Factoring and Primality Testing

Jodie Miu Margaret Dorsey

Spring 2016

Introduction

While the identification of primes and the factorization are clearly motivated by the computational needs of cryptography in modern times, the roots of current methods can be traced back as far as Eratosthenes' sieve, developed in the 3^{rd} century BCE. While it is difficult to say why mathematicians continued to study factorization, save for hunger for knowledge, the findings in these areas contributed immensely to other areas of both Number Theory and other fields of mathematics, particularly in allowing for work with computation using large numbers that would otherwise have been unfeasible in a pre-computing world.

This paper attempts to illustrate the progression from the very early, intuitive results and methods of Eratosthenes, Fermat, Euler, and Lehmer, among others, to the more sophisticated modern methods more commonly used today, while proving some of the more approachable results. The intention of the author is to build the theory of primality testing and factorization from elementary knowledge, so that a reader with a modest mathematical background may understand each method as it is introduced, with each building upon previous methods in a way that seems natural and intuitive.

Given the computational nature of the topic, some comments will be made on the computational viability of methods where appropriate.

1 History

Although the computational necessity for factorization and primality testing methods is patently obvious to us in the modern information age, the study and classification of primes and interest in the factorization of large numbers dates back to at least the Ancient Greeks. The notion of primes is at least as old as Euclid (circa 300 BCE), considering a definition of primality is contained in his *Elements* [7], and Eratosthenes' famous sieve, the earliest known method for algorithmic identification of primes and factorization of composites, was soon to follow in the 3^{rd} century BCE.

It is a fairly safe assumption that these first forays into primality testing and factorization were more motivated by curiosity than necessity - and indeed, it was not until the modern era that truly efficient and sophisticated methods were developed - but with the preservation of Greek mathematics by Arabic scholars, the theory continued to develop until the first deterministic primality test by trial division was outlined by Fibonacci in the early 13^{th} century. Further developments in the field of primality and factorization came with the study of perfect numbers by the Italian mathematician Cataldi, whose observations on the relationship between perfect numbers and primality were later made rigorous by Fermat, and the French priest Mersenne, who developed his famous Mersenne primes.

In the 17^{th} century came Fermat and his little theorem, which as we will see, forms the basis for a vast number of primality tests. This result, and more similar to it, arose from his study of Mersenne primes and their classification, which then developed into the discovery of so called Fermat numbers, the only numbers of the form $2^n + 1$ that can be prime. Fermat was also credited for the development of the difference of squares method of factoring, a much more sophisticated method than trial division or Eratosthenes' sieve.

With the 18^{th} century came Euler, and a veritable landslide of significant Number Theoretic results. Continuing from the work of Fermat, Euler provided many results, some covered in this paper, that further refined the theory of primality, factorization, and classification of primes. Soon after came Legendre, whose work in quadratic residues (along with the work of Euler and Gauss in the same field) allowed him to develop factorization methods based on quadratic sieves, which inform some of the methods still seen today.

Gauss, who had a deep interest in primality testing and factorization methods, also made significant strides in the field, both extending Legendre's sieve and contributing many results and methods of his own. After Gauss came the nineteenth and twentieth centuries and the work of Reuschle, Landry, Lucas, and Lehmer, all of whom significantly contributed to the more contemporary methods for factorization and primality testing, and with the advent of the computer age, the development of truly modern methodologies was not far behind.

It is unlikely that many of the mathematicians that built the foundations of primality testing and factorization methods had any idea that it would become such an important area of study in its own right, but the fact that their study was motivated by other problems and pursuits long before use in cryptographic systems and big data remains. Although many of the results discussed in this paper are stated without context, it is useful to remember that they were generally developed for a wide variety of purposes, very few of which align with their current applications, and that the sophisticated methods of today developed only after centuries of refinements made to the crude elementary techniques of trial division and straightforward sieves.

2 Background

In this section, we examine a few topics that will be necessary in the main paper of which the reader may be unaware.

Primitive Roots

Definition. *A number g is a primitive root modulo m if every number a coprime to n is congruent to some power of g modulo n. This k is called the index of a to the base g modulo n.*

The primitive residue classes modulo n have many useful properties, but we will not examine them very thoroughly - we only state without proof that n has a primitive root if it is of the form $2, 4, p^k,$ or $2p^k$ where p is an odd prime and $k \geq 1$, and if an integer n has a primitive root, then it has $\phi(\phi(n))$ of them. Additionally, the lowest power of a primitive root a modulo n that is equivalent to 1 modulo n is $\phi(n)$. An interested reader may find more on primitive roots and their theory in [9, Appendix 2]

We conclude with a small lemma that will be useful to us.

Lemma. *If k is the least positive integer such that $a^k \equiv 1 \pmod{n}$, $a^j \equiv 1 \pmod{n}$, and $k < j$, then $k \mid j$.*

Proof. Let k be the least power such that $a^k \equiv 1 \pmod{n}$. $j = k + m$, with $m \in \mathbb{N}$. if $k \nmid m$, then $m = qk + r$, $0 < r < m$, and $j = (q+1)k + r$. However, $a^j = a^{(q+1)k} a^r \equiv a^r \equiv 1 \pmod{n}$ contradicts the minimality of k. Thus k must divide j. □

Quadratic Residues

Quadratic Residue. *Let a and n be integers such that $(a, n) = 1$. If the congruence*

$$x^2 \equiv a \pmod{n}$$

has solutions x, then a is a quadratic residue of n. If there are no such solutions, then a is a quadratic non-residue modulo n.

It should be noted that the condition $(a, n) = 1$ does not prevent us from considering only the primitive residue classes modulo n when searching for quadratic residues.

We now have a result related to quadratic residues.

