Experiments in Literate Programming

Brian Beckman

August 18, 2013

Contents

Abstract

Fast, parallel factoring of integers is the “Hello, world!” of good hackers and of bad guys.

The problem looks like this: given a big integer $n \in \mathbb{N}$ such that

$$n = pq$$

where p and q are big primes, find p and q.

RSA uses numbers like n as public encryption keys. Anyone can encrypt you a message using n. Decrypting is easy only if you know p and q.

Because RSA and most of internet security depends on the assumption that factoring is hard, this problem is critical. Whoever can ‘break’ RSA by factoring keys will ‘own the world’ for a short time, until internet security is reformulated in some new way.

The biggest RSA key factored as of *<2013-08-18 Sun>* is

```
(defn wrapped-lines-to-bigint [& strs]
  (bigint (apply str strs)))

(def RSA-768
  (wrapped-lines-to-bigint
   "12301866845301177551304949583849627207728535695953347921"
   "97322452151726400507263657518745202199786469389956474942"
   "77406384592519255732630345373154826850791702612214291346"
   "16704292143116022212404792747377940806653514195974598569"
   "02143413"
   ))

(def TEST
  (*
   (wrapped-lines-to-bigint
    "33478071698956898786044169848212690817704794983713768568"
    "91243138898288379387800228761471165253174308773781446799"
    "9489")

   (wrapped-lines-to-bigint
    "36746043666799590428244633799627952632279158164343087642"
    "67603228381573966651127923337341714339681027009279873630"
    "8917")))

(== RSA-768 TEST)
```

⟶

```
true
```

1 Literate Programming With Org-Mode and Clojure

Today's internet security depends upon factoring being hard. If factoring isn't hard, internet security must be rethought in new terms.

First, add the following to the `:dependencies` section of your Leiningen *project.clj* file:

```
[org.clojure/core.contracts "0.0.5"]
```

2 First Experiment

In line 1 of section 1, the Introduction, and perhaps even in 3, But Wait, There's More, the version of **contracts** was specified; this doesn't yet have anything to do with internet security.

Next, shift attention to the file "core.clj," which implements the principal functions of our demonstration.

```
2  (ns big-prime.core
3  (:import java.util.Random)
4  (:use [big-prime.utils]
5        [big-prime.sqrt :as nt]
6        [clojure.core.contracts :as contracts]
7        [clojure.set :only [difference]]
8        ))
```

3 One More

The mass of the sun is $\mathrm{M}_{\mathrm{sun}} = 1.989 \times 10^{30}$ kg. The radius of the sun is $\mathrm{R}_{\mathrm{sun}} = 6.96 \times 10^{8}$ m.