Rapid Factorization of Composite Primes

An alternative to the sieve method

André Brisson
Whitenoise Laboratories Canada Inc.
#701 – 1736 W. 10th Ave.
Vancouver, British Columbia Canada V6J 2A6
abrisson@wnlabs.com

***Abstract*—Trusted computing is based on PKI and a 2048-bit RSA public and private key pair that is created randomly on the chip at time of manufacture and cannot be changed. Trusted Computing relies on the secrecy of a prime number composite which is the product of two prime factors. Reversing this is factorization. The security underpinning is that prime number composites can only be attacked by brute force and that the work space is so large that this kind of attack is not feasible in any usable time frame.**

The factorization technique examined in this paper is novel because it dramatically reduces the factorization work space, leverages the Newton bisection theory and performs a simple comparison. It might further be leveraged with prime number dictionaries.

Keywords—RSA; factorization; breaks; hacks; sieves;Newton; Lenstra; Fermat; NIST; quantum computing attacks; dictionary attack

I. INTRODUCTION

This technique starts by taking the square root of a prime number composite (modulus) to radically reduce attack work space. Thereafter, Newton's bisection theory is applied and a simple volatility test tells us which direction one of the prime factors is located relative to subsequent bisections. The Whitenoise factorial utility has a large number math library capable of handling numbers many hundreds of digits long on a basic computer that can be easily expanded upon for increasing number sizes.

Starting in 1991 RSA ran factoring challenges to encourage research and to validate their security proposition which requires the secrecy of prime factors of a modulus. They no longer hold these challenges.

Quantum computing attacks are now an existential threat to any mathematical or arithmetic based cryptography like RSA Integer Factorization Cryptography (IFC) and Elliptical Curve Cryptography (ECC). Factorization algorithms for successful attacks already exist and include the quadratic sieve, general number sieve, continued fraction factorization, and Dixon's factorization. Elliptical curves can be factored with the general number field sieve, the multiple polynomial quadratic sieve and the Lenstra "elliptic curve factorization method (ECM)". [1]

The sieve techniques extend Fermat's work satisfying the equality $n = x2 - y2$ and then factoring with $n = x2 - y2 = (x + y)(x - y)$. Those approaches work but they are slow.

The significance of quantum computing attacks is that they threaten traditional cryptography with sheer computational speed. Its advantage is mechanical and not mathematical.

We will first look at the Whitenoise Factorial Utility which exploits Newton's bisection theory and a simple volatility test. Next we will look at the feasibility of prime number dictionary attacks beginning first with smaller composite primes and moving up to the size of RSA IFC keys actually recommended by NIST.

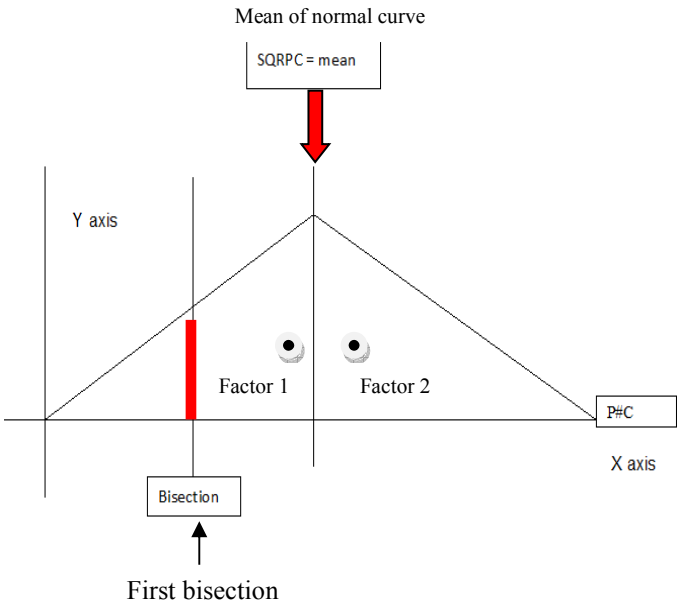
Scientific advancement usually begins with the observation of a phenomenon that stimulates research leading to a scientific or mathematical explanation. The mathematician that can define the observed phenomenon will advance the understanding of prime number behavior in general number theory. They would impact how we approach advanced computational and computer sciences. It would dramatically impact cyber security.

It was observed that if one takes the 6th derivative of two points within a range on a normal curve, the larger point divided by the smaller point, that the closer you are to the factoring solution the more volatile the remainder becomes. This then becomes the primary criteria in a decision tree in deciding which way to jump if one is deploying a bisection method to try to locate a prime number factor.

"On a basic system deploying this technique, it is possible to factor 60 to 80 bit prime number composites in a fraction of a second. It is believed that expanding upon this technique will make the factoring of large prime composites an easy task and hold out great possibilities for the manipulation of primes in general number theory." [5] **A video demonstration of this is available on YouTube. [2] The utility is available for researchers upon request and permitting.**

It is to be noted that this is not a simple mathematical shortcut but because of the ready access to computers and even the basic computational speed of a simple desktop computer, it is a technique that should be far superior to existing techniques and hold out great possibilities.