Theorem. *The product of two quadratic residues a and b modulo n is always a quadratic residue of n, and the product of two quadratic non-residues α and β modulo n is always a quadratic non-residue.*

Proof. Suppose that two integers a and a both relatively prime to n are quadratic residues modulo n. Thus $x^2 \equiv a \pmod{n}$ and $y^2 \equiv b \pmod{n}$ for some x and y. Because a and b are relatively prime to n, we can say $x^2 y^2 = (xy)^2 \equiv ab \pmod{n}$, which gives us our first result.

Now suppose there is an x such that $x^2 \equiv \alpha \pmod{n}$, but no y such that $y^2 \equiv \beta \pmod{n}$, and there exists some z such that $z^2 \equiv \alpha\beta \pmod{n}$. This gives us $z^2 \equiv x^2 \beta \pmod{n}$, and thus $\left(\frac{z}{x}\right)^2 \equiv \beta \pmod{n}$, which contradicts our assumption that β is a non-quadratic residue. □

Note that we do not address the product of two non-residues - we will state without proof that for powers of an odd prime p (including p itself), 4, or 2 times a power of p, the product of two non-residues is a residue, and that the issue is more complex for other composite numbers. Thus we can treat the product of two non-residue primitive roots as a residue. A complete classification of the product of two non-residues is given in [9, Appendix 3].

Legendre's Symbol, Euler's Criterion, and Jacobi's Symbol

We define Legendre's Symbol $\left(\frac{a}{p}\right)$ as a symbol given the value 1 if a is a quadratic residue of p and the value -1 if a is a quadratic non-residue of p, where p is a prime.

It is, of course, possible to compute Legendre's Symbol directly by trying every congruence class of p, but luckily there is a more elegant way, given by Euler:

Euler's Criterion. *If $(a,p)=1$ and p is an odd prime, then*

$$\left(\frac{a}{p}\right) \equiv a^{p-1/2} \pmod p$$

Proof. The result can be broken into two results:

$$a^{p-1/2} \equiv 1 \pmod p \text{ if and only if } a \text{ is a quadratic residue of } p$$

and

$$a^{p-1/2} \equiv -1 \pmod p \text{ if and only if } a \text{ is a quadratic non-residue of } p$$

We begin with the second of these. Let a be a quadratic non-residue of a prime p, and b be some natural number less than p. $bx \equiv a$ has a unique solution $x = b^{-1}$, because $(b,p)=1$. However, $b \not\equiv b^{-1} \pmod p$, because otherwise $b^2 \equiv a \pmod p$ contradicts the assumption that a is a quadratic non-residue. Thus all natural numbers less than p can be paired into ${p-1}/2$ pairs (m,n) such that $mn \equiv a \pmod p$.

Multiplying these pairs together, we have a product of ${p-1}/2$ integers all equivalent to a modulo p, which will be equivalent to $a^{p-1/2}$ modulo p and equal to $(p-1)!$. By Wilson's Theorem, we also have $(p-1)! \equiv -1 \pmod p$, and so when a is a quadratic non-residue of p, $a^{p-1/2} \equiv -1 \pmod p$.

Now let a be such that $a^{p-1/2} \equiv -1 \pmod p$. By Wilson's Theorem, $a^{p-1/2} \equiv (p-1)! \pmod p$, and thus this implies that the product of all natural numbers $b < p$ produce ${p-1}/2$ factors equivalent to a. If any b exists such that $bx \equiv a \pmod p$ has its unique solution x equivalent to b, then b will not be able to form a product equivalent to a when paired with any other factor of $(p-1)!$, which is a contradiction. Thus there is no b such that $b^2 \equiv a \pmod p$, and a is a quadratic non-residue.

Now let a be a quadratic residue of p. We can pick a natural number $b < p$ such that $b^2 \equiv a \pmod{p}$, and thus $b^{p-1} \equiv a^{p-1/2} \pmod{p}$. By Fermat's Theorem, we have $a^{p-1/2} \equiv 1 \pmod{p}$.

Finally, we examine the case that $a^{p-1/2} \equiv 1 \pmod{p}$ for some arbitrary a relatively prime to p. Let b be a primitive root modulo p such that a can be written as b^j for some j. From this we have $b^{j \cdot p-1/2} \equiv 1 \pmod{p}$. Because the least power of b that is equivalent to 1 modulo p is $p-1$, $p-1 \mid j \cdot p-1/2$. Thus j must be even, and $j/2$ is an integer. Thus we have that $(b^{j/2})^2 \equiv a \pmod{p}$, and thus a must be a quadratic residue.

□

You may have noticed that the Legendre Symbol is only defined for a prime p. We define an analogue for composites, *Jacobi's Symbol*, as

$$\left(\frac{a}{n}\right) = \prod_i \left(\frac{a}{p_i}\right)^{\alpha_i}$$

for n an odd integer, $(a, n) = 1$, and $n = \prod p_i^{\alpha_i}$.

From our earlier classification of the products of quadratic residues and quadratic non-residues, it can be concluded that if n has a primitive root, then

$$\left(\frac{a}{n}\right)\left(\frac{b}{n}\right) = \left(\frac{ab}{n}\right)$$

Recalling that we never addressed the case of the product of two quadratic non residues of other composite numbers, it makes intuitive sense that the Jacobi symbol occasionally incorrectly takes the value 1 for a quadratic non-residue, given it is the product of two non-residues. However, it never incorrectly identifies a quadratic residue. Thus it can be used to quickly determine that something is not a quadratic residue modulo n, but cannot prove something as a quadratic residue mod n alone. More on the theory of quadratic residues can be found in nearly any elementary Number Theory textbook, including [8].

3 Primality Testing

First, we will treat the various methods of determining whether or not a number is prime, or attempting to show that a number is composite, which is generally a much more approachable problem. At first glance, it seems strange to have methods for primality testing that are distinct from factorization methods - after all, if you can obtain the prime factorization of a number, it is trivial to determine primality from that factorization. However, factorization methods are in general more computationally expensive and difficult than the methods we will examine in this section.

