Is There A Shortage of Primes for Cryptography?

Samuel S. Wagstaff, Jr.
Center for Education and Research in Information Assurance and Security
Department of Computer Sciences, Purdue University
West Lafayette, IN 47907-1398, USA. (Email: ssw@cerias.purdue.edu)

(Received Oct. 12, 2005; revised and accepted Nov. 23, 2005)

Abstract

Cryptographic algorithms often prescribe the use of primes whose length in bits is a power of 2. Recently, we proved that for $m > 1$, there is no prime number with 2^m significant bits, exactly two of which are 0 bits. Here we generalize this theorem to impose many more restrictions on primes whose length in bits is a power of 2. No similar restrictions apply to primes of other lengths. We consider whether the restrictions on primes with length 2^m bits are so great that one should choose other lengths for primes to be used in cryptography.

Keywords: Binary representation, cryptography, primes

1 The Problem

Many cryptographic algorithms need to choose secret random large primes. Quite often, they specify that the length in bits of the primes be a power of 2. For example, the following table, adapted from Figure 1.18 on page 38 of [3], gives the menu of cryptographic algorithms used by the Secure Socket Layer. When you send your credit card number over the Internet to buy something at a business web site, the number is encrypted and signed using one or two of these algorithms. Your computer and the company's computer negotiate which algorithms from the menu are most suitable. Note that the key lengths, meaning the number of bits in a prime number (or the product of two primes, each of half the key length), in Table 1 are either 512 or 1024 bits, both powers of 2. Key lengths of 2048 bits are proposed for future ciphers. So far as I know, the only reason for choosing a power of 2 for the length of a cryptographic key is that a key of this size just fits into one or more standard hardware registers.

The primes chosen for most of these algorithms must be secret. Are there enough primes available for this purpose? If there were too few primes, then an enemy might be able to guess your prime. The prime number theorem says that the number $\pi(x)$ of primes $\leq x$ is approximately $x/\ln x$. Accepting this approximation naively, we would conclude that the number of 512-bit primes is about

$$\pi(2^{512}) - \pi(2^{511}) \approx \frac{2^{512}}{\ln 2^{512}} - \frac{2^{511}}{\ln 2^{511}} \approx 18.85 \times 10^{150} \qquad (1)$$

so there would be plenty of 512-bit primes to go around. But the prime number theorem really says that $\lim_{x\to\infty} \pi(x)/(x/\ln x) = 1$. This means that the percentage error in the approximation $\pi(x) \approx x/\ln x$ goes to zero as x goes to ∞. But how fast does the percentage error go to zero? Just how accurate is the estimate in (1)?

About 150 years ago, Chebyshev proved Bertrand's postulate, which says that there is at least one prime number between x and $2x$ for every positive integer x. In particular, there must be at least one prime between every two consecutive positive integer powers of 2. But one is not enough. For cryptographic security, we need to have so many primes available that no one could perform an exhaustive search for the one we used.

Consider another factor which may restrict the number of available primes even more than the length in bits. Some cryptographic algorithms, including many in Table 1, use a prime as an exponent or a modulus. There are special techniques (see Chapter 9 of [1]) for performing exponentiation and remaindering modulo a large prime which run faster when the binary representation of the exponent or the modulus either has few 1 bits or has few 0 bits. These techniques should be used when possible to make the cryptographic algorithm run as fast as possible. This question prompted us [5] to study primes with either few 1 bits or few 0 bits. Are there enough of these restricted primes?

2 Two Theorems that Cause the Shortage

In [5] we computed the number of primes p whose binary representation has n significant bits in each of these four categories: (a) p has exactly three 1 bits, (b) p has exactly four 1 bits, (c) p has exactly one 0 bit, and (d) p has exactly two 0 bits, for $n \leq 100$. A simple heuristic argument given in [5] predicts that, for each n, the average number of primes in categories (a) and (c) each should be about 2 or 3. The same heuristic argument predicts linear growth on average as a function of n for the number of primes in categories (b) and (d). The following table summarizes

Table 1: Cryptographic algorithms used by the Secure Socket Layer

Algorithm	Key Length in Bits
RSA encryption	512
RSA encryption	1024
DH agreement	512
DH agreement	1024
RSA signature	512
RSA signature	1024
RSA verification	512
RSA verification	1024
DSS signature	512
DSS signature	1024
DSS verification	512
DSS verification	1024

Table 2: Number of n-bit primes in each of four categories

n	(a)	(b)	(c)	(d)	n	(a)	(b)	(c)	(d)
3	1	0	1	0	15	2	16	0	18
4	2	0	2	0	16	4	18	3	0
5	1	2	2	1	17	1	25	2	21
6	2	2	3	2	18	3	15	8	21
7	3	5	0	4	30	2	31	4	45
8	3	4	4	0	31	8	51	0	47
9	0	10	4	9	32	6	39	7	0
10	4	6	3	5	33	0	37	5	69
11	2	13	1	14	62	2	81	3	50
12	3	11	5	4	63	1	106	0	110
13	2	9	1	16	64	2	56	3	0
14	2	16	4	9	65	0	102	1	108

the results reported in [5]. Table 2 corrects errors in the count of 30-bit primes in categories (c) and (d) in that paper. The author thanks Jason Gower for finding these errors.

