Reviewing Recent Prime Generation Methods for Breaking Cryptographic Keys

Mark Carney
University of Leeds

July 2019

Abstract

A recent publication by Grant *et al.* [2] has revealed some innovations with respect to the checking and generation of prime numbers with which to crack cryptographic keys. We argue that their method is minimal, and go on to prove some general cases of the mathematics they present - specifically refuting two of their claims. We also present more computationally efficient methods, and use these as a spring board to refute the existence of any practical efficiency improvements coming from this methodology.

1 Introduction

"Chebyshev said it,
And I'll say it again,
There's always a prime
Between n and $2n$..."

— Mathematician's Nursery Rhyme

The primes are fundamental building blocks of mathematics and cryptography, and have been in active study since Eratosthenes. Their quiddity is renowned, and yet we have many useful, intriguing, and solid results spanning the two thousand years that have lead to modern mathematics.

Grant in [2] states that they have made significant advances in checking primes for the cracking of cryptographic keys. In this paper we argue that there are more general results in the background mathematics that guarantees this work. We prove the general case of the modulus method given by [2], and offer potentially better methods to efficiently generate primes based off some deficiencies in the previous work as presented by Grant et al.

In section 3, we present some proofs of well-known results for interest and convenience. In section 4 we present a construction that is designed to elucidate that the methods from [2] are minimal, when considered in the case of moduli that are powers of two.

We then go on to show that these methods do not scale to larger moduli for highly composite numbers, as the sets in question are bounded below by the unbounded asymptotic value for $\pi(n)$, given by the Prime Number Theorem, in section 5.

Although we will use more formal presentation and language, we consider that the content of this paper is fairly elementary. The required mathematical background is broadly covered by most pre-university courses that we know of.

2 Previous Work

In [2] we find methods around the "digital roots" of prime numbers, in an attempt to optimize the prime search when it comes to breaking cryptographic keys. Their method revolves (literally) about the icositetragon, allowing us to map the primes onto the corners of this 24-sided regular polygon. It is well stud-

ied in online forums the fact that if for prime p,

$$x \equiv p \mod 24$$

then x must be one of

$$\{1, 5, 7, 11, 13, 17, 19, 23\}$$

as is documented on MathExchange at [4].

However, we find no mention whatsoever of other work in related studies on the primes and a given modulus, including [3] which explicitly studies primes based off congruence $\mod 24$. It is the author's opinion, however, that these are not immediately relevant in light of the work we present in this paper.

We also disagree with the claim in [2] that the icositetragon is a "unique polygon pertaining to prime numbers and their ultimate incidence and distribution" and provide a proof.

However, in lieu of immediately available resources that generalise the ideas and properties cited in [2], we present some straightforward proofs and results that lead naturally into a prime generation method we could not find in the literature so far.

2.1 Omitted Points of Discussion

We will omit the discussion of various rather trivial arithmetical errors.

For example, [2, p.5] concerns the following 'digital root' function $D(n)$, defined as the recursive sum of the digits of n in base 10 until one digit remains. So,

$$D(589) = 5 + 8 + 9 = 22 \to D(22) = 2 + 2 = 4$$

so $D(589) = 4$ when all is done. They claim the following distributive laws hold, for all $a, b \in \mathbb{Q}$ with $a + b = c$ and $a \times b = d$:

- $D(a + b) = D(a) + D(b) = D(c)$
- $D(a \times b) = D(a) \times D(b) = D(d)$

Whilst $D(a + b) = D(c)$ and $D(a \times b) = D(d)$ are both fine, we can see that the distributive nature of the laws are false:

$$\begin{aligned} \text{Let } a = 7, b = 4 \text{ then } D(a) = 7, D(b) = 4 \\ D(a + b) = D(c) = D(11) = 2 \neq 7 + 4 \\ D(a \times b) = D(d) = D(28) = 1 \neq 7 \times 4 \\ (\text{via } D(28) = 10 \to D(10) = 1) \end{aligned}$$

As such, clearly these distributive laws are at best badly written down, and at worst simply wrong. Whilst such an error belies a lack of care in their presentation, the distributive non-laws are not utilized in the original paper.

We also note that there is no formal definition of a 'quasi-prime' number that is in line with the literature. We take the definition from [1]:

Definition 2.1. *A* quasi-prime *n is an integer without small prime factors, where the prime factors of n must be greater than some $\mathcal{P}(n)$, a function that grows more slowly than n.*

Such a function can be given by:

$$\mathcal{P}(n) = n^{1/(\ln \ln n)^2}$$

as given in [1]. Again, we do not linger on such an error as it is not integral to the arguments presented in this paper.