Primality Tests and Composite Tests

Any viable algorithmic method for testing primality straightforwardly is essentially a condition on a number n that, when met, necessitates that n is prime. Thus if the condition is not met, n is composite. Such tests are certainly extremely convenient to directly determine the primality of n, but unfortunately methods of this form are usually rather complex and usually restricted to n either of a particular form or within some bounded range. We call these tests *primality tests.*

In addition to primality tests, which never incorrectly indicate primality, we also have many tests that are comparatively simple and computationally efficent, which occasionally fail to identify primes as prime, but never indicate that a composite number is a prime. We will call these tests, which never incorrectly indicate compositeness *compositeness tests.*

It is of the utmost importance to note before moving on that while the conditions of a primality or compositeness test being met guarantees that the number is either prime or composite respectively, and a failed primality test proves compositeness, a failed compositeness test does *not* necessitate that the number is prime.

The Sieve of Eratosthenes

Sieving, in general, is a process where a series of operations are applied to every number in a large, regularly spaced set of integers in order to find numbers with certain characteristics.

The Sieve of Eratosthenes is considered the first algorithmic method developed for factorization and primality testing. As such, it is admittedly very crude, but the theoretical basis of the method gives us some perspective on the properties and distribution of prime numbers, and of the fundamental ideas upon which further primality tests can be built. Because the Sieve of Eratosthenes is more generally suitable for algorithmic factorization of a number, we will treat it rather lightly here and revisit it in more detail when we examine factorization methods.

The applicability of Eratosthenes' sieve to primality testing stems from the idea of *trial division*, where given an integer n, we attempt to divide n by every integer that could possibly be a factor. If the number divides n, then it is a factor of n, and because possible factors must be bounded above by n itself, we are guaranteed a finite number of computations. In fact, we need only test factors less than $\sqrt{n}$, recovering greater factors as the quotient of a successful division by a lesser factor. Additionally, after each successful division, only the quotient need be tested further, resulting in easier computation as more factors are found (although for our purposes we have no need to ever find more than one factor). None of these facts are particularly reassuring computationally, but they provide some background for the use of our sieve.

The sieve can be thought of as a list or array of $N-1$ consecutive integers, beginning with 2. We recognize that 2 is prime, and thus every multiple of 2 is composite. We then remove 4,6, etc. from our list, and are left with 2 and the odd integers. We then recognize 3 as prime, because it is still in the list, and remove all multiples of 3 in the same way. 5 has not been removed from the list, and so it is prime. We then repeat our method. Continuing this way until the end of the set has been reached, we have constructed a set of all prime numbers up to N. It is intuitive that slight modifications to this process can provide the least prime factors, or even the entire factorization, of all numbers up to N as well.

It is a fair observation that this is not a test that can be applied to a single number, but rather construction of a set that will prove a number n prime if n is an element of the set, and composite if $n < N$ and n is not in the set. This is not ideal, but combined with our method of trial division, and given that 76% of odd integers have a prime factor less than 100 [9], precompiling a list of primes up to a certain bound and performing trial divisions of only those prime numbers on a given n can prove many numbers composite without having to resort to more rigorous compositeness or primality tests.

Fermat's Theorem and Resulting Compositeness Tests

Among the many limitations of Eratosthenes' sieve is its lack of precision - it is impossible to simply prove a given number n prime or composite, you must construct the entire set of primes up to n in the worst case to say anything about n at all. To remedy this problem, we move on to another elementary method that can test n directly, a compositeness test based on a theorem of Fermat.

Fermat's Theorem

Fermat's Theorem. *If a positive integer p is a prime and $(a, p) = 1$, then*

$$a^{p-1} \equiv 1 \pmod{p}$$

Fermat's theorem cannot be used as a primality test, as we will examine further when we define pseudoprimes, but we can use the converse of the theorem to develop a test for compositeness.

Converse of Fermat's Theorem. *If n and a are positive integers such that $(a, n) = 1$ and*

$$a^{n-1} \not\equiv 1 \pmod{n},$$

then n cannot be prime. Hence n is composite.

This simple compositeness test will prove very useful to us, given its theoretical and computational simplicity and the relative ease of multiplication of integers modulo n. Of course, one of the foremost weaknesses of Fermat compositeness tests is immediately apparent - the need for an appropriate a. As we will see in the very next section, not every a satisfying the conditions will prove a composite n composite. For now, we will assume that we simply iterate through all a relatively prime to n, searching for an a that proves n composite. This is, of course, not a bounded process, and so in practice we would establish an upper bound for a which would result in an inconclusive test if reached before a suitable a is found.

Pseudoprimes and Carmichael Numbers

To expand upon our problem with the infinitude of possible a values, there are composite numbers that will behave similarly to primes when Fermat's theorem is applied to them for certain a. We will call these composites *Fermat pseudoprimes base a.*

Definition. *A Fermat pseudoprime base a is an odd composite number m for which*

$$a^{m-1} \equiv 1 \pmod{m} \textit{ holds.}$$

In general, a *pseudoprime* is a composite number that behaves like a prime in the context of a compositeness test (we will see other pseudoprimes very soon.) Interestingly, if all Fermat pseudoprimes could be identified for a fixed base a, we could then use Fermat's Theorem with only base a as a primality test. Unfortunately, this cannot be generally accomplished, but in computational applications this idea is exploited for fast primality testing of n of a fixed length by classifying all Fermat pseudoprimes base a up to some upper bound $N > n$, applying Fermat's test to n, and checking to see if it is among the pseudoprimes if it is not proven composite. This rather practical sidestep of the limitations of a compositeness test is extremely useful for comparatively small n.

