Cryptography, Primality Testing and The Extended Riemann Hypothesis

Radu Mihaescu*

January 19, 2002

Mathematics Department
Princeton University
Princeton, NJ 08544

Abstract

Fast primality testing, with numerous applications, one of which being to cryptography, is one of the most important and difficult problems in algorithm theory. The Miller-Rabin primality testing algorithm tests the primality of a number p in time proportional to $log^3(p) * loglog(p) * N(p)$, where$N(p)$ is the smallest quadratic non-residue modulo p. Based on the Extended Riemann Hypothesis, Ankeny proved that the size of N(p) is $O(log^2(p))$. In this project I will present the Miller-Rabin test and other alternative methods for primality testing, study the proof given by Ankeny for the size of $N(p)$ and provide some empirical data for the value of $N(p)$ on the first ten million primes and a few samples of larger consecutive primes.For primes of size up to approximatively 200 billion I have found values of $N(p)$ that range up to values of 45,000, suggesting that, if true, Ankeny's bounds have a quite large constant of proportionality (around 98).

*E-mail: `mihaescu@princeton.edu`

1 Motivation: Cryptography and Public Key Cryptosystems

A cryptosystem represents a method of enciphering plaintext. This usually means a 1-1 function between all possible plaintext messages to all possible ciphertext messages. The function consists of an algorithm and a set of parameters. The algorithm is usually made public, while the parameters are kept secret and are called the enciphering key. The recipient of the message is provided with a deciphering algorithm and a set of parameters, called the deciphering key, which enable her to decipher the message. Most commonly, the algorithms represent a set of computations in a large but finite group that is made public together with the enciphering keys, while the deciphering keys are known by the two users but are theoretically unobtainable by others.Below is one example of a public key cryptosystem.[1]

1.1 The RSA (Rivest, Shamir, Adleman) Cryptosystem:

The receiver of the message, A, will randomly choose two large primes p_A and q_A and a random number e_A that is invertible modulo $\phi(N)$ where $N = p_A * q_A$. A then computes d_A, where d_A is the inverse of e_A modulo $\phi(N)$.The enciphering key $K_{E,A} = (N, e_A)$ is then made public. The sender of the message, B, will than raise the contents of the message to the power e_A modulo N and send the result. Then A will raise the contents of the result to the power d_A modulo N to get the initial message (note that since $e_A * d_A = 1(mod\phi(N)$ we have that $x^{e_A * d_A} = x(modN))$. $K_{D,A} = (N, d_A)$ is called the deciphering key [1].
The RSA system is based primarily on the assumption that any intruder that intercepts $K_{E,A}$ will not be able to retrieve $K_{D,A}$ by finding the prime decomposition of N and thus the value of $\phi(N)$ and d_A fast enough. If one is able to factor the number N in fast enough time, one will be able to intercept the message. Other public key cryptosystems are based on essentially the same idea, but have different methods of encipherring, decipherring and exchanging information. In the following section I will present the Miller-Rabin algorithm which tests the primality of a number p and returns a proper factor if p is composite within$O(log^3(p) * loglog(p) * N(p))$, where$N(p)$ is the smallest quadratic non-residue modulo p.

2 Definitions, Theorems and Notation

2.1 Definitions and Theorems

Definition 2.1 ($f(n)$) *We say that an algorithm tests primality in $O(f(n))$ steps if there is a constant K and a deterministic Turing machine that will determine whether n is prime in less than $K * f(n)$ binary operations.*

Definition 2.2 ($N(n,q)$) *For prime q and integer n such that $q|\phi(n)$ we say that $N(n,q)$ is the lowest residue mod n that is not a q'th power mod n. By $N(n)$ we mean $N(n,2)$.*

Definition 2.3 ($(\frac{q}{p})$: Jacobi's symbol) *The value $(\frac{q}{p})$ for a prime p is 1 if q is a quadratic residue mod p and -1 otherwise.*

Definition 2.4 ($\chi(t)$: Character (mod k)) *We say $\chi(t) = \chi(t(modk))$ is a character mod k if it is a homomorphism from Z/kZ to the multiplicative group of complex numbers.*
$|\chi(t)| = 1$ for g.c.d.$(t,k) = 1$ and $|\chi(t)| = 0$ otherwise.If $\chi(t) = 1$ for all t such that g.c.d.$(t,k) = 1$, we say χ is the princpal character mod k. If $\chi(t)^2 = 1$ for all t such that g.c.d.$(t,k) = 1$, we say that χ is the quadratic character mod k.