3 Modular Arithmetic and Primes

We will discuss some interesting but fairly elementary points regarding the interaction of primes with modular arithmetic.

Definition 3.1. *For $a, b \in \mathbb{N}$, let the following notations be given:*

- *Let $(\exists^{\infty} x)$ denote "there exist infinitely many x such that..."*
- *$gcd(a, b)$ denote the greatest common divisor between a and b.*

We shall use that a and b are co-prime iff $gcd(a, b) = 1$, with

$$(\forall x \in \mathbb{N})(\forall \text{ prime } p) gcd(x, p) = 1 \text{ or } p$$

We state Dirichlet's Theorem, first proved in 1837, as found in [5]:

Theorem 3.2 (Dirichlet, 1837)**.** *For any $a, b \in \mathbb{N}$ such that*

$$gcd(a, b) = 1$$

there exist infinitely many $n \in \mathbb{N}$, such that the arithmetic progression given by

$$(an + b)$$

is prime.

3.1 A General Theorem about Primes and Moduli

This stalwart result gives us an easy theorem regarding modular arithmetic and primes:

Theorem 3.3. *Let*

$$A_m = \{a : 1 < a < m \ \& \ gcd(a, m) = 1\}$$

and let $t_m > 1$ be the first number such that $gcd(t_m, m) = 1$. For any $m \in \mathbb{N}$, and each prime $p \geq t$

$$(\exists a \in A_m)\ p \equiv a \mod m$$

Proof. Fix $m \in \mathbb{N}$, and let A_m be given. By basic number theory, we require that for any prime $p \geq t_m$

$$p \equiv x \mod m \rightarrow gcd(x, m) = 1$$

. To see this, suppose for contradiction that there exists $n > 1$ such that $gcd(x, m) = n$, with

$$p \equiv x \mod m$$

for some prime p. Then, there exists a such that $am + x = p$, and given n is a factor of each part of the left-hand side of the sum, there exists b such that $bn = p$, so p is not prime. $\rightarrow\leftarrow$[1]

Next, we notice that

$$(\forall a \in A_m)(\exists^{\infty} \text{ prime } p \geq t)\ a \equiv p \mod m$$

follows directly from theorem 3.2. Thus, for any $a < m$ such that $gcd(a, m) = 1$, the progression given for all $n \in \mathbb{N}$ by $nm + a$ is prime infinitely often.

Thus, it suffices to finally show that the primes are only found along such a given a modulus m, which are all contained in A_m. This fact follows by the pigeonhole principle, as all of the primes $p \geq t_m \mod m$ must appear for each of the $a \in A_m$, given each a 'finds' infinitely many primes p. The theorem follows from these arguments, as we have that it is necessary that for any prime p, $(p \mod m)$ is contained in A_m. □

Note that for prime m the result is immediate and trivial, as for every $n < m$, if m is prime, then $gcd(n, m) = 1$. Our minimum threshold t_m is given as any prime below t_m will be a factor of m.

4 New Methods and Limits in Prime Generation

We next utilize theorem 3.3 to improve efficiency in a similar way found in [2] - we will make use of some facts about modular arithmetic with modulus 2^m, and then show how this method, as does each method, have ceilings of efficiency.

4.1 2^m Moduli as Binary Filters on Binary Strings

Binary representations of natural numbers are the bedrock on which all modern computations acts. We propose some nice equivalences between binary moduli and actions on binary strings with a view to applying such 'binary filters' to generate binary stems which are infinitely prime. The author does not propose to be the first to have noticed the below results, however these are presented in an as accessible way as possible.

For any number n, let n_2 be the base 2 binary representation of n. The *initial segment* of n_2, is some finite number of the least significant bits corresponding to the lowest powers of 2 making up n_2, which are the initial bits of n_2 going from the right.

The following holds:

Lemma 4.1. *For all $m \in \mathbb{N}$, let $n > m$, then*

$$n \mod 2^m$$

is equivalent to the bitwise AND-mask

$$n_2\ AND\ \underbrace{1111111\ldots1}_{m\text{-}many}$$

[1] We apologize if this proof is being rather explicit and basic, but we wish to present a full argument in this paper.

which we denote by $n_2 \restriction m$.

Proof. This holds for for $m = 1$, as this AND mask is equivalent to asking whether some number is odd or even, which is 0 or 1 mod 2.