The existence of Fermat pseudoprimes prompts a question - can all Fermat pseudoprimes be proven composite eventually, given enough bases a are tested? The answer, unfortunately, is no. There exist numbers that will be Fermat pseudoprimes base a for all valid a, and these *Carmichael numbers* are explicitly defined as follows:

Definition. *A composite number n such that $a^{n-1} \equiv 1 \pmod{n}$ for all a relatively prime to n is called a Carmichael Number.*

A Carmichael number, the smallest of which is 561 [9], will never be revealed as composite by a Fermat's Theorem test. The existence of Carmichael numbers are a serious hindrance to the Fermat compositeness test, although we have some luck in they can be characterized further than their definition.

Theorem. *A positive integer n is a Carmichael number if and only if $p - 1$ divides $n - 1$ for every prime factor of n, n is composite, and n is squarefree.*

Proof. First, suppose that n is a composite number that square free, and $p-1 \mid n-1$ for every prime factor p of n, and that a is an arbitrary integer relatively prime to n.

By hypothesis, n is square free, so $n = p_1 p_2 p_3 \ldots p_k$, with p_i all distinct primes. Because a is relatively prime to n, a is also relatively prime to p_i for all $i \in 1, \ldots, k$. Thus, $a^{p-1} \equiv 1 \pmod{p_i}$ for all such i by Fermat's Theorem.

However, because $p - 1 \mid n - 1$, $a^{n-1} = (a^{p-1})^t \equiv (1)^t = 1 \pmod{p_i}$ for some $t \in \mathbb{Z}$. Because $p_i \mid a^{n-1} - 1$ for all i, and all p_i are distinct primes, $n = p_1 p_2 p_3 \ldots p_k \mid a^{n-1} - 1$. Thus $a^{n-1} \equiv 1 \pmod{n}$ for any arbitrary a relatively prime to n, and n is a Carmichael number.

We now suppose that a composite n is a Carmichael number, so $a^{n-1} \equiv 1 \pmod{n}$ for all a relatively prime to n. $n = p_1^{\alpha_1} \cdot \ldots p_k^{\alpha_k}$, and so $p_i^{\alpha_i} \mid a^{n-1} - 1$ and $(a, p_i) = 1$ for all $i \in \{1, \ldots, k\}$. Let r be an integer such that $n = p_i^{\alpha_i} r$. Obviously, $p_i^{\alpha_i}$ and r are relatively prime, so by the Chinese Remainder Theorem we have a unique solution of the system

$$x \equiv a_1 \pmod{p_i^{\alpha_i}}$$

$$x \equiv 1 \pmod{r}$$

where a_1 is a primitive root of $p_i^{\alpha_i}$ (Recall from our brief introduction of primitive roots that $p_i^{\alpha_i}$ is guaranteed $\phi(\phi(p_i^{\alpha_i}))$ of these).

Suppose that $\alpha_i = 1$. x is relatively prime to both p_i and r, so we have $x \equiv a \pmod{p_i \cdot r = n}$, which gives us $x^{n-1} \equiv 1 \pmod{n}$ and $x^{n-1} \equiv 1 \pmod{p_i}$ as well. However, $x^{p_i - 1} \equiv 1 \pmod{p_i}$, and $p_i - 1$ is the least power of x such that x is equivalent to 1 modulo p_i, and have $p_i - 1 \mid n - 1$ for all i.

Now suppose that $\alpha_i > 1$ for some i, so that n is not squarefree. $x^{n-1} \equiv 1 \pmod{p_i^{\alpha_i}}$, and by Euler's theorem, $x^{\phi(p_i^{\alpha_i})} = x^{p_i^{\alpha_i - 1}(p_i - 1))} \equiv 1 \pmod{p_i^{\alpha_i}}$. Thus because $p_i^{\alpha_i - 1}(p_i - 1)$ is the least power of a_1 equivalent to 1 modulo pi^{α_i}, $p_i^{\alpha_i - 1}(p_i - 1) \mid n - 1$. However, $p_i^{\alpha_i - 1} \mid n - 1$ is a contradiction, because $(n, n-1) = 1$ and $p_i^{\alpha_i - 1} \mid n$. Thus n must also be squarefree, and we are done. □

Despite this nice result, it is very troubling to have composite numbers that we could not identify even with infinite computational time, and because there are infinitely many Carmichael numbers, they cannot even be completely identified and compensated for. (A rather formidable proof of the infinitude of Carmichael numbers can be found in [1]).

Improving the Fermat test with Euler's Criterion

There is clear motivation to improve our method, if possible, to avoid the significant problem presented by the existence of Carmichael numbers. One tool that we have at our disposal is the *Euler Criterion for Quadratic Residues*, which is is stated below and proven in the Background section on quadratic residues.

Euler's Criterion. *If p is an odd prime and $(a, p) = 1$,*

$$a^{n-1/2} \equiv \pm 1 \pmod{p}$$

where the sign of 1 is chosen to match the value of Legendre's symbol $\left(\frac{a}{p}\right)$.

This criterion's logical converse, much like Fermat's theorem, gives us a test for compositeness.

Euler's Criterion as a Compositeness Test. *If n is odd, $(a, n) = 1$, and*

$$a^{n-1/2} \not\equiv \pm 1 \pmod{n}$$

then n is a composite number.

To strengthen this compositeness test, we use Jacobi's symbol over Legendre's symbol, remembering that Jacobi's symbol sometimes misidentifies quadratic non-residues as residues when n is composite.

Euler's Criterion as a Compositeness Test. *If n is odd, $(a, n) = 1$, and*

$$a^{n-1/2} \not\equiv \pm 1 \pmod{n}$$

then n is a composite number.

If $a^{n-1/2} \equiv \pm 1 \pmod{n}$ and

$$a^{n-1/2} \not\equiv \left(\frac{a}{n}\right) \pmod{n}$$

then n is composite.