Definition 2.5 ($L(s,\chi)$: The Dirichelet L-functions) $L(s,\chi) = \sum_{n=1}^{\infty} \frac{\chi(n)}{n^s}$ *where s is a complex number with $Re(s) > 1$ and χ is a character modulo k, for some integer k.*

Theorem 2.6 (Riemann) *For χ a non-principal character mod k,$L(s,\chi)$ can be analytically continued to the entire complex plane.*[2]

Theorem 2.7 (Extended Riemann Hypothesis, or E.R.H.) *Let χ be a non-principal character mod k. Then the only zeros of $L(s,\chi)$ lie on the line $Re(s) = 1/2$.* [2]

2.2 A Few Notations

Let $Exp_p(n)$ be the exponent of prime p in the prime decomposition of n.
Let $M(n)$ be the minimum number of binary operations needed to multiply or divide two integers of binary length n.
Let $\lambda\prime(n) = lcm(p_1 - 1, \ldots, p_m - 1)$ where the p's are the prime factors of n.

3 The Miller-Rabin Primality Test

3.1 Description of the Algorithm[3]

Let $f(n)$ be a function of n whose value will be specified later.
To test the primality of n perform the following:
(1) Check whether n is a perfect power. If it is, output composite and halt.
(2) For all a between 1 and $f(n)$ repeat the following tests. If some test holds for some a, output prime and halt.
(2i)$a|n$
(2ii)$a^{n-1} \not\equiv 1(modn)$
(2iii)$g.c.d.((a^(\frac{n-1}{2^k}) - 1, n) \neq 1orn$ for some k,$1 \leq k \leq Exp_2(n-1)$
(3)output prime and halt.

3.2 Proof of Algorithm[3]

Case A:
n is a perfect power. For all $k < log_2(n)$ check if n is a perfect k'th power.If so, output $log_k(n)$, composite, and halt.
This step takes $O(log_2(n))$ steps.

Case B:
$\lambda\prime(n)$ does not divide $(n-1)$. Then there exist primes p and q so that:$p|n, (p-1)$ does not divide $(n-1)$, $q|p-1$ and $Exp_q(p-1) > Exp_q(n-1)$

Lemma 3.1 *If a is any residue mod p that is not a q'th power, then $a^{n-1} \not\equiv 1(modn)$.*

Therefore, for this case, we have to go up to $a = N(p,q)$ to find that n is composite. If this case holds, test (2ii) will hold. We need $O(log_2(n)M(log_2(n)))$ steps to perform test (2ii), therefore we will need $O(log(n)M(log(n))) * N(p,q)$ steps for this case.After finding a, we output p, composite and halt.

Case C:
$\lambda\prime(n)|(n-1)$ and $Exp_2(p-1)$ is the same for all prime divisors p of n.

Lemma 3.2 *Take $1 < a < n$ such that $(\frac{a}{p})*(\frac{a}{q}) = -1$ for two prime divisors p, q of n. Then either a or $a^{\lambda\prime(n)/2} - 1$ has a nontrivial common divisor with n.*

From the lemma we infer that a will cause test (2iii) to fail.
Test (2iii) takes $O((log_2 n)^3 + (log_2 n)^2 * M(log_2 n))$ steps. We need to check

each a up to N(pq), therefore we have a running time of $O((log_2 n)^3 + (log_2 n)^2 * M(log_2 n)) * N(pq)$ for this case.After finding a, we output p, composite and halt

Case D:

$\lambda\prime(n)|(n-1)$ and $Exp_2(q-1) < Exp_2(p-1)$ for two prime divisors p and q of n.

Lemma 3.3 *If $0 < a < n$ so that $(\frac{a}{p}) = -1$, then either a or $a^{\lambda\prime(n)/2} - 1$ has a nontrivial common divisor with n.*

From the lemma we infer that a will cause test (2iii) to fail.
Test (2iii) takes $O((log_2 n)^3 + (log_2 n)^2 * M(log_2 n))$ steps. We need to check each a up to N(p), therefore we have a running time of $O((log_2 n)^3 + (log_2 n)^2 * M(log_2 n)) * N(p)$ for this case.After finding a, we output p, composite and halt.

