HS Projects In Classical Cryptography

by William Gasarch

Here is a list of projects and ideas for projects. They can also be combined.

1. Code up RSA, DH, and ways to crack them that are on the slides. See how well the methods-to-crack do.

2. If you do the factoring or Discrete Log projects then use them to try to crack RSA.

3. For RSA you find a (p, g), prime and generator as follows:

   (a) Input n (you want an n-digit prime)

   (b) KEY STEP: Pick a random string x of length $n-1$ and then let $p = 1x$ and $q = \frac{p-1}{2}$.

   (c) Test if p is prime. If not then goto KEY STEP

   (d) (p is prime) Test if q is prime. If not then goto KEY STEP

   (e) (p and q are prime). OTHER KEY STEP: Pick $g \in \{2, \ldots, p-2\}$ at random.

   (f) If $g^2 \not\equiv 1 \pmod{p}$ AND $g^q \not\equiv 1 \pmod{p}$ then output (p, q) and our done. If not the goto OTHER KEY STEP.

   There are two ways to speed this up:

   - Rather than pick a random x pick an x such that $p = 1x$ is not even. Or also is not div by 2 or 3, or ....
   - Rather than insist that $p-1 = 2q$ where q is prime you could relax this to insist that either $p-1 = 2q$ OR $p-1 = 6q$. If $p-1 = 6q$ then to test if g is a generator you need $g^2 \not\equiv 1 \pmod{p}$ AND $g^3 \not\equiv 1 \pmod{p}$ AND $g^q \not\equiv 1 \pmod{p}$. (One can extend this to, say, $p = 30q$.)

   I want to know if this technique really does speed things up. The calculation is so fast that I don't think timing how many nanoseconds will be informative; however, you can ount the number-of-operations (only count mults and mods).