If $a^{n-1/2} \equiv \left(\frac{a}{n}\right) \pmod{n}$ holds, the test is inconclusive.

As the last line of our definition and our knowledge of Jacobi's symbol might suggest, this compositness test also has pseudoprimes. We will call these *Euler pseudoprimes*.

Euler Pseudoprimes and Strong Pseudoprimes

We define an Euler pseudoprime in an analogous way to our definition of a Fermat pseudoprime.

Definition. *Let n be an odd composite number such that*

$$a^{n-1/2} \not\equiv \left(\frac{a}{n}\right) \pmod{n}$$

for some a relatively prime to n. Then n is an Euler pseudoprime base a.

Unfortunately, there are some numbers that are both Carmichael numbers and Euler pseudoprimes. An example of one such number is 1729.

Despite this disappointing fact, many Carmichael numbers are revealed as composite by Euler's Criterion, and additionally, there is no analogue for Carmichael numbers in Euler pseudoprimes. If enough bases a are tested, we will eventually prove the compositeness of any composite number.

To close this section, we present a final type of pseudoprime, which, like Euler pseudoprimes, can always be eventually proven composite by some base a.

Definition. *An odd composite number n with $n - 1 = d \cdot 2^s$, d odd, is called a strong pseudoprime for base a if either $a^d \equiv 1 \pmod{n}$ or $a^{d \cdot 2r} \equiv -1 \pmod{n}$, for some $r \in \{0, 1, \ldots, s-1\}$.*

This test for pseudoprimes is intended, much like Euler pseudoprimes, to eliminate the problem of Carmichael numbers in the Fermat Compositeness test. Indeed, any Fermat pseudoprime will be eventually proven composite by the strong pseudoprime test if enough bases a are tested.

The motivation (which is not a proof) for the test is as follows:
Any Fermat pseudoprime n to base a will satisfy the equivalence

$$a^{n-1} - 1 \equiv 0 \pmod{n}$$

Because we assume that n is odd if it is being tested for primality, $n = 2m + 1$ for some integer m and we have

$$a^{2m} - 1 = (a^m - 1)(a^m + 1) \equiv 0 \pmod{n}$$

If n is a prime, it must divide one of these factors, but it cannot divide both because then it would need to divide all linear combinations of them, including $(a^m - 1) - (a^m + 1) = -2$. Therefore we have

$$a^m \equiv \pm 1 \pmod{n}$$

We can write n as $2^\alpha k + 1$, where k is odd, and have

$$a^{n-1} - 1 = (a^k - 1)(a^k + 1)(a^{2k} + 1) \ldots (a^{2^{\alpha-1}k} + 1)$$

If n divides exactly one of these factors but is composite, then it is a strong pseudoprime. Interestingly, if a number is a strong pseudoprime to the base a, it will also be a Euler pseudoprime to a.

Elementary Primality Tests

In the previous section, all the tests we examined could only possibly prove compositeness - because we are unable to test infinitely many bases in an Euler or Strong Pseudoprime test, there is no way for us to computationally prove that a number is prime using these methods, only to say that it is somewhat likely to be prime, or say with certainty that it is composite.

In this section, we examine some methods of proving primality. These primality tests are much more complicated computationally and theoretically than compositeness tests, and unfortunately usually depend on factorization of a number, which is a slow and laborious process when performed, and very limiting to the form of n when avoided. However, they are theoretically motivating, despite not being as practically useful as more modern primality tests.

Lehmer's Theorem

The first method of actually proving primality that we will look at is based on a theorem by Lucas, proven by Lehmer.

Lehmer's Theorem. *Suppose $n-1=\prod_{j=1}^{n} q_j^{\beta_j}$, with all q_j distinct primes. If an integer a exists such that*

$$a^{n-1/q_j} \not\equiv 1 \pmod{n} \text{ for all } j=1,\ldots,n$$

and such that

$$a^{n-1} \equiv 1 \pmod{n},$$

then n is a prime number.

We note that this theorem is an extension of Fermat's theorem.

Proof. Consider $a^k \equiv 1 \pmod{n}$. The smallest such k divides all possible k, including $n-1$. However, every divisor k not equal to $n-1$ of $n-1$ divides at least one $\frac{n-1}{q_j}$. Thus if k is less than $n-1$, $a^{n-1/q_j} \equiv a^{tk} \equiv 1 \pmod{n}$ will hold for some q_j. This contradicts the assumptions of the theorem, and so $k=n-1$. However, by Euler's theorem, we know that $a^{\phi(n)} \equiv 1 \pmod{n}$ for all a relatively prime to n, and that $\phi(n) < n-1$ for all composite numbers. If $\phi(n) < k$, it contradicts the assumption that k is the smallest power of a equivalent to 1, and thus $\phi(n)=k=n-1$, and n must be prime. □

From our knowledge of primitive roots, we know that any primitive root will be a suitable a for Lehmer's Theorem. The problem inherent in this is that there is not an efficient, deterministic method to find primitive roots, or even quadratic non-residues. If n is prime, then we will have $\phi(\phi(n)) = \phi(n-1) = (n-1)\prod(1-\frac{1}{q_j})$ primitive roots, but as the equation shows, if $n-1$ has many prime factors, especially many smal primel factors, primitive roots become a smaller proportion of possible a values.

Fortunately, Selfridge provides us with a relaxed version of Lehmer's theorem such that we do not need a primitive root to prove n prime.

Theorem. *Suppose* $n-1=\prod_{j=1}^{n} q_j^{\beta_j}$, *with* q_j *all distinct primes. If for every* q_j, *there is an* a_j *such that*

$$a_j^{n-1/q_j} \not\equiv 1 \pmod{n}$$

while

$$a_j^{n-1} \equiv 1 \pmod{n},$$

then n *is a prime.*

The proof of this statement can be found in [2].

