GARRETT, STEPHANI LEE, M.A. On the Quadratic Sieve. (2008)
Directed by Dr. Paul Duvall. 34pp.

Factoring large integers has long been a subject that has interested mathematicians. And although this interest has been recently increased because of the large usage of cryptography, the thought of factoring integers that are hundreds of digits in length has always been appealing. However it was not until the 1980's that this even seemed fathomable; in fact in 1970 it was extremely difficult to factor a 20-digit number. Then in 1990 the Quadratic Sieve factored a record 116-digit number.

While the Quadratic Sieve is not the most recent development in factoring, it is more efficient for factoring numbers below 100-digits than the Number Field Sieve. This paper will discuss the methodology behind the Quadratic Sieve, beginning in its roots in Fermat and Kraitchik's factoring methods. Furthermore our objective is to fully describe the Quadratic Sieve with the goal that the reader could implement a reproduction of the sieve for small numbers.

ON THE QUADRATIC SIEVE

by

Stephani Lee Garrett

A Thesis Submitted to
the Faculty of The Graduate School at
The University of North Carolina at Greensboro
in Partial Fulfillment
of the Requirements for the Degree
Master of Arts

Greensboro
2008

Approved by

_____
Committee Chair

This thesis is dedicated to

My family, without whose support and inspiration
I would never have had the courage to follow my dreams.

APPROVAL PAGE

This thesis has been approved by the following committee of the Faculty of The Graduate School at The University of North Carolina at Greensboro.

Committee Chair_____

Committee Members_____

_____

_____
Date of Acceptance by Committee

_____
Date of Final Oral Examination

ACKNOWLEDGMENTS

I would especially like to thank Paul Duvall for his guidance and encouragement throughout the preparation of this thesis. I greatly appreciate the time and effort that Paul dedicated to helping me complete my Master of Arts degree, as well as his support and advice over the past two years.

I would also like to thank the thesis committee members for their time and efforts in reviewing this work.

TABLE OF CONTENTS

CHAPTER I
PRELIMINARIES

Mathematicians have long been concerned with factoring large numbers. Factoring has always been an important process in mathematical computations, but has become increasingly more important because of its role in modern cryptosystems. In fact, the security of some widely used cryptography systems is based on the difficulty surrounding the factorization of large numbers, mainly upwards of 100 digits. It is this fact that has mathematicians seeking the most efficient ways to factor these large integers.

Since the development of public key cryptography in the 1970's, the difficulty of factoring large numbers has granted security to companies and organizations alike. Although the infeasibility of factoring makes these systems secure, public key cryptosystems are not widely used for encrypting general information, but rather for the exchange of the key. This probably stems from the fact that it takes large amounts of both memory and time to encrypt and decrypt messages using public key cryptography on computers, much more than other methods such as the Data Encryption Standard (DES). As previously mentioned, public key cryptography is used to exchange keys for DES on other systems. Moreover, because the secrecy of the public key cryptosystem depends entirely on the difficulty of discovering the decryption key it is imperative that the numbers used require large amounts of time to factor [22].

Introduced in 1978 by Ronald Rivest, Adi Shamir and Leonard Aldeman, the most renowned public key cryptosystem is the RSA Cryptosystem. This cryp-

tosystem revolves around finding two large primes, call them p and q, which are ideally at least 100-digits in length. Then one would let $n = p \cdot q$. Additionally, let $m = (p-1) \cdot (q-1)$. Next the person establishing the key would find a number E such that $\gcd(E,\ m) = 1$, by choosing random integers E until the E described above is found. We then use the Euclidean Algorithm to find an integer D such that $DE \equiv 1 \pmod{m}$. Next the author of the key publishes n and E, keeping the rest secret [17]. Now, according to the RSA website, the current recommended length of this number m should be approximately 231 digits; if one plans to keep this particular m for long term usage the website recommends a number of approximately 308 digits [23]. Hence it is believed that for someone else to find the integer D, that person would have to factor n and there is no efficient, reasonable, way to do so [17].

Therefore many ways have been developed with the purpose of factoring larger numbers in the least possible amount of time; from Fermat's Difference of Squares to the Quadratic Sieve. Each method has developed upon the last, improving upon efficiency and offering different strategies that ultimately seem to become stepping stones for the next factorization method. Notably, the most practical factoring algorithms as of the late 1980's were the Quadratic Sieve and the Elliptic Curve Method. It is typical of methods such as the Quadratic Sieve that they involve a great deal of overhead in implementation, so that they only become practical for truly big numbers. In this report, we will focus on the Quadratic Sieve. It has been superseded by the more complicated Number Field Sieve, but it illustrates the main features of modern factor base methods, and is still the most efficient method for moderate sized numbers around one hundred digits in length.

CHAPTER II
DEFINITIONS

Before we can discuss the Quadratic Sieve and ideas that led to its development we must define some terminology.

A *residue class* is the set of integers that are congruent to some integer k modulo n. In other words, it is the set S such that for all $l \in S$, we have that $l \equiv k \pmod{n}$.

A number a is said to be a *quadratic residue mod* m if for coprime integers m, a with $m > 0$, the congruence

$$x^2 \equiv a \pmod{m} \tag{2.1}$$

has a solution. If equation (1.1) is unsolvable then a is said to be a *quadratic nonresidue* [6].

An *exponent vector* for a factored integer is an integer vector $v(m)$ such that each entry represents the exponent on the i^{th} prime, where the integer 2 is the first prime, 3 the second and so forth. For example if we have $m = 56 = 2^3 \cdot 7$ then $v(m) = (3, 0, 0, 1)$.

The *Legendre Symbol* is, for an odd prime p, defined as

$$\left(\frac{n}{p}\right) = \begin{cases} 0 & \text{if } n \equiv 0 \pmod{p} \\ 1 & \text{if } n \text{ is a quadratic residue } \pmod{p} \\ -1 & \text{if } n \text{ is a quadratic nonresidue } \pmod{p} \end{cases} \tag{2.2}$$

We will also be interested in working with numbers that have only small prime factors. So if B is a positive integer, we say that an integer m is *B-smooth* if its prime divisors are all less than or equal to B [19].

CHAPTER III
FACTORIZATION METHODS

As previously enunciated, factoring can be extremely lengthy, so mathematicians have developed a sort of order of operations for factoring large integers. Once presented with a number n, one begins the factoring process with trial division; usually we perform trial division on n by all primes $p < \log(n)$ [15]. Next one would try using the Pollard-Rho method and following that the Elliptic Curve Method, which while these will not be described in this paper can be found in *Prime Numbers* by Carl Pomerance. Then finally one would use some version of the Quadratic Sieve. We should mention that the reason that the Quadratic Sieve is reserved for the last is because of all these methods it is the most lengthy to initiate. But to understand the Quadratic Sieve we must briefly discuss its roots, which are embedded in previous factoring methods.

Many factorization methods currently exist such as that of the Continued Fraction Algorithm, the Miller-Western Algorithm, and Schroeppel's Linear Sieve, however most stem from Fermat's Difference of Squares Method and Kraitchik's Algorithm. But even Kraitchik's algorithm begins by altering that of Fermat.

3.1 Fermat's Difference of Squares Method

One of the more elementary methods for factoring is Fermat's Difference of Squares Method. Suppose we want to factor a large odd composite number n [19] and let a be the first perfect square larger than n. Now say $a = t^2$ for some $t \in \mathbb{Z}$, then we look at $a - n$ and see if it too is a perfect square, if not then we examine

$(t+1)^2 - n$ to see if it is a perfect square and so on until we find an integer k such that $k^2 - n = u^2$ for some $u \in \mathbb{Z}$. Subsequently, we look at $n = k^2 - u^2$ which factors as $(k-u)(k+u)$. Finally we then check these two numbers by division on n [15].

