Fremont, California

August 4, 2017

No commercialization of the large math library without a license is granted. The integer factorization utility is free and available for download at the bottom of this page.

This is for research and development.

This article is based on a paper presented to IEEE Smart World Cyber Trust Workshop August 4, 2017 and published by the IEEE Advanced Trusted Computing.

http://www.wnlabs.com/news/IEEE_copyright_WN_papers.php

http://www.wnlabs.com/pdf/IEEE-ATC-R-CyberTrust-27.pdf

Download this web page as a pdf for easier reading and reference:

http://www.wnlabs.com/pdf/ECC_is_easier_to_break.pdf

Article

abrisson@wnlabs.com

Breaking ECC is easier than breaking RSA. Here's why. Here's how. Here is the integer factorization utility.

It is common discussion in the Cyber Belt that facilities for quantum computing and conducting quantum computing attacks has been built.

This paper examines why it is easier to break ECC Elliptical Curve Cryptography than it is to break RSA Integer Factorization Cryptography. Public key networks are almost completely dependent upon both ECC and RSA for their security controls.

PKI uses ECC and IFC to create the public-private key pairs (modulus and prime number composites) for the asymmetric framework to work. ECC has become the preferred algorithm since it is believed that ECC creates keys of comparable strength to RSA with smaller key sizes which in turn reduces high computational overhead required to process these complex algorithms.

The benchmark of key strength relative to key size is comparing the strength of these asymmetric algorithms against a symmetric key algorithm like Whitenoise. RSA 1024 bits is roughly equivalent to the strength provided by an 80 bit symmetric key. This bodes well for next generation security since the smallest (way out of scope) Whitenoise key that can be created is 1600 bits.

As factorization skills and techniques improve, and as processing becomes faster and faster, cryptologists look for signs of vulnerability. Then a body like NIST should dictate longer key sizes and more complex algorithms to try to regain a sufficient level of security for different security controls and different data secrecy levels. These decisions are influenced by political and commercial expediencies. So critical is an accurate assessment of factorization resistance that in 2007 four hundred computers were used to share the work to break a 1024 bit prime number composite in eleven months to show the impending and certain vulnerability of RSA.

ECC appeared to be secure with shorter keys relative to RSA. NIST states that ECC keys should be twice the length of a symmetric key like Whitenoise. Because Whitenoise keys can be any length up to 10^60 it is a relationship that ECC cannot sustain. Furthermore, NIST hedges this recommendation with a disclaimer that their statement is true if no major breakthroughs like the Boren factorization method have been made in solving the underlying arithmetic problem.

NIST and NSA already recognizes the vulnerability of both ECC and RSA to brute force attacks from tests likely conducted on quantum computers they have. As a result, it was announced in 2015 that NIST and NSA plan to transition away from the use of ECC for secret and top secret data. RSA is already degraded and relegated to the lowest level, non-secret government use. Brute force attacks that break RSA break smaller ECC modulus faster than factoring larger RSA modulus.

The IEEE paper examines a technique for the fast factorization of prime number composites. This simple approach combines Newton bisection and a simple remainder volitility test for faster factoring.

Public key infrastructures requires public-private key pairs to create a modulus (a prime number composite.) This is fundamental to the PKI architecture and the process is used for certificates, signatures, authentication, encryption etc. and are found in many different areas, many different sizes, and many different implementation approaches like simple seeding in asymmetric network environments.

ECC and IFC are simply two different arithmetic processes that allow the selection of primes to be used in the PKI process and handshakes. It was thought that each had different levels of difficulty in reversing the process to determine those prime factors if brute force attacks are used or arithmetic factorization approaches like sieves are used.

The Boren rapid factorization method simply bypasses those mathematical approaches completely using an approach leveraging Newton's bisection theory and a simple comparative remainder volitility test.

This utility has been shown to CSE. This utility was first demonstrated publicly to the Telecom Council of Silicon Valley.

It was demonstrated publicly at an RSA conference in San Francisco that Whitenoise Laboratories attended as a delegate of an International Trade Canada Mission.

This rapid factorization approach was again demonstrated at the IEEE Smartworld Trusted Computing workshop on August 4, 2017.

This rapid factorization approach is published in the IEEE Advanced and Trusted Computing 2017 Journal. Qualified mathematicians and computer scientists can rapidly build their own factorization utility to test and validate and improve upon this fast factorization research. It can be found in the conference proceedings and in the IEEE Explore digital library.

Factorization of ECC is significant because it shows that we need to move to next generation cryptography.

The best current approach and sieve to attack ECC is the Lenstra elliptic curve factorization approach.

A publicly available example and explanation of the Lenstra approach can be found at

https://en.wikipedia.org/wiki/Lenstra_elliptic_curve_factorization.

It is quoted and used herein.

"Why does the Lenstra factorization algorithm work?