This allows us to perform tests with multiple bases, and prove primality usually much more quickly than Lehmer's theorem applied directly, especially for primes where the least primitive root is large.

However, there is still a major stumbling block in practical applications of Lehmer's Theorem - the need to factorize $n-1$. Luckily, it is possible to relax the conditions of Lehmer's theorem in a way that allows us to test with only a partial factorization.

Theorem. *Let* $n-1=r \cdot f$, *where* f *is the factorized part of* $n-1$ *and* r *is the unfactorized or remaining part such that* $(r,f)=1$, $r<f$, *and* $f=\prod_{j=1}^{n} q_j^{\beta_j}$, *with all* q_j *distinct primes.*

If an integer a *exists such that*

$$\gcd(a^{n-1/q_j}-1, N)=1 \textit{ for all } j,$$

and

$$a^{n-1} \equiv 1 \pmod{n},$$

then n *is a prime number.*

There is an analogous theorem used in factorization, the Lehmer-Pocklington Theorem, developed from this theorem and a result called Pocklington's theorem.

Pocklington's Theorem. *Let* $n-1=rq^n$, *where* q *is prime and* $q \nmid r$. *If there exists an integer* a *satisfying*

$$\gcd(a^{n-1/q}-1, n)=1 \text{ such that } a^{n-1} \equiv 1 \pmod{n},$$

then each prime factor p *of* n *has the form* $p=q^n m+1$.

Pocklington's theorem was developed for factorization methods - and as such is a bit of a detour from our primality tests - but in certain cases it can be used to prove, for instance, that all prime factors of n are greater than $\sqrt{n}$, thus proving n prime. In fact, we will see Pocklington's theorem again in the Elliptic Curve primality test.

Pépin's and Proth's Theorems

Another possible sidestep of the need to factor $n-1$ is available when n happens to be a Fermat number, or a number of the form $2^{2^r}+1$ (These happen to be the only sums of a power of 2 and 1 that can be prime). In this case, $n-1$ has the minimum possible number of distinct prime factors, and thus is very suited to a Lehmer's Theorem test. In fact, for a Fermat number n, the requirements of Lehmer's Theorem become

$$a^{n-1/2} = a^{2^{2^{r-1}}} \not\equiv 1 \pmod{n}$$

and

$$a^{n-1} = a^{2^{2^r}} \equiv 1 \pmod{n},$$

which essentially reduces to

$$x^2 \equiv 1 \pmod{n}$$

and

$$x \not\equiv 1 \pmod{n}$$

where $x = a^{2^{2^{r-1}}}$.

If n is prime, then $x^2 \equiv 1 \pmod{n}$ will have exactly two solutions, namely $x \equiv \pm 1 \pmod{n}$. $x \equiv 1 \pmod{n}$ violates the conditions of Lehmer's theorem, and so $x \equiv -1 \pmod{n}$ when n is prime. This motivates a need for an a such that

$$x \equiv a^{n-1/2} \equiv -1 \pmod{n}$$

Noting that Euler's Criterion gives us that a must be a quadratic non-residue of n, along with the fact (that we will state without proof) that 3 is a quadratic non-residue of all primes of the form $12n \pm 5$ and the observation that $2^{2^k} \equiv 4 \pmod{12}$ for all k inspires (but does not prove) Pépin's Theorem, which is stated below.

Pépin's Theorem. *A Fermat number $n = 2^{2^r}+1$, $n \geq 1$ is prime if and only if*

$$a^{n-1/2} \equiv -1 \pmod{n}$$

It is worth mentioning that the binary computer makes the division by n necessary to reduce powers of a modulo n very computationally simple, due to n being a power of 2 plus 1 and therefore very simply expressed in binary. This makes Pépin's test a very attractive test for Fermat number primes.

With both Pépin's Theorem and relaxed version of Lehmer's theorem, we obtain Proth's Theorem, stated below.

Proth's Theorem. *Suppose n is of the form $n = m \cdot 2^k + 1$, with $2^k > m$ and m odd. If there exists an integer a such that*

$$a^{n-1/2} \equiv -1 \pmod{n},$$

then n is prime.

A proof of both Pépin's and Proth's theorems can be found in [6].

There is yet another way to ease the difficulty of factoring $n-1$. By using a different compositeness test than Fermat's theorem that allows us to attempt to factorize a number other than $n-1$, we may end up with a number that proves easier to factor. For example, primality tests based on Lucas Sequences give an analogue to Lehmer's theorem in quadratic fields, where $N+1$ may be factored instead of $N-1$. The theory of quadratic fields is beyond the scope of this paper, and so we do not address Lucasian or similar methods, but their existence is worth noting and can be investigated further in [9] and [6].

Modern Primality Tests

Of the primality tests covered in the previous section, none are of practical use for all numbers. Computationally viable methods such as Pépin's Theorem only apply to primes of a certain form, and the need for factorization, even partial factorization, and identification of primitive roots cripples Lehmer's Theorem computationally. Fortunately, there exist more suitable methods used today in primality testing.

These tests are, in general, quite theoretically and algorithmically complicated, and so we give only a brief introduction to the ideas behind modern primality tests and do not attempt to rigorously present them.

The Jacobi Sum Primality Test

The foundation of the Jacobi Sum Primality test is the strong pseudoprime test we have seen and its use of cyclotomic number fields. Put in the simplest terms possible, the test uses information from a combination of compositeness tests based the strong pseudoprime test in cyclotomic rings. The results of these tests is then used to construct a sieve for the possible prime divisors of a number n until in the end it is proven to be its own sole prime divisor, or a prime number, or a prime divisor is found that proves n composite.

Also called the Adleman-Pomerance-Rumely primality test, the Jacobi Sum Primality test does not rely on randomness like many of the more efficient primality tests, allowing it to be a deterministic primality test in exchange for computational superiority. Interestingly enough, our strong pseudoprime test may also form a deterministic primality test, if the Extended Riemann Hypothesis is assumed to be true. It was this near miss that prompted its adaptation into what is now the Jacobi Sum test, which thankfully does not rely upon unproven results.