Example. Suppose $n = 2257$, which is both odd and composite. As an initial step, we perform trial division by all primes up to 7, none of which divide n. Now because $\lceil \sqrt{n} \rceil = 48$, we begin by looking at $a = 48^2$. So we have that $a - n = 47$, which is not a perfect square. Consequently, we examine the next square, $49^2 = 2401$ and $2401 - n = 144 = 12^2$. Thus we have that $n = 49^2 - 12^2$ which is the difference of squares so it factors as $(49 - 12)(49 + 12) = (37)(61)$. We can easily confirm that both numbers divide n evenly; because both numbers divide n, they are both factors of our number. Finally, since our goal is to find the prime factorization of n, we must check to see if both numbers are prime. Since n has no prime divisors below $\log 2257 = 7.7$, and we only need to check primes below the square roots of 37 and 61 it is clear that both numbers must be prime [19]. □

3.2 Kraitchik's Alterations

In the 1920's Maurice Kraitchik developed the idea that rather than $u^2 - v^2 = n$, as in Fermat's method, it might suffice for $u^2 - v^2$ to just be a multiple of n. Then it is only necessary to find integers u and v such that $u^2 \equiv v^2 \pmod{n}$. Equations of this nature, assuming they have solutions, have solutions that look like $u \equiv \pm v \pmod{n}$ or $u \not\equiv \pm v \pmod{n}$. If we have the second of these two options, it is true that while we have $n \mid (u^2 - v^2)$, n divides neither $(u - v)$ nor $(u + v)$. This means that the factors of n must somehow be split between the above sums, so we must take the greatest common divisor of n with each of $u - v$ and $u + v$. These greatest common divisors should be factors of n and from there we can completely factor the number [15].

Example. For our example of Kraitchik's Method suppose $n = 3427$. Then since $\lceil \sqrt{n} \rceil = 59$, the first square above n is $59^2 = 3481$, so then we take the next few squares modulo n. This gives us the list:

$$\begin{aligned} 59^2 &\equiv 54 \,(\text{mod } n) \\ 60^2 &\equiv 173 \,(\text{mod } n) \\ 61^2 &\equiv 294 \,(\text{mod } n) \\ 62^2 &\equiv 417 \,(\text{mod } n) \\ 63^2 &\equiv 542 \,(\text{mod } n) \\ 64^2 &\equiv 669 \,(\text{mod } n)\,. \end{aligned}$$

With no obvious square in sight, we move to the next step in Kraitchik's Method; we now factor each one of the above equivalences:

$$\begin{aligned} 54 &= 2 \cdot 3^3 \\ 173 &= 173 \\ 294 &= 2 \cdot 3 \cdot 7^2 \\ 417 &= 3 \cdot 139 \\ 542 &= 2 \cdot 271 \\ 669 &= 3 \cdot 223. \end{aligned}$$

Using the above factorizations we get that the product of 59^2 and 61^2, modulo n, is $2^2 \cdot 3^4 \cdot 7^2$ which is a perfect square. Hence we have $u^2 \equiv v^2 \,(\text{mod } 3427)$ where $u = 59 \cdot 61 \equiv 172 \,(\text{mod } 3427)$ and $v = 2 \cdot 3^2 \cdot 7 \equiv 126 \,(\text{mod } 3427)$. Since $172 \not\equiv \pm 126 (\text{mod } 3427)$, then we take the $\gcd\,(172 - 126,\ 3427) = \gcd\,(46,\ 3427) = 23$ and $\gcd\,(172 + 126, 3427) = \gcd\,(298,\ 3427) = 149$. Thus we get the factorization $3427 = 23 \cdot 149$. Note, this example contains the central idea of the quadratic sieve

method. In the Fermat Method we search exhaustively to find $u^2 - v^2 = n$, but in this example we compute relatively few squares, reduce them modulo n, factor them and try to put them together to form a square. □

In the Quadratic Sieve our goal is to find two integers x and y such that $x^2 \equiv y^2 \pmod{n}$. So, in practice, we will set a bound B and then we search for integers N, such that $N^2 - n$ is B-smooth. The idea behind this, which will be discussed in greater detail as the article proceeds, is that we will use the products of these $N^2 - n$ to find the above integers x and y by multiplying these integers together.

CHAPTER IV
SMOOTH NUMBERS

We will later describe in detail the Quadratic Sieve but it is helpful to have a general idea of what lies ahead. So, in general in Kraitchik's idea we want to find a large number of integers $N_1, N_2, \ldots, N_t$ with $({N_i}^2 \bmod n) = {p_1}^{e_1} \cdot {p_2}^{e_2} \cdot \cdots \cdot {p_r}^{e_r}$, for primes $p_1, \ldots, p_r < B$. Then one would multiply some of the N_i together to find a square that is congruent to another square modulo n. To do this we will need to solve a system of equations. In this system of equations our goal will be to create an even exponent for each of the p_i; because then we are guaranteed to have a square. Therefore we have two computational issues: the first is the difficulty of finding the B-smooth ${N_i}^2$, and the second is how big a system we will deal with. We first will study the smoothness issue.

As noted in the previous section we need to find a large number of B-smooth integers of the form $N^2 - n$. So when given an integer n with the intention of factoring it, is important to be able to determine a value for B. However, we must first be able to identify the number of primes below our chosen B. Consequently, define the number $\pi(B)$ as the number of primes on the interval $[1, B]$ [19]. Using this, in order to identify B we will be using some powerful results from number theory. The celebrated estimate for $\pi(B)$, given in the Prime Number Theorem gives us the following.

Theorem 1. *As* $B \longrightarrow \infty$, $\pi(B) \sim \dfrac{B}{\ln(B)}$ *[6].*

We mentioned before that our goal was to find a sequence of B-smooth numbers so that we get a product that is a square, so that we can factor with the Quadratic Sieve. So it is ideal that we find a theorem that proves that this is always possible with certain constraints.

Theorem 2. *If $m_1, m_2, \ldots, m_k$ are positive B-smooth integers and $k > \pi(B)$ then some non-empty subsequence of m_i has a product that is a square [19].*

Proof. Let m be a B-smooth number, and let $v(m)$ be the exponent vector of m. Suppose that the prime factorization of m is:

$$m = \prod_{i=1}^{\pi(B)} p_i^{v_i},$$

where p_i is the i^{th} prime number and each exponent, v_i, is a nonnegative integer such that $v(m) = \big(v_1, v_2, \ldots, v_{\pi(B)}\big)$. Then we can see that a subsequence of the above m_i has a product that is a square if and only if the sum of their exponent vectors has all even entries, that is, if and if the sum of their exponent vectors is congruent to the zero vector modulo 2. Now let $\mathbf{F}_2$ be the field with two elements and $\mathbf{F}_2^{\pi(B)}$ be the $\mathbf{F}_2$ vector space of dimension $\pi(B)$. By assumption we have that $k > \pi(B)$; hence by theorems of linear algebra the sequence of exponent vectors is linearly dependent. Therefore since this work is over the field $\mathbf{F}_2^{\pi(B)}$ there exists some subsequence whose sum is the zero vector. As a consequence, the product of the subsequence is a square [19]. □

4.1 Identifying B-Smooth Numbers

One option when trying to find B-smooth numbers is of course trial division. Now if B is small then there are few primes below B to check; so the number of trial divisions is $\max(\log_2 n,\ \pi(B))$. However since that division takes the most time of

all the operations we will be performing, from modular arithmetic to multiplication. Hence this quickly becomes very time-consuming. Therefore one can easily see that if B is a large number, say more than even fifteen digits, then the amount of time that trial division would take is unreasonable for our situation, especially considering that there are other possibilities. Hence, to find B-smooth numbers we will be avoiding division and instead use other methods to manipulate the data. In order to do this one may want to sieve through a sequence of numbers; a *sieve* is an algorithm that takes a given set and extracts numbers with desirable properties. The classic example of this is the Sieve of Eratosthenes which takes a list of n numbers and pulls out the primes. So, before defining our method of identifying B-smooth numbers we will first discuss the Sieve of Eratosthenes [15].