Case E:
n is prime and the algorithm returns prime and halts

We can see that the dominant terms in the runtime are the ones for cases B, C, or D. According to the Schonhage-Strassen algorithm for multiplication, it is possible to create a machine that will perform multiplications in $M(k) = O(k * log_2 k * log_2(log_2(k)))$ binary operations, for integers of length up to k in binary representation. [3] Also, $gcd(m, n)$ can be computed in $O(log_2^2(max(m, n)) * M(log_2(max(m, n))))$. [4]
If we can insure that the value of $f(n)$ we choose is large enough for all the cases we have shown, we can guarantee that the algorithm will run in $O((log_2 n)^3 * log_2(log_2 n) * log_2(log_2(log_2(n))) * f(n))$ binary operations. Also note that, since the Jacobi symbol is multiplicative, if a fails our tests in any of the above cases, so will at least one of its prime divisors.Therefore we only need to look at prime values of a. From the prime number distribution theorem, the number of primes less than a is of the order $a/ln(a)$. It is reasonable to assume, and will be apparent from the following theorems, that the value of $f(n)$ will be small enough for practical purposes so that primes smaller than $f(n)$ could be extracted from a previously stored table. Consequently, we could take out another factor of $ln(f(n))$ from our running time bound. The programs that I wrote for this project will take about 5 minutes to generate samples of one million consecutive primes of size up to

several tens of billions, while running on our SUN-SPARC server.

3.3 Values for $f(n)$

By looking at the above cases, we can see that a sufficient condition for $f(n)$ so that the algorithm returns correct results is that $f(n)$ is larger than $N(p,q)$, $N(pq)$ for any possible divisors p,q of n.

Theorem 3.4 (Burgess)

$$N(p,q) = O(q * p^{1/4*e^{\frac{1}{2}}+\epsilon})$$
$$N(pq) = O((q * p)^{1/4*e^{\frac{1}{2}}+\epsilon}) \tag{3.1}$$

for any $\epsilon < 0$ [3].

Through manipulations that are beyond the scope of this paper, Burgess's result gives us a running time of approximately $O(n^{0.134})$ [3][5].

Assuming that the Extended Riemann Hypothesys holds, Ankeny has proved that, in the limit, the following result holds:

Theorem 3.5 (Ankeny, assuming E.R.H.)

$$N(n,q) = O(ln^2(n)) \tag{3.2}$$

Essentially, what this provides is a bound for $f(n)$ that is small enough to make the running time of The Miller-Rabin test of size $O((log_2 n)^3 * log_2(log_2 n) * log_2(log_2(log_2(n))) * f(n)/ln(f(n)))$, which, from Ankeny's theorem, means $O((log_2 n)^5 * log_2(log_2(log_2(n))))$.[3]
Unfortunately I was not able to retrieve an actual value for the constant corresponding to the $O(ln^2(n))$ bound on $N(n,q)$ from Ankeny's proof.This makes it impossible to know an exact value for $f(n)$. Empirical evidence that will be presented in the last section of this paper proves that we may not assume a value of $N(p)$ that is smaller than $97.88 * ln^2(n)$. What this means is that, for small enough primes, the running time of the Miller-Rabin test is comparable to $\sqrt{n}$, even if we assume the validity of E.R.H.. However, for large enough primes, this bound may prove to be very efficient, especially

for cryptosystems like R.S.A., where we know that the number to be tested is definitely composite, and further more, it is the product of two primes. It has also been proven by Graham and Ringrose that there exist infinitely many primes p such that $N(p) > C * ln(p)ln(ln(ln(p)))$ for some constant C. Montgomery has shown that, if we are to assume E.R.H.,there are infinitely many p with $N(p) > C * ln(p)ln(ln(p))$ for some constant C. Therefore, the bound given by Ankeny on $N(p, q)$ is not too far away from it's theoretical optimum.[5][7] A sketch of the proof given by Ankeny is presented in the next section.

3.4 Some Probabilistic Arguments on the Size of N(p)

Theorem 3.6 (Law of Quadratic Reciprocity) *Let p, q be two odd primes. Then* $(\frac{p}{q}) * (\frac{q}{p}) = (-1)^{(p-1)(q-1)/4}$.

