Pushing the boundaries of cryptography in a security vulnerability report

devblogs.microsoft.com/oldnewthing/20180724-00

July 24, 2018

Raymond Chen

A security vulnerability report arrived that implied that the finder had made an earth-shattering breakthrough in cryptography. Specifically, the finder claimed to have found an efficient way to factor the large numbers used in RSA cryptography.

This would be a remarkable breakthrough if true, but the description of the algorithm, while mathematically correct (and I bothered to read through it all and understand it), didn't actually produce an efficient algorithm. It boiled down to trying to find a collision among a population whose size is proportional to the smaller factor, with an additional optimization to reduce the amount of calculation to the square root of the population space. (I believe it was this additional optimization that the finder considered to be the groundbreaking discovery.)

Now, a geometric reduction in complexity is a great thing, but it's minuscule compared to exponential growth.

In 2013, Web browsers required a minimum of RSA-2048, which means that the collision space is around 2^{1024}, and the birthday paradox tells us that you'll need to generate around 2^{512} items to have a 50% chance of finding a collision. Applying the groundbreaking discovery reduces the number of items to 2^{256}.

This is even worse than enumerating all the GUIDs. At least there are only 2^{128} of those.

This algorithm for factoring an RSA-2048 number would require storing 2^{256} values. That's the *square* of the number of GUIDs.

We calculated earlier that storing all the GUIDs on SSDs would require 100 earth-sized planets. Storing all the values required for this algorithm to factor an RSA-2048 number would require, um, a lot more than that.

Current upper estimates for the mass of the Milky Way put it at 4.5×10^{12} solar masses, or (rounding up) 10^{19} earth masses. If we need 100 earth masses to store 2^{128} 128-bit values, then storing 2^{256} 1024-bit values will require around $2^{132} \times 100 \cong 10^{41}$ earth masses $\cong 10^{22}$

Milky Way-sized galaxies.

This seems impractical.

The finder, however, disagreed with our analysis and insisted that their trial runs with smaller values indicated that the running time was linear in the exponent. “I was able to factor numbers up to 64 bits in size, with the largest taking less than a second.”

We decided to take a tip from a number theorist who had to deal with factorization algorithms submitted by crackpots and suggested to the finder that they use their advanced algorithm to factor one of the root signing certificates.

The finder replied back, “I know what you’re trying to do, but I’m telling you that I cannot run the algorithm on numbers that large on my laptop. But you can certainly run it on one of your more powerful computers. I have demonstrated that the algorithm is linear in the key length, and my personal lack of access to supercomputers does not invalidate that fact. I have contacted the media about this discovery, but fortunately for you, they don’t seem to be interested, which gives you more time to address the problem.”

If the algorithm were truly linear in the exponent, then going from 64-bit numbers to 2048-bit numbers would take only 32 times as long. The 1-second run time would increase to just 32 seconds. So let it run for a minute. Five minutes just to be sure. Your laptop can certainly handle that.

But instead of replying, we decided to disengage. Never wrestle with a pig. You get dirty, and the pig likes it.

Raymond Chen

Follow