Sieve of Eratosthenes

Eratosthenes developed what we today call the Sieve of Eratosthenes which is a way of finding prime numbers by eliminating multiples of primes. This sieve begins with a list of numbers, starting with the number 2 and running integrally through any bound one may set; say that our bound is $X = 40$. Then the sieve allows us to find all prime numbers between 2 and 40. So we set it up as follows:

	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40

So we begin by circling or skipping the number 2, and crossing out every multiple of 2.

	$\overset{\downarrow}{2}$	3	~~4~~	5	~~6~~	7	~~8~~	9	~~10~~
11	~~12~~	13	~~14~~	15	~~16~~	17	~~18~~	19	~~20~~
21	~~22~~	23	~~24~~	25	~~26~~	27	~~28~~	29	~~30~~
31	~~32~~	33	~~34~~	35	~~36~~	37	~~38~~	39	~~40~~

Then the next number that we have that is not crossed out is 3; so we do the same as above with this prime, to get:

	2	$\overset{\downarrow}{3}$	~~4~~	5	~~6~~	7	~~8~~	~~9~~	~~10~~
11	~~12~~	13	~~14~~	~~15~~	~~16~~	17	~~18~~	19	~~20~~
~~21~~	~~22~~	23	~~24~~	25	~~26~~	~~27~~	~~28~~	29	~~30~~
31	~~32~~	~~33~~	~~34~~	35	~~36~~	37	~~38~~	~~39~~	~~40~~

Continuing in this manner, one identifies the next number that is not crossed out and eliminates the multiples of that number as well. This persists until all numbers are either prime or eliminated.

Still, aside from identifying primes, this sieve also allows one to find all of the prime factors of each number– if one additionally crosses out or records divisibility of all powers of p, the number we are currently sieving by, each time. One way to optimize the usefulness of this idea is to replace each number with the quotient of it with the current prime that is being sieved. If we also were to sieve with all powers of primes up to a number B then the entries with a 1 remaining are B-smooth. One can better understand the time that this process takes when we recognize that the time is similar to taking X times the sum of the reciprocals of each prime p and the reciprocals of all the powers and multiples of each p up to X. This is approximately $X \cdot \log(\log(B))$. Hence, the sieve gives us all B-smooth numbers up to X [15].

For our purposes, we are interested in finding B-smooth numbers in the sequence of polynomial $x^2 - n$ as $x \to \infty$. This will allow us to find numbers whose product is a square, thus using and then to apply Kraitchik's idea, with a few extra

complexities. Suppose we begin sieving with a prime p; we evaluate the congruence $x^2 - n \equiv 0 \,(\text{mod } p)$. There are three possible results: zero, one or two solutions. If there are no solutions, there is no sieving to do; on the other-hand if there are two solutions, call them s_1 and s_2 with both $(s_1)^2 - n$ and $(s_2)^2 - n$ are congruent to 0 modulo p. Then we find sets of numbers also congruent to 0 modulo p by adding multiples of p to each of the s_i. In this case, too, we have that the number of steps for each prime p is $\log(\log(B))$. As a result we gather B-smooth numbers by the above process [15].

Example. Suppose $n = 4183$ and let $B = 13$. So this sieve should gather all the 13-smooth numbers in the sequence of values, $x^2 - n$. Note that our quadratic residues need only go to $\lfloor\sqrt{4183}\rfloor = 64$. Then we look at the congruences $x^2 - 4183 \equiv 0 \pmod{p}$ for $p \in \{2, 3, 5, 7, 11, 13\}$. Then evaluating the congruences we get:

$$\begin{aligned} x^2 - 4183 &\equiv 0 \pmod 2 \\ x^2 - 1 &\equiv 0 \pmod 2 \\ x^2 &\equiv 1 \pmod 2 \\ x &\equiv 1 \pmod 2 \end{aligned} \qquad \begin{aligned} x^2 - 4183 &\equiv 0 \pmod 3 \\ x^2 - 1 &\equiv 0 \pmod 3 \\ x^2 &\equiv 1 \pmod 3 \\ x &\equiv 1, 2 \pmod 3 \end{aligned}$$

$$\begin{aligned} x^2 - 4183 &\equiv 0 \pmod 5 \\ x^2 - 3 &\equiv 0 \pmod 5 \\ x^2 &\equiv 3 \pmod 5 \\ x &\not\equiv 1, 2, 3, 4, 0 \pmod 5 \end{aligned} \qquad \begin{aligned} x^2 - 4183 &\equiv 0 \pmod 7 \\ x^2 - 4 &\equiv 0 \pmod 7 \\ x^2 &\equiv 4 \pmod 7 \\ x &\equiv 2, 5 \pmod 7 \end{aligned}$$

$$\begin{aligned} x^2 - 4183 &\equiv 0 \pmod{11} \\ x^2 - 3 &\equiv 0 \pmod{11} \\ x^2 &\equiv 3 \pmod{11} \\ x &\equiv 5, 6 \pmod{11} \end{aligned} \qquad \begin{aligned} x^2 - 4183 &\equiv 0 \pmod{13} \\ x^2 - 3 &\equiv 0 \pmod{13} \\ x^2 &\equiv 3 \pmod{13} \\ x &\equiv 6, 7 \pmod{13} \end{aligned}$$

Then by the above equivalences we can see that for $p = 2$ there is one solution, for $p = 3, 7, 11, 13$ there are two solutions and for $p = 5$ there are no solutions. We now take these solutions and add multiples of the modulus to them, giving us a set of quadratic residues for each modulus. For example if we look at $p = 3$, the solutions

are 1 and 2 modulo 3; then we add multiples of 3 to 1 getting $\{1, 4, 7, \ldots, 61, 64\}$ and do the same with 2, getting that the residue class for 3 is $\{1, 2, 4, 5, \ldots, 61, 62, 64\}$. So we obtain sets of possible B-smooth values, bounded by 64 to be odd integers, $\{1, 2, 4, 5, \ldots, 61, 62, 64\}$, $\{2, 5, 9, 12, \ldots, 60, 63\}$, $\{5, 6, 16, 17, \ldots, 60, 61\}$, and $\{6, 7, 19, 20, \ldots, 58, 59\}$. Consequently we have a large list of B-smooth numbers, which we can combine using multiplication to fully factor our number, $n = 4183$. □

4.2 Choosing B

As a general rule the choice of B is an important part of the Quadratic Sieve; this is because with a proper value for B, we can limit and shorten the time necessary for the overall process of sieving and consequently the entire sieve. Choosing the parameter B involves a delicate balance of size; if we choose B too small then while there are not many B-smooth residues to find, we may not have enough B-smooth values to use [6]. On the other hand if B is too large, even though B-smoothness is more common, we have to find many more numbers in order to get a linearly dependent set. So in order to be efficient we must find a balance, and to do so we need a measure of the probability that $x^2 - n$ is B-smooth [19].

Theorem 3. *The probability that a number, n, is B-smooth is approximately u^{-u}, where* $u = \dfrac{\ln(n)}{\ln(B)}$.

Proof. For our purposes we will assume that this theorem is true. However, the proof of this theorem can be found in the article *On a problem of Oppenheim concerning 'factorisatio numerorum'* in the 1983 Journal of Number Theory [6]. □

To find the size of B we need to compute the frequency of B-smooth numbers as a function of B and n. We search for B, computing the values of $x^2 - n$ from $x = \lceil \sqrt{n} \rceil$ to $x' = \lceil \sqrt{2n} \rceil$. Now if $\sqrt{n} < x < \sqrt{n} + n^{\epsilon}$ where $\epsilon > 0$ is small, then

the order of magnitude of $x^2 - n$ is $n^{\epsilon+\frac{1}{2}}$. So then we should change our value of u to equal $\frac{\ln(n)}{2\ln(B)}$ which is smaller than the previous u, allowing for a greater probability that a number is B-smooth [6].