Theorem 3.7 (Dirichlet) *Let* a, m *be integers such that* $gcd(a, m) = 1$. *Then there is an infinite number of primes of the form* $a + k * m$ *with* k *an integer.*

In fact, the stronger result holds: The prime numbers are, in the limit, roughly equidistributed among the residue classes a (mod m) such that $gcd(a, m) = 1$. This is implied by the following theorem:

Theorem 3.8 (Dirichlet) *Suppose a, m are integers such that* $gcd(a, m) = 1$. *Let* $\wp(a, m)$ *be the set of positive primes p such that* $p \equiv a(modm)$. *Then* $d(\wp(a, m)) = frac1\phi(m)$, *where* $d(\wp) = lim_{s \to 1} \frac{\sum_{p \in \wp} p^{-s}}{ln(s-1)^{-1}}$.

By looking at the above two theorems, we can infer that,in the limit, for a fixed prime q, the number of primes p such that $(\frac{p}{q}) = 1$ is about the same with the number of primes p for which $(\frac{p}{q}) = -1$, since the number of quadratic residues mod q is the same with the number of residues that are not quadratic. So the probability that $(\frac{p}{q}) = 1$ is about $\frac{1}{2}$.
But from quadratic reciprocity, and from the fact that, in the limit, the probability that a prime $p \equiv 1(\text{mod } 4)$ is $\frac{1}{2}$, we can infer that the probability that $(\frac{q}{p}) = 1$ is also about $\frac{1}{2}$.
Therefore, if q is the n'th prime number, then, in the limit, the probability that a prime p would satisfy $N(p) = q$ is $\frac{1}{2^n}$. Therefore, for a fixed q, which is the n'th biggest prime, we would have to go to about the $O(2^n)'th$ biggest prime to find p such that $N(p) = q$.

4 Ankeny's proof of Theorem 3.5, based on E.R.H.

4.1 More Notation

Let $\Gamma(s) = \int_0^\infty e^{-u} u^{s-1} du$.

Let $\xi(s,\chi) = (\frac{k}{\pi})^{s/2} L(s,\chi)\Gamma(\frac{s+a}{2})$, for $a = 0$ if $\chi(-1) = -1$ and $a = 1$ if $\chi(-1) = 1$

Let $\Lambda(n) = log(p)$ if n is a power of the prime p, and 0 otherwise.

From now on, by $\rho = \sigma + i\gamma$ we will mean a zero of the $L(s,\chi)$ function. By E.R.H., $0 < \sigma < 1$, and χ is considered to be a character mod k, for some integer k.

4.2 Ankeny's proof

Lemma 4.1 *Assuming E.R.H. for $X > 3$ and $T > 2$, we have:*

$$\sum_{0 \leq \gamma \leq T} X^{i\gamma} = O(\frac{Tlog(X)log(k(T+1))}{loglog(k(3+T))} + \frac{Tlog(k)}{log(X)} + Tlog(T)) \tag{4.1}$$

$$\sum_{0 \leq \gamma \leq T} X^{\rho}\Gamma(\rho) = O(X^{1/2}(\frac{log(X)log(k)}{loglog(k)} + \frac{log(k)}{log(X)})) \tag{4.2}$$

Lemma 4.2

$$\xi(s,\chi) = W\xi(s,\chi) \tag{4.3}$$

for $|W| = 1$.

Lemma 4.3 *From (4.3) we have:*

$$|L\prime/L(\frac{1}{3} + it, \chi)| = O(log(k)(|t| + 2)) \tag{4.4}$$

Theorem 4.4 *If $X > 3$, then*

$$\sum_p \chi(p)(logp)e^{-p/X} = O(X^{1/2}(\frac{log(X)log(k)}{loglog(k)} + \frac{log(k)}{log(X)})) + O(X^{1/3}logk) \tag{4.5}$$

Proof:
If $Re(s) > 1$ then the following holds:

$$|L\prime/L(s,\chi)| = -\sum_{n=2}^{\infty} \frac{\chi(n)\Lambda(n)}{n^s}$$

By Mellin's integral:

$$\sum_{n=2} \chi(n)\Lambda(n)e^{-n/X} = (\frac{-1}{2\pi i}) \int_{2-i\infty}^{2+i\infty} X^z (L\prime/L(z,\chi))\Gamma(z)dz$$

By moving the line of integration we get

$$\sum_{n=2} \chi(n)\Lambda(n)e^{-n/X} = \sum_{\rho} X^{\rho}\Gamma(\rho) + X^{1/3} \int_{\infty}^{\infty} X^{iy}(L\prime/L(1/3+iy,\chi))\Gamma(1/3+iy)dy$$

By applying Lemma 4.1 and 4.3 we obtain:

$$\sum_{n=2} \chi(n)\Lambda(n)e^{-n/X} = O(X^{1/2}(\frac{log(X)log(k)}{loglog(k)} + \frac{log(k)}{log(X)})) + O(X^{1/3}logk)$$

which implies our theorem.