II. Rapid Factorization Architecture



If one looks at a Cartesian plane and a [very poor] rendition of a normal curve, the following elements are readily identifiable.

- We can draw a normal curve with the endpoint of the curve on the X-axis being equal to the prime number composite [P#C].
- We can take the square root of the prime number composite and this becomes the square root of the prime composite (SQRPC) bisection.
- We know that one of each of the two factors we are looking for will reside somewhere on either side of the SQRPC.

Looking at the graph above, we will search for the smaller prime factor (factor 1) of the prime composite to the left or small end of our bisection. Our first step is to bisect the smaller area of the curve along the x-axis at the red line.

First, we rapidly test to see whether the point of bisection is a whole number. If it is (unlikely) then we have found one of our factors. If this was the case, we simply divide this number into the prime composite.

The challenge now is to decide on which direction of the bisection line we should jump in order to make the next bisection.

We take the 6th derivative of two points within a range on a normal curve, the larger point divided by the smaller point. The closer you are to the factoring solution the more volatile the remainder becomes. This is performed on both field sides of the last bisection. We jump to the side of the greater volatility of the remainder to make the next bisection.

This continues until the factor is found and it divides evenly into the prime composite.

The Whitenoise Large Factor utility to factor prime composites can be further optimized in several ways: create a 64-bit and greater utility, parallel processing and multi-core processing. Or, run it from a quantum computer. Researchers may discover other factors to use with the decision tree process.

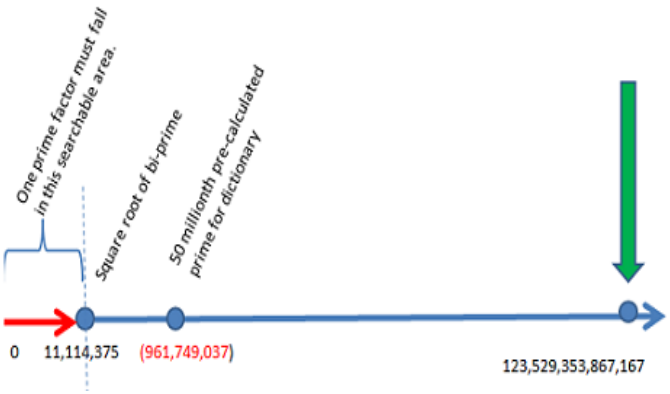
III. Attack Work Space

Here is an example of a 128-bit prime composite:

11,114,329 X 11,114,423 = 123529353867167. By adding commas we see that the prime composite value is 123,529,353,867,167.

A simple brute force attack would test every single number from 1 to 123,529,353,867,167 including non-prime numbers to see which number divides evenly into our prime composite value. When we have discovered the small factor because it divides evenly into the prime composite value then we have also identified the other, larger factor. However, testing over 123 trillion numbers is a BIG task. Doing one test per second of all possibilities could take over 3,900 centuries to find the brute force solution just for this small example.

This is the most important step. If we take the square root value of the prime number composite then we **KNOW** that this will bisect the range of possible values (123,529,353,867,167) in a manner that insures that one prime factor will fall **BELOW** the square root value and one prime factor will fall **ABOVE** this value. This is shown below where we see that the square root of the prime number composite falls between the two prime numbers we multiplied together.



In our example: The square root of 123,529,353,867,167 equals 11,114,376 and we see that: Factor 1 (11,114,329) < square root of the bi-prime key - 11,114,376 < Factor 2 (11,114,423).

By solving for the smaller prime factor value we are reducing our "brute force testable universe" from 123,529,353,867,167 calculations to 11,114,376 calculations. This is a small fraction of the size of the range that would otherwise need to be tested. In our example, this step reduced the work space by 99.99999101 percent. That is a significant reduction in the work effort required to factor prime composites. Adapting known factorization techniques to incorporate just this step and bisections would be an existential challenge to IFC cryptography.

The practical problem with IFC cryptography is that simply increasing key size increases computational requirements significantly with very little payoff in terms of actual key strength. A 2048-bit modulus is 258 characters long. It takes a lot of effort to process such a number. Elliptical Curve Cryptography became preferred because much smaller numbers created equivalent key strengths.

The National Institute of Science and Technology (NIST) oversee the recommendations for network and data security for the United States federal government. **NIST has determined that 1024 bit IFC keys provide only 80-bit strength security and that 2048 bit keys provide only 112 bit strength**.