Now if we let K be the number of primes up to B that we are going to sieve over then, heuristically, we have that $K \sim \frac{\pi(B)}{2}$. Then we need $K+1$ vectors so that we are guaranteed to have linear dependence, which is what we are striving to get. If the probability that x leads us to a B-smooth number is u^{-u} then the odds that a number will give us a successful result is 1 in u^u. So because we need to get $K+1$ values then the time to find them, $T(B)$, is represented as the function:

$$T(B) = u^u \cdot (K+1) \cdot \ln(\ln(B)) \text{ where } u = \frac{\ln(n)}{2\ln(B)} \text{ [6].}$$

Using the Prime Number Theorem paired with the fact that $K \sim \frac{\pi(B)}{2}$, we get that $\ln(T(B)) \sim S(B)$ where $S(B) = u \cdot \ln(u) + \ln(B)$. Then if we differentiate u with respect to B, we get $\frac{du}{dB} = \frac{-\ln(n)}{2 \cdot \ln^2(B)}$. By the definition of u we have, $\ln(u) = \ln(\ln(n)) - \ln 2 - \ln(\ln(B))$. Then

$$\begin{aligned}
\frac{dS}{dB} &= \frac{du}{dB} \cdot \ln(u) + \frac{du}{dB} + \frac{1}{B} \\
&= \frac{du}{dB}(\ln(u) + 1) + \frac{1}{B} \\
&= \frac{-\ln(n)}{2 \cdot \ln^2(B)} \cdot (\ln(\ln(n)) - \ln 2 - \ln(\ln(B)) + 1) + \frac{1}{B}.
\end{aligned}$$

Now in order to find an optimal value for B with the goal of minimizing time, we set the derivative equal to zero, getting that

$$c \cdot \sqrt{\ln(n)} < \ln(B) < d \cdot \sqrt{\ln(n) \cdot \ln(\ln(n))},$$

where c and d are constants. Hence $\ln(\ln(B)) \sim \frac{1}{2} \cdot \ln(\ln(n))$ [19]. Then using this we get that

$$\ln(B) \sim \tfrac{1}{2} \cdot \sqrt{\ln(n) \cdot \ln(\ln(n))},\ u \sim \sqrt{\frac{\ln(n)}{\ln(\ln(n))}},\ S(B) \sim \sqrt{\ln(n) \cdot \ln(\ln(n))}\ .$$

Thus we get that the optimal value for B is the upper bound of

$$L(n) \quad = \quad e^{\frac{1}{2} \cdot \sqrt{\ln(n) \cdot \ln(\ln(n))}}. \tag{4.1}$$

Hence we can accurately choose a value B to be the upper bound for our primes. In addition this adds some efficiency to the process of sieving– if we do not choose a B at least close to the optimal value identified in (4.1) the time may be lost, by sieving through unnecessary integers [19].

CHAPTER V
THE QUADRATIC SIEVE

Created in 1981 by Carl Pomerance, the Quadratic Sieve out-performed all previously known methods. In fact by 1990 it had doubled the previous factorable length of a number to 116-digits. And in 1994, Pomerance's Sieve factored the 126-digit RSA challenge number. This method is based, as its predecessors, on finding squares whose difference is 0 modulo n, where n is the number to be factored [15].

5.1 Basic Algorithm

We assume the integer n that we are given to factor, is an odd composite integer that is not a power of some number a, in other words $n \neq a^t$, $\forall\ a, k \in \mathbb{Z}^+$; it is safe to assume this because otherwise it would be easily factored. Let $B = \left\lceil L\left(n\right)^{1/2} \right\rceil$ and set $p_1 = 2$, $a_1 = 1$ with, as above, $K \sim \dfrac{1}{2}\,\pi\left(B\right)$. We now need to find our factor base; a *factor base* is a set, M, of primes such that each element of M is less than B, the smoothness bound, and for $q \in M$ we have that $\left(\dfrac{n}{q}\right) = 1$ [11]. It is important to mention that the calculation of the Legendre symbol is not difficult and the time approximation to do so is $O(\ln m^2)$ where m is essentially the number of digits of n; for more information one may look to *Prime Numbers* by Carl Pomerance [6]. Now, as we find these primes we shall label them $p_2, p_3, \ldots, p_K$. It is then necessary to find the values $\pm a_i$ such that $a_i^2 \equiv n \left(\mathrm{mod}\ p_i\right)$. We can do so in one of two methods: evaluate the primes modulo 8 or use $\mathbf{F}_{p^2}$ arithmetic [6].

Using the above conditions, the first method evaluates primes modulo 8. It is important to note that since p is an odd prime and 8 is an even integer, we have no cases where p is congruent to an even number modulo 8. For this method we will need to recall a well-known fact from number theory, namely for p a prime, $n^{p-1} \equiv 1 \pmod{p}$ for an integer n with $\gcd(n, p) = 1$. As in [6] we have, for

$$p \equiv 3, 5, 7 \pmod{8}$$

the square root of $n \pmod{p}$ is as follows. If $p \equiv 3, 7 \pmod{8}$ then

$$x \equiv n^{(p+1)/4} \pmod{p}$$

If $p \equiv 5 \pmod{8}$ then

$$x \equiv n^{(p+3)/8} \pmod{p}$$

Then let $c \equiv x^2 \pmod{p}$, which implies that $c \equiv \pm n \pmod{p}$. Suppose that $c \neq x \pmod{p}$, then $x = x \cdot 2^{(p-1)/4} \pmod{p}$. Next we consider the case where $p \equiv 1 \pmod{8}$, and begin by picking an integer $d \in [2, p-1]$ such that $\left(\frac{d}{p}\right) = -1$. Now because p is an odd prime, it is clear that $p - 1$ is even; so let $p - 1 = 2^s \cdot t$ where t is odd. Also let $A = a^t \pmod{p}$, $D = d^t \pmod{p}$ and $m = 2\mu$ where $0 \leq \mu < 2^{s-1}$. So for $0 \leq i < s$, i an integer, we get that $AD^m \equiv 1 \pmod{p}$. Hence

$$x = n^{(t+1)/2} D^{m/2} \pmod{\text{p}}$$

The second option for solving the congruence $a^2 \equiv n \pmod{p}$ is to solve using $\mathbf{F}_{p^2}$ arithmetic. To do so we use the same criterion as in the first method, noting that p is an odd prime. We begin by finding an integer $t \in [0, p-1]$ such that $\left(\frac{t^2 - n}{p}\right) = -1$. Now the probability that any particular t will be successful

in this step is $(p-1)/2p$. But once such a t is found, we can let

$$x = \left(t + \sqrt{t^2 - n}\right)^{(p+1)/2}$$

in $\mathbf{F}_{p^2}$. From here, we need only do simple arithmetic to solve the congruence $x^2 \equiv n \pmod{p}$ [6]. So with this step completed, we have found a solution for the congruence $a_i^2 \equiv n \pmod{p_i}$.

The next or second step in the Quadratic Sieve algorithm is to sieve the sequence $(x^2 - n)$ for $x = \lceil \sqrt{n} \rceil, \lceil \sqrt{n} \rceil + 1, \ldots$ to find B-smooth values until we have a set, call it S, of $K+1$ pairs of the form $(x, x^2 - n)$. At times it may be more efficient to center the numbers x at $\lceil \sqrt{n} \rceil$ rather than going strictly above this value; the advantage here is that we are able to work with smaller numbers, which is more efficient. Suppose that $2 \leq B \leq \sqrt{N}$. Then we adapt the Sieve of Eratosthenes and sieve with not only the primes $p \leq \sqrt{N}$ but also for each p we sieve with the powers of p less than or equal to $\sqrt{N}$. Furthermore, we also must sieve with the powers of p that do not exceed B; sieving in this manner allows for the factorization of these numbers without using division, which as we mentioned before is expensive time-wise. Beside each of the numbers we start a list of the prime factors as we sieve through our factor base. And each time we sieve by a number in our factor base it is added to the list if and only if it is indeed a factor [6].

