Applied Mathematics for Reversers II

(including Crackme)
by Haldir[RET]

This essay is just a little addition to the previous one, with some corrections and additional hints for my crackme.

1. Solving a system of linear equations which is underdetermined means that you have
many solutions, my explanation was for the case that you need one specific solution, maybe because it is checked against other values, with the solution presented in Part I you can cycle through every possible solution.
I'm sorry if this wasn't clear enough.

2. 15^2 = 225 ;)
Now to the new part:

You might have asked yourself, how the hell can you recover those huge arrays (matrices) from a program.
Well you can again set a system of linear equations up for it:

Remember this:

```
for (i=0; i<sizeof(multipliers)/4; i++)
{
  multiplier = multipliers[i];
  for (j=0,sum=0; j<sizeof(m_1); j++)
  sum += (ushort)(mod1[j] * multiplier[j]);
  mod2[i] = sum % 0x0F; // Here you see our Modulo Operator %
}
```

So somewhere in our program you will have a "sum" for each loop. Now in this case we want to recover the array multiplier, so we feed in several mod1 arrays, record the sum and set some new linear equations up. Let's say your Array is 16*16, then you need 16 sums for each line and 16 input mod1 arrays to recover one line of the multiplier array, just do that 16 times and you have your complete multiplier array, just write a little program which does that and that's it. That's very useful if the application used loop unrolling and you don't see the two "for loops" clearly.

Ok, now to my little Crackme:

Your task: Write a keygen (no patching) for it and a short tutorial :)

The number layout is: <11digits>-<4 letters>-<4 letters>-<4 letters>-<4 letters>

Example: 01234567890-JHIH-BTHE-MKFD-RGSG (<-- valid serial number)
Run it like: crackme.exe 01234567890-JHIH-BTHE-MKFD-RGSG

It's based on a serial number scheme I saw last year, it was the reason to write
my two essays about modular arithmetics.

Some hints:

1.) Read my first essay
2.) I added some strings for the different checks in the crackme
3.) It's compiled with Intel C Compiler, with some fairly heavy optimization settings (loop-unrolling *hint*)
4.) Use NTL
5.) The matrix sizes are between 11*11 and 16*16
6.) Don't even think about bruteforcing it, we're talking about around 2^70 possible combinations ;)

This crackme is hard, if you beat it you can be sure that you understood the math behind it :)

Greets,
Haldir[RET]
03/05/03