematica or Maple).*

The roots won't always be nice numbers. Unless you specifically construct the polynomial to have integer roots [or roots with limited precision].

>> *Also factoring polynomials is much easier than factoring integers.*

So

>> *once you factor the polynomial you're done for.*

>>

>> *...Also you don't say what the trap door is. You find r_i from*

p ...

>> *but p is the public key. Can't an attacker just find them as well?*

>

> *The trick is to use "symbolic calculation" in order to prevent the use*

of numerical algorithms (M has "symbolic" coefficients so the knowledge

of numerical solutions is useless).

> *As you know (Galois-Abel-Ruffini), is very difficult to find the*

> *"symbolic" solutions of a polynomial equation of degree greater than*

4.

What the hell is "symbolic solutions" did sci.crypt just get another JSH? A polynomial is just a polynomial. An evaluation of a polynomial at a point would be "numerical".

So you're saying your polynomial is then of the form

$$p(x) = a + bx + cx^2 + dx^3 + \dots$$

???

That's hardly unique [nor would it have roots without being in terms of variables... e.g. everyone would have the same roots].

> *But how can I create a polynomial equation with real or complex roots in*

> *a way that those are not obvious?*

> *If I write*

> $34x^5 + 23x^4 + 2x^3 - 7x^2 - 76x + 43$

> *the solutions are not obvious, but this is true even for who has created*

> *the polynomial!*

The problem is root finding isn't that hard. For most roots simple things like newton-raphson can find roots.

...

This scheme isn't well thought out because it involves rather inefficient storage or generation of polynomials...

Tom