Example. Suppose that $N = 201316694471$. Then by the above instructions for selecting B, $B = 10$ and technically we need only sieve up to $\sqrt{N} = 448683$. However, for this example we shall only sieve up to 40, since our immediate goal is not to actually factor N. So after sieving through using our factor base, which is $\{2, 5, 7\}$ and the powers of the primes not exceeding 40 in that factor base, we get the following list:

	$2_{\{2\}}$	$3_{\{3\}}$	$4_{\{2^2\}}$	$5_{\{5\}}$
$6_{\{2,3\}}$	$7_{\{7\}}$	$8_{\{2^3\}}$	$9_{\{3^2\}}$	$10_{\{2,5\}}$
$11_{\{11\}}$	$12_{\{2^2,3\}}$	$13_{\{13\}}$	$14_{\{2,7\}}$	$15_{\{3,5\}}$
$16_{\{2^4\}}$	$17_{\{17\}}$	$18_{\{2,3^2\}}$	$19_{\{19\}}$	$20_{\{2^2,5\}}$
$21_{\{3,7\}}$	$22_{\{2,11\}}$	$23_{\{23\}}$	$24_{\{2^3,3\}}$	$25_{\{5^2\}}$
$26_{\{2,13\}}$	$27_{\{3^3\}}$	$28_{\{2^2,7\}}$	$29_{\{29\}}$	$30_{\{2,3,5\}}$
$31_{\{31\}}$	$32_{\{2^5\}}$	$33_{\{3,11\}}$	$34_{\{2,17\}}$	$35_{\{5,7\}}$
$36_{\{2^2,3^2\}}$	$37_{\{37\}}$	$38_{\{2,19\}}$	$39_{\{3,13\}}$	$40_{\{2^3,5\}}$

In the above list the set to the right of each number is that number's prime factorization, for instance $2 \cdot 7$ is the prime factorization of 14. So, if we were to continue up to 448683 we would then go through the chart and gather a list of all B-smooth values, in other words gather a list of all numbers, $x^2 - n$, with no prime divisor exceeding 10 in our case. So in the above example, the B-smooth values form the set $\{2, 4, 5, 7, 8, 10, 14, 16, 20, 25, 28, 32, 35, 40\}$. □

While this is useful it is not exactly what we need to use to complete this step of the algorithm; our algorithm asks for B-smooth values when sieving through polynomial values, $x^2 - n$, and thus far what we have described sieves through a numerical sequence. To do this we let $f(x) = x^2 - n$ and we will look at the list $f(\lceil\sqrt{n}\rceil), f(\lceil\sqrt{n}\rceil + 1), \ldots, f(n)$. Now we should note that when we sieve through with each prime from our factor base, as described above, if $f(a) \equiv 0 \pmod{p}$ then p clearly divides $f(a)$. Then $f(a + kp) \equiv 0 \pmod{p}$, $\forall\ k \in \mathbb{Z}$. Because this is true for each prime p in our factor base and each one can have as many solutions as $deg(f) = 2$ this leaves few options for us to check. Now from here we can adopt the above sieve to complete the process; allowing us to gather the required $k + 1$ B-smooth numbers [6].

The third step of the algorithm is, for each pair $(x, x^2 - n) \in S$, where S is our set of $K + 1$ B-smooth values, to find the prime factorization of the second element in each pair, which we write as

$$\prod_{i=1}^{k} (p_i)^{e_i}$$

Then we establish an exponent vector for each value x_i, which we notate as

$$\vec{v}\left({x_i}^2 - n\right) \;=\; (e_1, e_2, \ldots, e_k)$$

Now, take the above vector for each value of x and reduce it modulo 2. Since we have $K+1$ elements in S we do the same to each element and then form a matrix that will be of size $(K+1) \;\times\; K$. Using linear algebra we can, from here, find a nontrivial subset of row vectors whose sum is the zero vector,

$$\vec{v}\,(x_1) + \vec{v}\,(x_2) + \ldots + \vec{v}\,(x_k) = \vec{0}.$$

This sum can be found by forming a matrix and performing Gaussian Elimination.

Now that we have the prime factorization of each $x^2 - n$ and have used the exponent vectors to find a sum that is the zero vector modulo 2, there is one final step, that is, let

$$x \;=\; x_1 \cdot x_2 \cdot \cdots \cdot x_k \,(\text{mod } n)$$

With this in place, we need to define a second variable, y such that y^2 is the product of these ${x_i}^2 - n$. In other words, let

$$y^2 \;=\; \left[\left({x_1}^2 - n\right) \cdot \left({x_2}^2 - n\right) \cdot \cdots \cdot \left({x_k}^2 - n\right)\right] (\text{mod } n)\,.$$

Then,

$$y \;=\; \sqrt{\left({x_1}^2 - n\right) \cdot \left({x_2}^2 - n\right) \cdot \cdots \cdot \left({x_k}^2 - n\right)}\,(\text{mod } n)\,.$$

So finally we let $d = \gcd(x - y, n)$. Then we divide n by d, which allows us to obtain the remaining factors of n. Upon division of n by d we can again use the

Quadratic Sieve Algorithm to find divisors of $\frac{n}{d}$, if it is needed [6]. As a general rule for a large number, say upwards of 130 digits, one may have to apply the Quadratic Sieve several times to find the complete factorization of n. To illustrate this we will do a small example.

5.2 Example

Suppose we wanted to factor the number $n = 62113$. Then $B = 4$; however as discussed in Chapter 3 of this paper, if our bound B is too small it is difficult to find enough B-smooth values to sieve through. So for our purposes we shall let $B = 37$. Then we need to find all primes smaller than B that satisfy the first step of the Quadratic Sieve algorithm–that is, the set $M = \{p_1, p_2, \ldots, p_K\}$ such that $\left(\frac{n}{p_i}\right) = 1$. Thus we shall begin by assigning $p_1 = 2$ since $\left(\frac{n}{2}\right) = 1 \ \forall \ n \in \mathbb{Z}$; note this is because we would then be solving the congruence $x^2 \equiv 62113 \pmod{p}$. Since the calculations for this require only simple modular arithmetic we will express our findings in the form of a chart, as seen below [8]; an explanation of these calculations can be found in the book *Prime Numbers* by Carl Pomerance.

p	2	3	5	7	11	13	17	19	23	29	31	37
$\left(\frac{62113}{p}\right)$	1	1	-1	1	-1	-1	1	1	1	1	1	1

Table 5.1: Legendre Symbol on Primes Below B

Hence our factor base is $M = \{2, 3, 7, 13, 23, 29, 31, 37\}$. It follows that $K = 8$, so we will need to find nine B-smooth values in order to factor n.