Table 2: Comparable strengths

Security Strength	Symmetric key algorithms	FFC (e.g., DSA, D-H)	IFC (e.g., RSA)	ECC (e.g., ECDSA)
≤ 80	2TDEA[21]	$L = 1024$ $N = 160$	$k = 1024$	$f = 160$-223
112	3TDEA	$L = 2048$ $N = 224$	$k = 2048$	$f = 224$-255
128	AES-128	$L = 3072$ $N = 256$	$k = 3072$	$f = 256$-383
192	AES-192	$L = 7680$ $N = 384$	$k = 7680$	$f = 384$-511
256	AES-256	$L = 15360$ $N = 512$	$k = 15360$	$f = 512+$

"The number of bits of security is not necessarily the same as the key sizes for the algorithms in other columns, due to attacks on those algorithms that provide computational advantages." [3]

We see that RSA IFC has only one step up to 3072 bits before it is discontinued. RSA IFC seems seriously downgraded from NIST perspective even though the "NIST recommendation applies to U.S. government agencies using cryptography for the protection of their sensitive, unclassified information." [3]

IV. RSA HASHES AND SIGNATURES – SMALLER TARGETS

Networks are complicated environments and integer factorization cryptography like RSA is used for security functions like signatures, hashes and certificates. NIST recommends using 2048 bit IFC keys for signatures (hashes). "RSA is approved in [FIPS186] for digital signatures." [3]

Each calculation of a 2048 bit IFC key requires a lot of computational resources and intense mathematical calculations to manipulate 256 digit prime composites used to accomplish handshakes for key agreement, encryption and other cryptographic functions.

Signatures and hashes for digest and data authentication (like Cycle Redundancy Checks) are the work horses of network security because they are called so often. In reality we find poor implementation by system architects and programmers. Many applications use much, much smaller prime composites for faster encryption and signature verification in high volume contexts or on small devices and components. Generally hackers attack the weakest link in any system.

V. RSA HASHES AND SIGNATURES – SMALLER TARGETS

A dictionary attack simply tests every possible solution contained within the dictionary with brute force. For 2048 bit prime composite RSA modulus this number will be 256 digits long. Its primes will be at least 1024 bits or 128 digits long. (That is the maximum size of the smaller factor we would be looking for.)

Prime number theorem tells us that there will be about 10^305 possible solutions for a 2048 bit composite prime. The first reaction is that this is simply too large a number of possible prime numbers to make a dictionary of primes that is usable. The prime number theorem could not anticipate the rapid advancement of computer processing power and speeds since the advent of integer factorization cryptography (IFC – RSA). Nor did it anticipate the interest in just calculating larger and larger prime numbers.

"The new record prime determined by Cooper, Woltman, Kurowski, Blosser & GIMPS (January 7, 2016) is 22,338,618 digits long." [4] "Perhaps the longest lists ever calculated (but not all stored) are those corresponding to the maximal prime gape (and twin prime constant) projects. See Nicely's lists. At the time I last updated this page, these projects had found all the primes up to 10^{18} and are available." [4]

A 2048 bit prime number composite is 256 characters long. Both factorization techniques have a first step of taking the

square root of the composite prime length to radically reduce our work space. The square root of 256 digit number is a 16 digit number. To create a 10^305 list from scratch would be daunting but creating a dictionary of all primes up to 16 digits easily falls within the range of prime number lists that are already calculated and are available on the internet.

Making such a dictionary is possible. Using such a dictionary is definitely feasible on smaller prime composites such as those often found in signatures and hashes. Using dictionaries to assist in attacking 2048 bit prime composites (i.e. certificates or applications following NIST recommendations) can be leveraged by using the square root of the prime number composite as a starting point and further using the bisection process to reduce the work space **within** the dictionary.

VI. CONCLUSION

The Whitenoise Large Factorial Utility is effective and can be tested and validated. It can be optimized. It requires continued research to determine the underlying reason as to why the use of a 6^{th} derivative is instrumental in predicting the location of a specific prime number.

Starting any widely available sieve method utility with the square root bisection method may dramatically speed up those factorization utilities by significantly reducing work space.

The Whitenoise Large Factorial Utility is available to researchers with permitting.

VII. ACKNOWLEDGEMENT

Stephen Lawrence Boren, Vancouver, British Columbia conducted the associated research and programmed the factorial utility.

VIII. REFERENCES

[1] Richard Crandall *and* Carl Pomerance *(2001). Prime Numbers: A Computational Perspective. Springer.* ISBN 0-387-94777-9. Chapter 5: Exponential Factoring Algorithms, pp. 191–226. Chapter 6: Subexponential Factoring Algorithms, pp. 227–284. Section 7.4: Elliptic curve method, pp. 301–313.

[2] André Brisson Factorization of a prime number composite [Online] https://www.youtube.com/watch?v=GwkwgR_78dQ&feature=youtu.be

[3] http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-57pt1r4.pdf page 52

[4] Nicely – Prime number lists [Online] https://primes.utm.edu/notes/faq/LongestList.html

[5] André Brisson "Rapid Factorization of Semi Primes," [Online] http://www.wnlabs.com/pdf/Rapid_Factorization_of_prime composites.pdf pages 6