Theorem 4.5 $N(k,q) = O((log(k)^2)$ *for any prime q.*

Proof:
Assume that $\chi(p) = 1$ for all primes q less than $c_1 X$ for some constant c_1 large enough. It is deduced that

$$|\sum_{p} \chi(p)log(p)e^{-p/X}| > c_2 X \tag{4.6}$$

for some constant c_2 depending on c_1.By Theorem 4.4 we obtain

$$X = O(X^{1/2}(\frac{log(X)log(k)}{loglog(k)} + \frac{log(k)}{log(X)})) + O(X^{1/3}logk) \tag{4.7}$$

which gives $X = O((log^2(k))$ q.e.d.

5 Empirical Data for the Value of N(p)

5.1 C code

For this project I wrote a set of C programs that were performing the following tasks:
primegen.c: find the first m (m given by user) primes starting with n (n given by user). This program was reasonably fast, allowing me to generate samples of 1 million consecutive primes in around 4-5 minutes on our SUN-Sparc server.
npq.c: This program takes a list of primes and outputs the list of values of N(p, q) for those primes p with $q|p-1$ in the list. Q can be user specified. For this program I wrote a series of functions that return the binary decomposition of a number and powers in a base k. Using the binary representation of the exponent, I was able to creata a power function that runs in time proportional to $O(log^2(exponent))$. This made the program fairly fast, allowing it to compute the values of $N(p)$ for a sample of 1 million primes in less than 15 seconds.
np_sort.c this is written for collecting statistical information from lists of N(p, q) values).

5.2 Data

Using primegen.c I generated files of prime numbers containing the first 1 million primes, the next 1 million primes and so on until the 9th to 10'th million primes. Next I created lists of 100000 consecutive primes starting at a value of 300 million 350 million and so on, up to a value of 800 million. The last two samples were 1 million consecutive primes starting at 1 billion and 2 billions.
In the appended tables there is a list of the largest values of N(p) that I have found for each of the prime number samples described above. A second chart shows the number of times that N(p) falls into range sizes of length 5000 for each of the mentioned lists. For instance for the 6th to 7th million greatest primes, we have 999989 numbers in the 0 to 5000 range and 11 numbers in the 10000 to 15000 range.
Finally, the last chart is a plot of the biggest values of N(p) for each sample, plotted against their corresponding $log^2 p$. The largest value of $N(p)$ that I found was 44927, corresponding to $p = 2018435329$. This implies that, if we were to assume Ankeny's result $N(p) < Klog^2(p)$, we cannot take a value of k that is less than 97.88. Assuming this bound, the Miller-Rabin test will have performance comparable to many less spectacular algorithms for

smaller values of n, but the bound gives us encouraging results as the size of n increases.

6 Notes For the Graders

The final version of this paper will include an additional section on elliptic curve cryptosystems and Lenstra's elliptic curve factoring algorithm. Also, more data will be collected and a complete analysis of the behaviour of $N(p,q)$ will be provided for values of q other than 2. Finally, similar analysis will be performed on the value of the first prime p for which $a_p = 0$, for elliptic curves of a specified family or with given bounds on the coefficients. This will be included in the full report on elliptic curves to be published as a result of this seminar.

7 Bibliography

[1]: Koblitz, N.: A course in Number Theory and Cryptography, Springer Verlag, New York 1987

[2]: Ellison, W. F.: Prime Numbers, Wiley Interscience, New York 1985

[3]: Miller, G.L.: Riemann's Hypothesis and Tests for Primality, Journal of Computer and System Sciences no.13 (1976),pp. 330-317

[4]: Silverman, J.H. Tate, J.: Rational Points on Elliptic Curves, Springer Verlag, New York, 1992

[5]: Montgomery, H.L.: Topics in Multiplicative Number Theory, Lecture Notes in Mathematics no. 227 (1971),pp.117-130

[6]: Ankeny, N.C.: The Least Quadratic Non-Residue, Annals of MAthematics Vol.55, No. 1 (1952), pp.65-72

[7]: Graham, S.W. Ringrose, C.J., Lower Bounds for the Least Quadratic Non-Residues,Analytic Number Theory, Progr. Math., 85, Boston, MA, (1990), pp.269-309