We begin by looking at an interval of length approximately fifty around $\lceil \sqrt{n} \rceil = 249$. Now we should observe that our interval, $[224, 273]$ was hueristically

x	p, $p \mid x^2 - n$	Factorization	x	p, $p \mid x^2 - n$	Factorization
224	3, 23		249	2, 7	$-1 \cdot 2^4 \cdot 7$
225	2		250	3	
226	3, 13		251	2, 3, 7	$2^3 \cdot 3 \cdot 7$
227	2, 3, 7	$-1 \cdot 2^3 \cdot 3^3 \cdot 7^2$	252	13	
228	7		253	2, 3	
229	2, 3, 13, 31	$-1 \cdot 2^3 \cdot 3 \cdot 13 \cdot 31$	254	3	
230	3, 37		255	2, 7, 13	$2^5 \cdot 7 \cdot 13$
231	2		256	3, 7	
232	3		257	2, 3	
233	2, 3		258		
234	7		259	2, 3, 23	$2^3 \cdot 3^3 \cdot 23$
235	2, 3, 7		260	3, 31	
236	3, 23, 31	$-1 \cdot 3^2 \cdot 23 \cdot 31$	261	2	
237	2		262	3, 7	
238	3		263	2, 3, 7	$2^4 \cdot 3^2 \cdot 7^2$
239	2, 3, 13	$-1 \cdot 2^7 \cdot 3 \cdot 13$	264		
240			265	2, 3, 13	$2^4 \cdot 3 \cdot 13^2$
241	2, 3, 7	$-1 \cdot 2^6 \cdot 3^2 \cdot 7$	266	3	
242	3, 7, 13	$-1 \cdot 3 \cdot 7 \cdot 13^2$	267	2, 31, 37	$2^3 \cdot 31 \cdot 37$
243	2		268	3, 13	
244	3		269	2, 3, 7	
245	2, 3, 29	$-1 \cdot 2^3 \cdot 3^2 \cdot 29$	270	7, 23	
246			271	2, 3	
247	2, 3, 23	$-1 \cdot 2^4 \cdot 3 \cdot 23$	272	3	
248	3, 7, 29	$-1 \cdot 3 \cdot 7 \cdot 29$	273	2	

Table 5.2: Sieving Table

determined; this is because there is no algorithm that allows one to accurately determine the sieving interval. Furthermore, if one were to pick the bound too small, meaning that one could not find $K+1$ numbers that are factorable using the set M, then it would be relatively easy to extend the interval to a larger one. On the other hand, if you initially pick your sieving interval too large it will take more time. A table with the listing of the factorizations of each value $x^2 - n$ can be seen below; note that in the table above $p \in M$ and the factorization column only records the factorization of numbers that are completely factored when sieving through by set M and the powers of the primes in M.

So after calculating the prime factorization for the fifty polynomial values surrounding $x^2 - \lceil\sqrt{n}\rceil$ we get a list of B-smooth numbers; note we only need the first 9 values. Hence we get the set $S = \{227,\ 229,\ 236,\ 239,\ 241,\ 242,\ 245,\ 247,\ 248\}$. Using table (5.2) we can gather the exponent vectors of each element modulo 2 to form the following matrix:

$$A = \begin{array}{c} \\ 229 \\ 236 \\ 239 \\ 241 \\ 242 \\ 245 \\ 247 \\ 248 \\ 249 \end{array} \begin{array}{c} \begin{array}{ccccccccc} -1 & 2 & 3 & 7 & 13 & 23 & 29 & 31 & 37 \end{array} \\ \begin{pmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \end{array}$$

Then we look for a vector $\vec{v}$ such that

$$[A] \cdot \vec{v} = \vec{0}.$$

Using Gaussian Elimination we find that one solution is

$$\vec{v} = (1, 1, 1, 0, 0, 1, 0, 1, 0, 1)\,.$$

Then we take the corresponding x values to get

$$\begin{aligned} x &= 229 \cdot 236 \cdot 239 \cdot 242 \cdot 247 \cdot 249 \\ &= 192245885018616 \\ &\equiv 24147 \pmod{n} \\ y &= \sqrt{(229^2 - n) \cdot (236^2 - n) \cdot (239^2 - n) \cdot (242^2 - n) \cdot (247^2 - n) \cdot (249^2 - n)} \\ &= 135961516255081660416 \\ &\equiv 28658 \pmod{n} \end{aligned}$$

Then we get the greatest common divisors:

$$\gcd(x - y,\ n) = 347, \qquad \gcd(x + y,\ n) = 179.$$

Thus we can get that the prime factorization of $n = 62113$ is $n = 179 \cdot 347$ [8].

5.3 Parallel Quadratic Sieve

Now in the Quadratic Sieve the most laborious part of the algorithm is the sieving over the given interval. Consequently, one may seek to reduce the time it takes to sieve in order to make factoring a given integer n more efficient. So we divide the sieving interval into blocks, with the number of blocks corresponding to the number of processors available. The benefit of performing the parallel process in this manner is that it requires minimum communication between the processors, allowing for a greater amount of energy to expended on the actual sieving. Other

than sieving, another lengthy process is the Gaussian Elimination of the exponent matrix. However this is not as well suited for division between computers because it requires the use of sequential steps. In most cases, the separation of these steps may actually end up increasing the time necessary to complete the Gaussian Elimination of the matrix.

5.4 Multiple Polynomial Quadratic Sieve

As a variation of the Quadratic Sieve, Peter Montgomery suggested taking several polynomials instead of the one polynomial, x^2-n, in the Quadratic Sieve. This Sieve as the name suggests takes multiple polynomials of the form $g(x) = ax^2 + 2bx + c$ with the values for $a,\ b,\ c$ determined by guidelines defined below [21].

We begin by letting a be a perfect square. Next pick b such that $b < a$ with $b^2 \equiv n \pmod{a}$. Note that this only works if $\forall\ p \mid a$, for p a prime, n is a square modulo p. Thus we pick a such that $\forall\ p \mid a$ we have $\left(\frac{n}{p}\right) = 1$. In order to complete the setup of our equation $g(x)$ we pick c so that $b^2 - ac = n$ [21]. Then

$$\begin{aligned} a \cdot g(x) &= (ax)^2 + 2abx + ac \\ &= a^2x^2 + 2abx + b^2 - n \\ &= (ax+b)^2 - n, \end{aligned}$$

implying that

$$(ax+b)^2 \equiv a \cdot g(x) \pmod{n}.$$

Next set up the sieving interval to be $2M$. With the above equation we can see that the conditions for b can be modified to limit the possibilities further, that is $|b| \leq \frac{1}{2}a$. And so we can let the interval $2M = [-M, M]$. Now in order to make use of this we want to maximize $g(x)$. Because $a \geq 0$, which is implied by the conditions

above for b, then the maximization of our equation must be at its endpoints. Then the value would be approximately $\left(a^2M^2 - n\right)/a$. Then the least or smallest value is at $x = 0$, which is about $-n/a$. So if we set these two expressions approximately equal to each other then we get $a \approx \sqrt{2n}/M$ [21].

In the Quadratic Sieve our smoothness bound B was $\left\lceil L\left(n\right)^{1/2}\right\rceil$, which made the length of our interval $M = B^2$. However because we have more than one polynomial we can create a smaller bound, mainly $M = B$. This allows us, for a large number n, to sieve over a much smaller interval than when we use only one polynomial. Then if all of the above criteria for a are met, then we will get only one possible equation using that particular a [21]. Then we have a polynomial where

$$|g\left(x\right)| \;<\; \tfrac{1}{\sqrt{2}}M\sqrt{n} \;=\; \frac{M\sqrt{2n}}{2}$$

5.5 Zhang's Special Quadratic Sieve

The speed of the quadratic sieve is determined by the size of the quadratic residues, mainly the smaller the better. By 1998 M. Zhang created a method of making the residues smaller than the quadratic sieve does normally, but only for certain integers n; thus his sieve is called the special quadratic sieve, SQS. To begin using this sieve take the number n which is an odd composite such that $n \neq a^d,\ \forall\ d > 1$ with $d \in \mathbb{Z}$. Suppose that our number can be written as

$$n \;=\; m^3 + a_2m^2 + a_1m + a_0 \tag{5.1}$$

with $m, a_i \in \mathbb{Z}$, $i \in \{0, 1, 2\}$ and $m = n^{1/3}$. Now let $b_i \in \mathbb{Z}$ with

$$x \;=\; b_2m^2 + b_1m + b_0 \tag{5.2}$$

with the same conditions on m as in the above equation. Now by (4.1) we have

$$m^3 \;\equiv\; -a_2m^2 - a_1m - a_0 \pmod{n}$$

$$m^4 \equiv \left({a_2}^2 - a_1\right) m^2 + \left(a_1 a_2 - a_0\right) m + a_0 a_2 \pmod{n}.$$