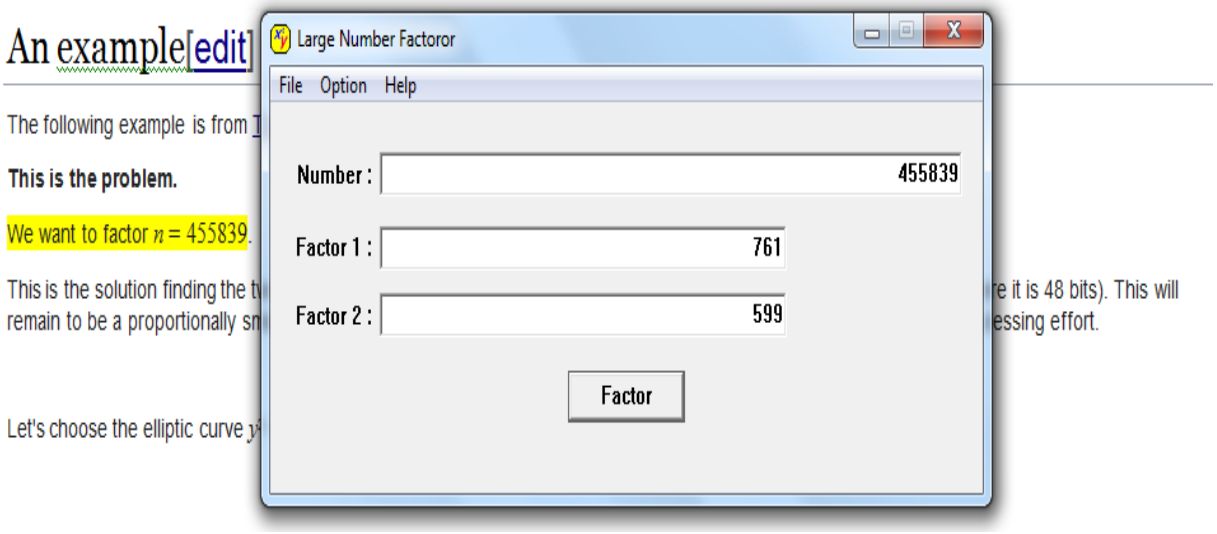
If p and q are two prime divisors of n, then $y^2 = x^3 + ax + b \pmod n$ implies the same equation also modulo p and modulo q. These two smaller elliptic curves with the addition are now genuine groups. If these groups have N_p and N_q elements, respectively, then for any point P on the original curve, by Lagrange's theorem, $k > 0$ is minimal such that on the curve modulo p implies that k divides N_p; moreover, . The analogous statement holds for the curve modulo q. When the elliptic curve is chosen randomly, then N_p and N_q are random numbers close to $p + 1$ and $q + 1$, respectively (see below). Hence it is unlikely that most of the prime factors of N_p and N_q are the same, and it is quite likely that while computing eP, we will encounter some kP that is 8 modulo p but not modulo q, or vice versa. When this is the case, kP does not exist on the original curve, and in the computations we found some v with either $\gcd(v, p) = p$ or $\gcd(v, q) = q$, but not both. That is, $\gcd(v, n)$ gave a non-trivial factor of n.

An example

The following example is from Trappe & Washington (2006), with some details added.

This is the problem. It is striking that just eleven years ago that an n of the size shown was considered a difficult problem. ECC security is degrading rapidly.

"We want to factor n = 455839."



The Whitenoise factorial utility finds such a small ECC n instantly. This example is only 48 bits. The ECC modulus will always remain proportionally smaller to RSA modulus which can be factored. You see in the dialog box that we find the two factors for 455839.

The accuracy of the factorization can be tested by plugging in the values of P (Factor 1 and Factor 2) into the elliptic curve formulae below. This simply confirms for us whether we were accurate and it will tell us what the coordinates for the corresponding points on the elliptical curve are if we were interested. (That doesn't provide us with any additional information pertinent to security.)

Let's choose the elliptic curve y 2 = x 3 + 5 x - 5, with the point P = (1, 1) on it, and let's try to compute (10!) P .

The slope of the tangent line at some point A =(x , y) is s = (3 x 2 + 5)/(2 y) (mod n) . Using s we can compute 2 A . If the value of s is of the form a/b where b > 1 and gcd(a , b) = 1, we have to find the modular inverse of b . If it does not exist, gcd(n , b) is a non-trivial factor of n .

First we compute 2 P . We have s (P) = s (1,1) = 4, so the coordinates of 2 P = (x' , y') are x' = s 2 - 2 x = 14 and y' = s (x - x') - y = 4(1 - 14) - 1 = -53, all numbers understood (mod n). Just to check that this 2 P is indeed on the curve: (-53) 2 = 2809 = 14 3 + 5·14 - 5.