Next, notice that for every successive m, 2^m enumerates successive powers of two, which are the successively added binary bits to any representation n_2 - as such, $n_2 \mod 2^{m+1}$ is the same as asking if the next power of two is in the sum of powers of two giving n, which is the same as asking if the $(m+1)^{\text{th}}$ bit is 0 or 1 in n_2.

Next we notice that the given AND mask is just the 'cut' of some n_2 to the first m bits left of the LSB, and our argument follows inductively from the above. □

Thus, we have a 'binary filter' for the stems of the base-2 representations of natural numbers. We can now construct sets of these stems by lemma 4.1, as follows. Let

$$B_m = \{b_2 : 1 < b < m \ \& \ gcd(b, 2^m) = 1\}$$

Theorem 4.2. *B_m is the set of initial segments of the binary representations of all primes $p \geq t$, ($p \mod 2^m$).*

Proof. Follows immediately from theorem 3.3 and lemma 4.1 - noting that t is given by theorem 3.3. □

Let $a_2 \widehat{} b_2$ denote the concatenation of the binary representations of a and b. The following corollary is given by theorem 4.2 and theorem 3.2:

Corollary 4.3. *for all $n \in \mathbb{N}$ and for all $b \in B_m$ for some m,*

$$n_2 \widehat{} b$$

is prime infinitely often.

4.2 Proposed Improvements for Prime Generation Methods are Minimal

Take the case above, with $m = 8$. We can thus take the 54 primes less than 256 as given, and generate candidates going forward by means of permuting $1, 2, 3, \ldots$ in binary, concatenated with candidates from B_8. We then check if some candidate c has some integer n for which $c^2 = 24n + 1$, and if yes, pass this into the full primality check. The proof that for every prime p, p^2 is of the form $24n + 1$ for some n can be found in the appendix.

This method cuts out the least 2^8 possibilities of search for each bit added to the MSB of some initial segment in B_8. This scales for each subsequent $(m + 1)$ for each candidate generated from an initial segment in B_m, as for each bit added, we restrict the least significant m-many bits to the contents of B_{m+1}. Thus, for arbitrarily large m, B_m will give 2^{m-1} stems in B_m.

Thus, this method is simply the following: **"skip the even numbers"** - given we have moduli that have prime factors of only two. We have literally just found a fancier, more roundabout way of stating this.

5 Discussion of Results and Limits of These Methods

We will discuss some general properties and limits of these kinds of approaches.

5.1 Discussion of Our Methods

Our approaches outlined above may, to some, seem counter-intuitive - particularly theorem 4.2. However, we have similar results for base-10 representations - no prime will end in $2, 4, 6$, or 8 else it will be divisible by 2, and no prime will end in 0 or 5 else be divisible by 5, so all base-10 primes must end $1, 3, 7$, or 9.

However, just as we found that the B_m will be bounded below by 2^{m-1}, which is not optimal, we can give similar lower bounds in general, as we shall see in Proposition 5.2.

We have not yet carried out any further formal analysis of any actual speed-up of this generation method when it comes to breaking cryptographic key material. We expect it to be minimal, as it currently is for [2], but more than a cursory analysis is beyond the scope of this paper.

5.2 Grant's Fallacy

Suppose we see each of the $a \in A_m$ as 'search opportunities' for finding primes in order to proceed and attack some cryptosystem. Grant's paper demonstrates how to take the 24 opportunities (mod 24) and reduce them down to 8 opportunities, which is a significant reduction in the search space, by two thirds.

We wish to note that these are still CPU expensive arithmetical operations - checking if something is prime or not is a computationally expensive thing to do.[2] However, less expensive options do not add any particular efficiency.

With our stated results we can instead consider the use of other highly composite numbers - numbers that are more divisible than their predecessors - to generate primes, and that are greater than 24. For a modulus of 60, a larger highly composite number, we would get 16 search opportunities for all primes $p \geq 7$. Whilst this is more than the 8 in Grant, this is a reduction overall of the search space by just over 73.33%. For a larger highly composite number 1260, there are 288 numbers $n < 1260$ such that $gcd(n, 1260) = 1$, which is $\sim 22.86\%$ of the numbers. We may fool ourselves into thinking that this trend will be very useful for prime number generation, but this is not the case.

During the preparation of this paper, we have, however, noted some rather interesting behaviours of the numbers with GCD of 1 below some highly composite number. This is cause for further mathematical investigation.