Elliptic Curve Primality Testing

Elliptic curve primality testing is based on the use of the properties of the group of points modulo n on an elliptic curve. Unlike the Jacobi Sum Test, Elliptic Curve testing is probablilistic, and therefore may theoretically fail. That isn't to say that it gives the wrong answer - that would make it not a primality test - but it is possible for the algorithm to run indefinitely, never proving even a prime n prime. Despite this, elliptic curve methods are among the fastest and most popular primality testing algorithms used today.

Stated very generally, elliptic curve methods use elliptic curves generated either by random integers a and b in the case of the Goldwasser-Killian algorithm or by methods guaranteed to generate curves that will be computationally simpler for the Atkin-Morain test. We then consider the equation $y^2 = x^3 + ax + b$ modulo n. If n is prime (and we may practically assume that it is, given that any n subjected to an Elliptic Curve test is likely to have passed all of the less expensive compositeness tests available), then the set of all integer solutions of this curve will have certain properties, which we will hope to verify and thus prove the primality of n.

The rest of the test depends on the fact that the only non-invertible equivalence class modulo a prime is 0, or the class containing multiples of the prime itself. The computations necessary for proving a number n prime using an elliptic curve cannot be performed on a non-invertible element modulo n, but if we end up encountering such an element outside of the multiples of n, we have shown that n cannot be prime and the test can conclude.

The integer points of our elliptic curve E are then manipulated and tested with an analogue of our earlier theorem by Pocklington. These tests will either prove n composite by finding a prime factor not equal to n, or allow us to prove n prime given that we can verify the primality of a computed probable prime factor of n q, because we will have chosen $q > \sqrt{n}$. Of course, the method can then be recursively applied to q, each time resulting in a smaller probable prime (and of course applying less computationally taxing compositeness tests to each new probable prime before returning to elliptic curves). With this recursion, we will either end up proving the compositeness of one and therefore all of our probable primes, or arriving at a well known prime and proving them all prime.

A more explicit and instructive treatment of elliptic curve methods can be found in [5].

4 Factorization

Prime factorization is the process of decomposing an integer into distinct prime powers. For relatively small integers, this is trivial to straightforwardly. As these integers scale, however, this task becomes prohibitively expensive to compute by even modern day algorithms. In fact, the security of the RSA cryptographic algorithm relies heavily on the relative computational difficulty of factoring large composite numbers. Before we examine some of the elementary methods of factorization, there are a few notes to be made on factorization in general.

In the previous section, we discovered that there are many ways to prove compositeness without too much computational expense. Because factoring large integers is so comparatively difficult, it is recommended to first confirm that the number we want to factor is indeed composite using the results of the previous section before even attempting factorization.

Another important consideration before attempting factorization is the form of the composite N. N of certain forms may lend themselves to certain specialized factorization methods that allow us to neatly avoid some of the difficulty of factorization, and identification of these N before trying more expensive algorithms can help us to alleviate the computational workload involved.

Classical Factorization Methods

Trial Division

As a factorization method, trial division is repeated division of our number N by small primes. We store the pre-computed primes and their number in some table. This would be very speedy, albeit at the cost of the storage. Alternatively, we could save space by generating our primes along the way, using the form $6k \pm 1$ (including 2 and 3).

Euclid's Algorithm

Interestingly enough, we can also use Euclid's algorithm to search for the factors of a number. Euclid's algorithm will give us the prime factors of our number N between g and G, where g is our lower search limit and G is our upper search limit.

To use Euclid's algorithm to factor a number, start by multiplying together all primes between the two limits. In fact, there even exists precomputed products for primes within certain ranges, to use here should we want to. Next, we apply Euclid's algorithm on the product of the primes and our number N. Prime factors of N within the search limits g and G will be found.

Application of Euclid's is fast because it simply is repeated division, multiplication, and subtraction. The only possible tricky thing here is the initial division of the (potentially) large product of primes by N. Aside from that, it is not too difficult to perform in step-wise fashion.

Although one may assume Euclid's is no longer used, it is not obsolete. In fact, it is used when N is too large to be easily divided by a small prime with the arithmetic computers provide. In that case, it is faster to divide the huge product of primes by N than to divide N over and over by small primes. It is also used when only small divisors are sought, like in the continued fraction method. To use this method to obtain small divisors, we can store products of small primes.

Fermat

Fermat's uses Legendre's congruence, which states that every number has the solutions $x \equiv \pm y \pmod{M}$ to $x^2 \equiv y^2 \pmod{M}$. These are the trivial solutions. If M is composite, however, then other solutions exist that can be used to factor M. Many other classical factorization methods also use Legendre's congruence to find the factorization of a composite number, but the way in which they differ is in how solutions to the congruence are discovered.

Here, the idea is that we can write an odd composite number as a difference between two nonconsecutive squares. Once we have that, it naturally gives us a factorization, in that: $N = ab = x^2 - y^2 = (x - y)(x + y)$. To find the two square numbers, we start by making the observation that any x that satisfies $N = x^2 - y^2$ must be $> \sqrt{N}$.

With that observation, we can start with $m = \lfloor\sqrt{N}\rfloor + 1$, which is the smallest x possible (aside from the case where N is x^2). Being a relatively simple factorization, we do not need to consider the case where N is x^2. We also want to discard trivial factorizations, like $N = ab = x^2 - 1^2$. To that end, we can set the restriction that the difference of our x, y must be greater than 1.

We consider $x = m^2 - N$, checking to see if it is a square. If it is, then we have found our x and y, y being m. If x is not a square, then we try the next possible x by incrementing our m by 1, and testing to see if that is a square. Repeat the process until our difference of squares is found. After the difference of squares is found, we can find our a, b by substituting our known x and y in.