Then we compute 3(2 P). We have s (2 P) = s (14,-53) = -593/106 (mod n). Using the Euclidean algorithm : 455839 = 4300·106 + 39, then 106 = 2·39 + 28, then 39 = 28 + 11, then 28 = 2·11 + 6, then 11 = 6 + 5, then 6 = 5 + 1. Hence gcd(455839, 106) = 1, and working backwards (a version of the extended Euclidean algorithm): 1 = 6 - 5 = 2·6 - 11 = 2·28 - 5·11 = 7·28 - 5·39 = 7·106 - 19·39 = 81707·106 - 19·455839. Hence 106 -1 = 81707 (mod 455839), and -593/106 = -133317 (mod 455839). Given this s , we can compute the coordinates of 2(2 P), just as we did above: 4 P = (259851, 116255). Just to check that this is indeed a point on the curve: y 2 = 54514 = x 3 + 5 x -5 (mod 455839). After this, we can compute .

We can similarly compute 4! P , and so on, but 8! P requires inverting 599 (mod 455839).

The Euclidean algorithm gives that 455839 is divisible by 599, and we have found a factorization 455839 = 599·761.

The reason that this worked is that the curve (mod 599) has 640 = 2 7 ·5 points, while (mod 761) it has 777 = 3·7·37 points. Moreover, 640 and 777 are the smallest positive integers k such that kP = 8 on the curve (mod 599) and (mod 761), respectively. Since 8! is a multiple of 640 but not a multiple of 777, we have 8! P = 8 on the curve (mod 599), but not on the curve (mod 761), hence the repeated addition broke down here, yielding the factorization."

Advance research! Join the challenge to optimize this utility and large number math library. Define the underlying operations that make the Whitenoise Boren factorial utility work.

DOWNLOAD

RAPID ECC FACTORIZATION RESEARCH MATERIALS: factoring executable, big math library, scope documents, code etc.

The large number math library intends to allow manipulation of 500 to 700 digit numbers on a typical desktop or laptop computer and it is included. The application will allow you to test and factor some sample composite primes. Grab two primes from prime number lists. Multiply them together and then use the utility to factor them. This does not require an install. It can be run from its folder. There are scope and design documents as well as sample code.

Bookmark this page and visit often. We will be announcing major ECC factoring challenges:

Can you optimize the code or library? The source code and math library are provided. What is the largest composite prime you can factor?

How long will it take you to factor a 256 bit ECC composite prime (128 bit strength) like ECC Curve25519?

Mathematically explain and prove why the 6th derivative of a remainder points to the correct prime? This would be a significant contribution to General and Prime Number Theories.

SIMPLE BENCHMARK COMPARISON OF BOREN VS LENSTRA FACTORING SPEED

The Lenstra elliptical curve factorization technique is considered the benchmark. However, "ECM is considered a special purpose factoring algorithm as it is most suitable for finding small factors. Currently, it is still the best algorithm for divisors not greatly exceeding 20 to 25 digits (64 to 83 bits or so), as its running time is dominated by the size of the smallest factor p rather than by the size of the number n to be factored." Make an 80 bit prime number composite. Time how long the Lenstra method takes to determine the factors. Run the same prime number composite through the Whitenoise Boren factoring utility and it will do numbers that size easily on a decent notebook. https://en.wikipedia.org/wiki/Lenstra_elliptic_curve_factorization

The Lenstra EMC elliptical curve factoring is currently the best approach. However, in his presentation Professor Lenstra indicates that since the 1970s general factoring has been stuck and is running out of steam. See: Cryptanalysis of Public Key Cryptographic Algorithms - http://cs.ucsb.edu/~koc/ccs130h/notes/lenstra.pdf.

Lenstra indicates that there is a need for an entirely new approach. The Whitenoise Boren Newton Bisection and volatility testing technique is a radical new approach to factoring that merits study, optimization and better definition.

This approach was developed for Whitenoise Labs by mathematician and computer scientist, Stephen Lawrence Boren, who is a quadriplegic and co-founded the original Whitenoise Labs www.wnlabs.com . Because he programmed with the back of one knuckle, his code is poorly remarked.

Using Lenstra's example from his paper, we know that 80 bit security should take more than 3 million processing years effort on a 10GHz machine to factor. Rapidly doing a benchmark test on such samples will show this approach works as quickly as EMC. This merits further research, optimization and understanding.

Download the Whitenoise Boren Newton Bisection utility from above.

Compare to Lenstra EMC online integer factorization calculator: online Lenstra EMC utility: https://www.alpertron.com.ar/ECM.HTM

For additional information or to provide feedback write to Andre Brisson: abrisson@wnlabs.com or correspond through Linked In at https://www.linkedin.com/in/andre-brisson-51077/?trk=nav_responsive_tab_profile_pic