Then by substitution we get

$$x \equiv c_2 m^2 + c_1 m + c_0 \pmod{n}, \tag{5.3}$$

with

$$\begin{aligned} c_2 &= \left({a_2}^2 - a_1\right) {b_2}^2 - 2a_2 b_1 b_2 + {b_1}^2 + 2b_0 b_2 \\ c_1 &= \left(a_1 a_2 - a_0\right) {b_2}^2 - 2a_1 b_1 b_2 + 2b_0 b_1 \\ c_0 &= a_0 a_2 {b_2}^2 - 2a_0 b_1 b_2 + {b_0}^2. \end{aligned}$$

Note that we can choose our b_i such that $c_2 = 0$, so for b an integer, let

$$b_2 = 2, \quad b_1 = 2b, \quad b_2 = a_1 - {a_2}^2 + 2a_2 b - b^2$$

Assuming that we have these values for the above b_i then we get

$$x\left(b\right)^2 \equiv y\left(b\right) \pmod{n} \tag{5.4}$$

for

$$\begin{aligned} x\left(b\right) &= 2m^2 + 2bm + a_1 + {a_2}^2 + 2a_2 b - b^2 \\ y\left(b\right) &= \left(4a_1 a_2 - 4a_0 - \left(4a_1 + 4{a_2}^2\right) b + 8a_2 b^2 - 4b^3\right) m + \\ &\quad 4a_0 a_2 - 8a_0 b + \left(a_1 - {a_2}^2 + 2a_2 b - b^2\right)^2 \end{aligned}$$

Now if we sieve to find smooth values of $y\left(b\right)$ as b runs through small numbers, we can then use the exponent vectors from our smooth vectors. If we then form a matrix and find a subset of the rows of the matrix then we can obtain two squares modulo n and proceed as in the original quadratic sieve [6].

5.6 Time Approximation for Factoring n

To fully understand the efficiency of the Quadratic Sieve we must investigate the time that it takes to perform it on a variety of numbers. In this section we will be using the notion of bits; a *k-bit* number is an integer n such that $k = \log_2 n$. Now to begin we will be using the computer algebra system Magma on a computer with 2 Dual Core Intel Xeon processors at 3 Ghz with 8 GB of RAM running LINUX [2].

Before observing time approximations for the Quadratic Sieve one can calculate an estimate for Gaussian Elimination, used after one has obtained a matrix of exponent vectors. Note that we expect that an $n \times n$ matrix would be similar to n^3 in complexity. Now, we began by looking at a 1000 by 1000 matrix; the elimination process was completed in 1.25 seconds. We then doubled the size of each side of the matrix and calculated that for a 2000 by 2000 matrix it takes about 5 seconds to complete the row reductions. Thus we can see that the time increases by a factor of four when the sides are doubled. Then for a 4000 by 4000 matrix it would take approximately 20 seconds to row reduce this matrix. Following this pattern, to reduce a 32000 by 32000 matrix it would take 1280 seconds which is about 21 minutes. Therefore Magma makes Gaussian Elimination of an $n \times n$ matrix closer to n^2 in complexity. Moreover, using the above information we can make a time approximation; for an $n \times n$ where $n = 2^k \cdot 1000$ it will take approximately $1.25 \cdot 4^k$ seconds. Hence it is clear that the time to perform Gaussian Elimination in Magma on a matrix with entries from $\mathbb{Z}_2$ increases quickly.

Using the Multipolynomial Quadratic Sieve in Magma we were able to compute the time it takes to factor integers of various bit sizes using the average of our trials. In the table below we should observe that the time approximation of numbers of bit length 120 to 210 were averaged from 200 trials while those with bit length 210 to 250 are an average of 100 trials; the difference in the trials was mainly

due to time constraints. Also note that the time approximation is done in seconds.

Bit Length	Time Approximation
120	.1264
130	.2192
140	.41235
150	.89385
160	1.4568
170	2.9315
180	5.75525
190	12.0601
200	24.2353
210	55.0786
220	93.4357
230	161.87
240	292.4
250	492.246

Table 5.3: Magma Time Approximation

These numbers were RSA numbers, meaning that they are of the form $n = p \cdot q$, and the time approximation estimates the time it took Magma using the above computer, to factor numbers of the specified bit number.

In addition to these smaller integers, with a 120-bit number being approximately 36 digits in length, we used Magma to factor to two 100 digit numbers. It took Magma 17 hours to factor the first number and 18 hours to factor the second. Therefore one can see how important efficiency becomes when our target integers take upward of 17 hours to factor.

CHAPTER VI
DEVELOPING NEW METHODS

The number field sieve, which stems from Pollard's 1988 suggestion for factoring numbers close to a power of another integer, has become the most powerful factoring method currently in use [12]. Pollard's method was based on the use of algebraic number fields [15].

6.1 Pollard's Algorithm

Pollard's method began with a number n to be factored and an irreducible monic polynomial, $f(x)$, over the integers. We then find an integer m so that $f(m) \equiv 0 \pmod{n}$. Pollard suggested that the polynomial have $\deg(d)$, where d would be approximately four or five for a 100 to 200 digit number. Now let $\alpha \in \mathbb{C}$ such that $f(\alpha) = 0$. Then because

$$f(\alpha) = 0, \qquad f(m) \equiv 0 \pmod{n}$$

we have a map $\phi : \mathbb{Z}[\alpha] \longrightarrow \mathbb{Z}/n\mathbb{Z}$, defined by $\phi(a + b\alpha) = a + bm \pmod{n}$ for $m \in \mathbb{Z}/n\mathbb{Z}$ and $a, b \in \mathbb{Z}$. From these conditions we can easily see that ϕ is well-defined and even better a ring homomorphism. Next we suppose that we have a set, S, of coprime integer pairs $\langle a, b\rangle$ such that two properties are met:

1. the product of all the $a - \alpha b$ with $\langle a, b\rangle \in S$ is a square, say γ^2 in $\mathbb{Z}[\alpha]$;

2. the product of $a - mb$ for all $\langle a, b\rangle \in S$ is a square, say v^2 in $\mathbb{Z}$.

So then for $u \in \mathbb{Z}$ where

$$\phi(\gamma) \equiv u \pmod{n}. \tag{6.1}$$

Then

$$\begin{aligned} u^2 &\equiv \phi(\gamma)^2 &&= \phi\left(\gamma^2\right) \\ &= \phi\left(\textstyle\prod_{\langle a,b\rangle \in S}(a-\alpha b)\right) &&= \displaystyle\prod_{\langle a,b\rangle \in S} \phi(a-\alpha b) \\ &= \textstyle\prod_{\langle a,b\rangle \in S}(a-mb) &&= v^2 \ (mod\ n) \end{aligned}$$

Thus we have two squares which are congruent modulo n. Now if we apply Kraitchik's method, since $u \not\equiv \pm v \ (mod\ n)$ all we have left to do is take $\gcd(u-v, n)$. Hence we have a factor of n [15].

Now in order to find this a and b, we use the quadratic sieve by fixing b and sieving a over a given interval so that $a - mb$ is a square over the integers. We then repeat this process to get all elements of S. From here we can essentially eliminate the values for which the first property does not hold. But unfortunately this does not generalize to work for an arbitrary integer n [15].