Despite the fact that the ration of the number of coprime numbers below any n will approach zero, this will not happen regularly enough to be practical. A such, we set out to disprove the effectiveness of this approach by means of the prime number theorem, generally stated as follows:

Theorem 5.1. *Let $\pi(n)$ be the usual* prime counting function *such that $\pi(n)$ is just the count of the number of primes below n.*

$$\lim_{x\to\infty} \frac{\pi(x)}{\left\lceil \frac{x}{\ln x} \right\rceil} = 1$$

[2]Assuming $P \neq NP$, this is a very expensive operation.

The usual asymptotic reinterpretation of this is $\pi(x) \sim \frac{x}{\ln x}$ and is a well explored result in number theory.

So, taking the use of other highly composite numbers into account, we can demonstrate the following growth limit:

Proposition 5.2. *For all m, $|A_m|$ is $\Omega(\frac{m}{\ln m})$.*

Where Ω here is the asymptotic lower bound for the size of A_m, the lower bound analogue of the Bachmann-Landau Big-O notation.

Proof. The size of A_m is bounded below by the fact that we enumerate all $n < m$ that are coprime to m. Note for prime m, $|A_m| = m - 1$, so our best cases are highly composite numbers to be the lowest values for the size of A_m.

For any m it the case that A_m contains the all primes p below m and all composite $x < m$ with x coprime to m by definition. We know that the $\pi(m) \sim \frac{m}{\ln m}$ from the above, so even assuming there are no composite $x < m$ coprime to m, our lower limit must be $(m/\ln m)$. □

An interesting aside is that, because $|A_m| = m - 1$ for prime m, then $|A_m|$ is $O(m)$, giving a Big-Θ value of $\Theta(m-(m/\ln m))$. Additionally, it may be possible to use the Erdős-Kac theorem (see [6]) to get a better lower bound based on the probabilistic distribution of the primes.

Thus, our computation is bounded below by the distribution of actual prime numbers, and so is bounded below by an unbounded function. As such there is little effective difference between these optimizations and simply pre-computing all primes below a certain bit-length, as whilst there may be some m which gives significant reductions in the upper search space above m knowing information about coprime numbers below m, the efficiency does not scale upwards as m grows, owing to the lower bound given above.

APPENDIX - A Direct Proof for a Proposition in [2]

A proof of the following proposition is offered in [2]:

Proposition 5.3. *For any prime $p \geq 5$ there exists an n such that*

$$p^2 - 1 = 24n$$

Their proof is not convincing as a direct proof, given that a fact they make use of, that 'every prime is of the form $6k \pm 1$ for some k', is never proven. In fact, it is usually a corollary of this result, which we now prove.

Proof. For any given prime $p \geq 5$, consider the triple

$$(p-1), p, (p+1)$$

and note the following facts. Given each prime $p > 2$ is odd, 2 is a factor for both $(p-1)$ and $(p+1)$, and given these are 2 apart, 4 is a factor for one of them as well. Likewise for any triple such as this, one of them must also be a multiple of 3, and it cannot be p, so it must be one of $(p-1)$ or $(p+1)$.

Thus, we can conclude there exists some n such that

$$(p-1) \times (p+1) = 2 \times 3 \times 4 \times n = 24n$$

as we identified factors 2, 3, and 4 above. The theorem follows from the multiplying out of both equations like so

$$(p-1) \times (p+1) = p^2 - 1 = 24n$$

□

References

[1] B.M. Bredikhin. Quasi-prime number, 2019. http://www.encyclopediaofmath.org/index.php?title=Quasi-prime_number&oldid=34441 - Last visited on 6/8/2019.

[2] Robert E. Grant and Talal Ghannam. Accurate and infinite prime prediction from novel quasi-prime analytical methodology. *arXiv:1903.08570*, 2019.

[3] Jose Porras and William Caballero. An analysis of the congruence 1 mod 24 as a generator of prime numbers greater or equal to 5. *British Journal of Applied Science & Technology*, Article no.BJAST.24367:1–8, 02 2016.

[4] Eric Towers. Safe primes mod 24, 2014. https://bit.ly/2GdpupG - Last Accessed 11/7/2019.

[5] Eric W. Weisstein. Dirichlet's theorem. From MathWorld A Wolfram Web Resource, 2018. http://mathworld.wolfram.com/DirichletsTheorem.html - Last visited on 11/7/2019.

[6] Eric W. Weisstein. Erdös-kac theorem. From MathWorld A Wolfram Web Resource, 2018. mathworld.wolfram.com/Erdos-KacTheorem.html - Last visited on 31/7/2019.