

Re: WAS Frobenius, so good?

Source: <http://www.derkeiler.com/Newsgroups/sci.crypt/2005-01/0911.html>

From: Cristiano (*cristiano.pi_at_NSquipo.it*)

Date: 01/15/05

Date: Fri, 14 Jan 2005 23:16:18 GMT

Marcel Martin wrote:

> *Cristiano a écrit :*

>>

>> *Marcel Martin wrote:*

>>> *Cristiano a écrit :*

>>>>

>>>> *Marcel Martin wrote:*

>>>>> *Assuming $b=1$ for FR (the fastest case), both FR*

>>>>> *and BW make use of the same algorithm to compute the Lucas*

>>>>> *sequence. Which program do you use for BW?*

>>>>

>>>>> *I translated an UltraBasic implementation (which implements the*

>>>>> *extra strong Lucas pseudoprime test with method A*) in C++ code.*

>>>>

>>> *So you compared the running times of FR with "frobenius.c" (which*

>>> *is written in C and makes use of gmp which is also written in C and*

>>> *which is one the fastest existing C libraries) with a program*

>>> *written in C++. Which library does your C++ program use?*

>>

>> *I took the times of the BW test running on Commodore 64 Basic V2. :-)*

>>

>> *C'mon, Marcel; I done the comparison using *exactly* the same*

>> *"parameters": all the tests are in the same program, they are*

>> *invoked in the same way and they have been written using the same*

>> *"technique" (language, structure, optimizations, ...).*

>

> *Great. So now, it just remains to know which programs you are*

> *comparing.*

BW

verbatim translation of UB's BWppt2

MR

HAC's Miller-Rabin code *only* to base 2 without

 If $y = 1$ then return("composite").

inside the while (it happens in very rare cases, so that the code with the 'if' is a bit slower)

sci.crypt: Re: WAS Frobenius, so good?

FR

frobenius.c with 1 iteration in which I changed a and b parameters:
their initial value is 2

mpz_urandomm() in the do..while loop replaced with a=a+1, b=b+47
this way I get the fastest code with no errors (I tested FR only up to 2^{33})

new entry: WD

Wei Dai's IsLucasProbablePrime() in Crypto++ (module nbtheory.cpp)
as I said, I have put mpz_perfect_square_p() before the while loop.

>> *Some years ago I have found the BW implementation in UB. My first
>> translation was for MIRACL with its C++ wrapper, but few weeks ago I
>> translated that code in C code for GMP (optimized for the Athlon).*

>

> *No, you have not found the BW implementation but a BW
> implementation. Is it the file "bwppt1.ub" that you translated? If
> so, of course, it is slower than "frobenius.c" (notice that if you
> translated "bwppt2.ub", that's even worst).*

Now you know that I *used* the verbatim translation of BWppt2.ub.

I say "used" because the very slow BW test can now be safely replaced by the
fast WD Lucas test.

In fact, the only reason for which I need the Lucas test is to do MR + WD
(because of the "famous" conjecture).

I tested MR + WD up to 2^{35} : no failure (in 7 hours I'll get the result for
 2^{36}).

In my program to find primes (which includes the sieving) MR + WD is only
0.6% slower than MR, but I need more testing...

Cristiano