6.2 Development of the General Number Field Sieve

Reevaluating Pollard's algorithm we want to find an $f(x)$ and m such that the polynomial $f(x)$ is monic and irreducible over the integers and $f(m) \equiv 0 \pmod{n}$. We do this by finding $f(x)$ last. The first step then is to pick a degree d for $f(x)$. Then let $m = \left\lfloor n^{1/d} \right\rfloor$. We can then write

$$n = m^d + c_{d-1}m^{d-1} + \cdots + c_0$$

where $0 \leq c_i < m$. One should note that if $n > (2d)^d$ then $c_d = 1$. Then we get that

$$f(x) = x^d + c_{d-1}x^{d-1} + \cdots + c_0. \tag{6.2}$$

With this polynomial now we must only discuss its irreducibility. So what if we suppose that $f(x)$ is reducible, then we can write $f(x) = g(x)\,h(x)$ for some $g, h \in \mathbb{Z}[x]$. Then

$$n = f(m) = g(m)\,h(m)$$

gives us two nontrivial factors– which is our goal. So if $f(x)$ is reducible then it quickens our result. However since more than likely $f(x)$ will be irreducible and we can proceed as above– using Pollard's method [15].

In 1990 the general number field sieve was established. The remaining changes paired with the few above made Pollard's sieve a major stepping stone for the general number field sieve. These changes were made by Joe Bueler, Hendrik Lenstra and Carl Pomerance [15]. Let norm of $a - \alpha b$ be notated as $N(a - \alpha b)$ over the rationals. This norm is constructed by taking a polynomial $g(x) \in \mathbf{F}[x]$ with $g(\beta) = 0$ where $\beta \in \mathbf{F}[\alpha]$. Then the norm of β is the constant term of our function g. So we can define a map, ϕ from $\mathbf{F}[\alpha]$ to $\mathbb{Q}$ such that $\phi(\beta) = N(\beta)$ and because as in linear algebra, this is a homomorphism then we get that $N(a - \alpha b)$ is the product of principal ideals $P_1, P_2, \ldots, P_k \in \mathbb{Z}$. Now we can see that

$$N(a - \alpha b) = b^d f(a/b).$$

We also want to define $a - \alpha b$ to be Y-smooth if $N(a - \alpha b)$ is Y-smooth. One should note that since the norm is multiplicative, if we have that the product of various algebraic integers, $a - \alpha b$, is the square of an algebraic integer, then so is the product of the corresponding norms. However one should also note that the

reverse is not necessarily true– mainly if the product of norms of algebraic integers is the square of an algebraic integer, the product of the corresponding $a - \alpha b$ need not be a square [15].

Example. Say we have the algebraic integers $1 - i,\ 1 + i$. Now the product of these is 2. But the norm of their product is $N(2) = 4$. So we can easily see that having the norm of the product of algebraic integers be a square does not guarantee that the product of the numbers is a square. □

Now let p be prime and let R_p be the set of solutions to the congruence $f(x) \equiv 0\ (mod\ p)$. Now if $p \mid N(a - \alpha b)$ then some prime ideal above p divides $(a - \alpha b)$. Another problem is that we have not established whether or not $\mathbb{Z}[\alpha]$ is the full ring of algebraic integers in $\mathbb{Q}[\alpha]$. While other difficulties still exist we should note that the reason that the Number Field Sieve is considered to be a more attractive alternative for n over 150-digits is because the running time for large numbers in the Number Field Sieve is much shorter than that of the Quadratic Sieve [15]. The basic idea of the number field sieve is that if $N(a - \alpha b)$ factors into small primes, then the ideal generated by $(a - \alpha b)$ in $\mathbb{Z}[\alpha]$ factors into small prime ideals. The gerneral plan is to use a factor base of prime ideals in $\mathbb{Z}[\alpha]$, and seek to mimic the scheme of the quadratic sieve. This can be done successfully, but there are many technical details thus making a full discussion of this topic beyond the scope of this paper.

6.3 Further Factoring Interests

Aside from the Quadratic Sieve and Number Field Sieve the Elliptic Curve Factorization Method, referred to as the ECM, is much used. The ECM is very efficient for finding factors of moderate size and even enormous integers n. However, generally

the application of this is small integers and as mentioned in a previous section, is often tried before the Quadratic Sieve in the factoring process of a given n. For our purposes the description of the ECM involves algebra and techniques that are beyond the scope of this paper, but further information can be found in the book *Prime Numbers* by Carl Pomerance.

It is known that the eventually the Number Field Sieve will reach its limits of efficiency, just as in the past it was discovered that over 150-digits the Quadratic Sieve is no longer the most efficient factorization method. Still, in any case, it is clear that there is much to discover in the efficiency and methods of factoring.

BIBLIOGRAPHY

[1] Briggs, Matthew E. "An Introduction to the General Number Field Sieve." Thesis. Viriginia Polytechnic Institute and State U, 1998.

[2] Bosma, Wieb, et al. "The Magma algebra system I. The user language". J. Symbolic Comput. 24(3-4), Springer-Verlag (1997):235-265.

[3] Bowman, Kim, et al. "Analyzing the Quadratic Sieve Algorithm." Clemson U, 2004.

[4] Case, Michael. "A Beginner's Guide to the General Number Field Sieve." Unpublished essay. Oregon State U, 2003.

[5] Chalkias, Konstantinos, et al. "Implementing Authentication Protocol for Exchanging Encrypted Messages via an Authentication Server based on Elliptical Curve Cryptography with the ElGamal's Algorithm." Proceedings of World Academy of Science, Engineering and Technology 7 (Aug. 2005): 137-42.

[6] Crandall, Richard, and Carl Pomerance. Prime Numbers. 2nd ed. New York: Springer, 2005.

[7] Davis, J.A., and D. B. Holdridge. "Factorization Using the Quadratic Sieve Algorithm." Report SAND. Albuquerque, N.M.: Sandia National Labratories, 1983.

[8] Hulpke, Alexander. "Factorization of $n = 87463$ with the Quadratic Sieve." Unpublished paper. Colorado State U, 2004.

[9] Kechlibar, Marian. "The Quadratic Sieve- Introduction to Theory with Regard to Implementation Issues." Charles University in Prague, 2005.

[10] Landquist, Eric. "Possible Ways to Extend Zhang's Special Quadratic Sieve." Unpublished essay. U of Illinois at Urbana- Champaign, 2003.

[11] Landquist, Eric. "The Quadratic Sieve Factoring Algorithm." Unpublished Essay. U of Virginia, 2001.

[12] Lenstra, A. and H. Lenstra, Jr. Eds. "The Development of the Number Field Sieve." Lecture Notes in Mathematics 1554 (1993).

[13] Lenstra, Arjen K. "Integer Factoring." Designs, Codes and Crytography. Vol. 19. Boston: Kluwer Academic Publishers, 2000: 101-128.

[14] Leyland, Paul, et al. "MPQS with Three Large Primes." in Proceedings of the 5th International Symposium on Algorithmic Number Theory. *Lecture notes in Computer Science* 2369, Springer-Verlag (2002):446- 60.

[15] Pomerance, Carl. "A Tale of Two Sieves." Notices of the American Mathematical Society Dec. 1996: 1473-1485.

[16] Pomerance, Carl, eds. Cryptology and Computational Number Theory. Vol. 42. Providence, R.I.: American Mathematical Society, 1990.

[17] Pomerance, Carl. "Cryptology and Computational Number Theory– An Introduction." Carl Pomerance, 1-12.

[18] Pomerance, Carl. "Factoring." Carl Pomerance, 27-48.

[19] Pomerance, Carl. "Smooth Numbers and the Quadratic Sieve." 2005, <http://websites.math.leidenuniv.nl/algebra/sieving.pdf>.

[20] Pomerance, Carl. "Smooth orders and Cryptographic Applications." Lecture notes in Computer Science 2369 (2002):338-48.

[21] Pomerance, Carl. "The Quadratic Sieve Factoring Algoritm." Lecture notes in Computer Science 209 (1985):169- 82.

[22] Rosen, Kenneth H. Elementary Number Theory and Its Applications. 5th ed. Boston: Pearson, 2005.

[23] "RSA BSAFE Crptyo-J: Crypto for Java, Developer's Guide." RSA. 2001. 21 Apr. 2008 <http://www.rsa.com>.