For example, take the number 1625. If we wanted to write it as a difference between two nonconsecutive squares, we can start by confirming that 1625 is not a square. $\sqrt{N}$ is not an integer, so we can proceed:

$$m = \lfloor\sqrt{N}\rfloor + 1 = 41$$
$$x = m^2 - N = 41^2 - 1625 = 56$$

56 is not a square, and so we repeat the process until we arrive at $x = 45, y = 20$. Notice that our x, y are not primes. To find the prime factorization, we then apply Fermat's on x, y.

Note that the method is not very efficient. Fermat's method performs best in cases where N is a product of two nearby integers, because the factorization can then be found in a relatively small number of rounds of the process described above.

However, we can consider the shortcuts that have been developed since. One of them uses the observation that the last two digits of squares cannot be just any number. To use this fact, we can test only the numbers that have a possible square ending.

Another one of the shortcuts that has been developed is to use a multiplier. The idea behind this is that we can find a factorization for a number whose prime factors are relatively close quickly. Although the number N we are trying to factor may not be a product of two nearby integers, the number n multiplied by some other integer k may be. In that case, we could compute the greatest common denominator of n and kN to obtain the prime factorization. To find a multiplier k, we could use the Lehman method, which is a formalization of the discovery of these multipliers k given N.

Euler

Euler's method uses Legendre's congruence as well, but does not start with attempting to directly search for the difference of squares. In fact, it starts with the Lagrange identity, which states that:

$$(x^2 + Dy^2)(u^2 + Dv^2) = \begin{cases} (xu + Dyv)^2 + D(yu - xv)^2 \\ (xu - Dyv)^2 + D(yu + xv)^2 \end{cases}$$

In other words, the identity states that a product of two integers of the form $a^2 + Db^2$ is itself of that form, and also has two different representations in that form. Note that D has to remain the same throughout all representations. Because this method relies on this identity, this method is only applicable to integers that are of this form.

We can use the converse of this identity to find how to write N as a product of two numbers of the form. The converse states that if we can find two different representations of the form $a^2 + Db^2$, with the greatest common denominator of the product of the b values of each representation and N being 1, then N can be written as a product of two integers of that same form.

Proof. □

Euler's method starts with the congruence $a^2d^2 \equiv -Db^2d^2 \equiv b^2c^2 \pmod{N}$. From here, we can see that Legendre's congruence can be used here. Then, we know the factors of N will be $\gcd(N, ad - bc)$ and $\gcd(N, ad + bc)$.

Because this method first requires us to know that our N has two different representations of the form $a^2 + Db^2$, we can start by searching for one representation first, using Fermat's method to do this. If no representations can be found, we can stop here and use another method to factorize N, for Euler's method is not applicable here.

5 Conclusion

As the world becomes more and more technologically dependent, the field of cryptography will no doubt generate even more advanced methods than those we have today. However, as we have seen, the nature of the theory behind both primality testing and prime factorization is deeply iterative and interleaved - without the early results of the pre-computing age, we would not have our modern methods, and it is profoundly likely that the work of today and tomorrow will be the foundations for future results. It is the hope of the authors that through the elementary methods described in this paper, the fundamental theory of the treatment of primes and prime factorizations has been revealed, at least partially, and a foundation of understanding has been formed that makes the general logic and application of even more theoretically complex algorithms more accessible.

References

[1] W.R. Alford, Andrew Granville, and Carl Pomerance. There are infintely many Carmichael numbers. *Annals of Mathematics*, 140:703–722, 1994. URL `https://math.dartmouth.edu/ carlp/PDF/paper95.pdf`.

[2] John Brillhart, D.H. Lehmer, and J.L. Selfridge. Primality Criteria and Factorizations of $2^m \pm 1$. *Mathematics of Computation*, 29(130):620 − 647, 1975. URL `http://www.ams.org/journals/mcom/1975-29-130/S0025-5718-1975-0384673-1/`.

[3] Richard Crandall and Carl Pomerance. *Prime Numbers: A Computational Perspective.* Springer, 2 edition, 2005. ISBN 0-387-25282-7.

[4] Peter Giblin. *Primes and Programming: An Introduction to Number Theory with Computing.* Cambridge University Press, 2 edition, 1993. ISBN 0-521-40182-8.

[5] Neal Koblitz. *A Course in Number Theory and Cryptography.* Springer, 2 edition, 1994. ISBN 0-387-94293-9.

[6] Evangelos Kranakis. Primality Tests. *Technical Report*, 345, 1984. URL `http://cpsc.yale.edu/sites/default/files/files/tr345.pdf`.

[7] Richard A. Mollin. A Brief History of Factoring and Primality Testing Before Computers. *Mathematics Magazine*, 75(1):18–29, 2002. URL `http://www.jstor.org/stable/3219180`.

[8] Ivan Niven, Herbert Zuckerman, and Hugh Montgomery. *An Introduction to the Theory of Numbers.* John Wiley and Sons, 5 edition, 1991. ISBN 0-471-62546-9.

[9] Hans Riesel. *Prime Numbers and Computer Methods for Factorization.* Birkhauser Boston, 2 edition, 1994. ISBN 0-8176-3743-5.

About the authors:

This paper was written to fulfill the requirements of Math 371, Number Theory, offered by Dr. Anurag Agarwal in the Spring 2016 semester at Rochester Institute of Technology.

Margaret Dorsey

Computational Mathematics
Game Design and Development
Computer Science

med7068@rit.edu

Responsible for all proofs provided directly in the paper in both the Background and Primality sections.

Author of the Introduction, Background, Primality, and History sections.

Jodie Miu

Computer Science

jm7481@rit.edu

Author of the Factorization section. Responsible for the proof provided within that section.