The results generally follow the heuristic predictions except for some surprising 0's in the population of primes in category (d). These occur when n is a power of 2. Most cases of this phenomenon are explained in [5] by the following theorem.

Theorem 1. *For all $m \geq 1$, there is no prime number whose binary representation has precisely 2^m significant bits, exactly two of which are zero bits. In other words, there is no prime number of the form $2^{2^m} - 2^i - 2^j - 1$, where $1 \leq i < j \leq 2^m - 2$.*

Proof. Write $N = 2^{2^m} - 1 - (2^i + 2^j)$ with $1 \leq i < j \leq 2^m - 2$. Let $j - i = 2^k e$, where e is odd. We show that $d = 2^{2^k} + 1$ is a proper divisor of N. First note that $2^k \leq j - i \leq 2^m - 3$, so $k < m$ and

$$1 < d = 2^{2^k} + 1 < 2^{2^m - 2} - 2 < N$$

when $m \geq 3$. (Check the cases $m = 1$ and $m = 2$ separately.) Clearly, $2^{2^k} \equiv -1 \bmod d$. Since $k < m$, we have $2^{2^m} \equiv 1 \bmod d$, and so d divides $2^{2^m} - 1$. Write

$$2^i + 2^j = 2^i(2^{j-i} + 1) = 2^i(2^{e2^k} + 1).$$

Since $2^{2^k} \equiv -1 \bmod d$, and e is odd, we have $2^{e2^k} \equiv -1 \bmod d$, and d divides $2^{e2^k}+1$. Therefore, d divides $2^i + 2^j$ and hence also N. It follows that N is not prime. □

Of course, one would not choose a 512-bit or 1024-bit prime in any of the four categories for cryptographic use, because it would be too easy to guess. It would be better to choose a secret random prime having at least several 1 bits and at least several 0 bits. However, that might not solve the shortage because one can extend Theorem 1 as follows.

Theorem 2. *Suppose $N = \sum_{i=0}^{2^m-1} b_i 2^i$, where each $b_i \in \{0, 1\}$ and $b_0 = b_{2^m-1} = 1$. Suppose the number of zero bits b_i is a positive even number $2z$. Suppose there is a nonnegative integer k and a pairing of the zero bits of N into z pairs so that the difference between the subscripts in each pair is exactly divisible by 2^k. Then N is composite.*

Proof. We can write

$$N = 2^{2^m} - 1 - \sum_{\text{the pairs } (i,j)} (2^i + 2^j).$$

Let (i, j) be one of the pairs of subscripts of 0 bits (b_i and b_j). Say $i < j$. Then $j - i = 2^k e$ for some odd e. As in the proof of Theorem 1, $d = 2^{2^k} + 1$ divides $2^i + 2^j$. But $d = 2^{2^k} + 1$ also divides $2^{2^m} - 1$, and therefore d is a proper divisor of N, so N is composite. □

If the pairing mentioned in the theorem is possible with $k = 0$, then $d = 3$ divides N. Likewise, if $k = 1$, then $d = 5$ divides N. It is possible for N to be divisible by 3 or 5 even if no pairing of 0 bits is possible, as the 8-bit examples $171 = 10101011_2$ and $145 = 10010001_2$ show.

3 Computing $\pi(2^m)$

How often does Theorem 2 apply? Does the theorem noticeably change the population count for the primes of these special lengths? It is not possible, with current knowledge, to determine the exact number of 512- or 1024-bit primes. Table 3 shows most of the known values of $\pi(2^m)$. Marc Deléglise, who with J. Rivat has computed $\pi(x)$ exactly [2] for certain large x, kindly computed the large numbers in this table.

Using this data, we tabulated the ratios $r(m) = \pi(2^m)/\pi(2^{m-1})$ and $s(m) = (\pi(2^m) - \pi(2^{m-1}))/(\pi(2^{m-1}) - \pi(2^{m-2}))$. By the prime number theorem, both $r(m)$ and $s(m)$ converge to 2 as $m \to \infty$. We hoped to discover some variation in these numbers when m passes through a power of 2. As you can see from

Table 3: Values of $\pi(2^m)$ for $4 \leq m \leq 45$

m	$\pi(2^m)$	m	$\pi(2^m)$	m	$\pi(2^m)$
4	6	18	23000	32	203280221
5	11	19	43390	33	393615806
6	18	20	82025	34	762939111
7	31	21	155611	35	1480206279
8	54	22	295947	36	2874398515
9	97	23	564163	37	5586502348
10	172	24	1077871	38	10866266172
11	309	25	2063689	39	21151907950
12	564	26	3957809	40	41203088796
13	1028	27	7603553	41	80316571436
14	1900	28	14630843	42	156661034233
15	3512	29	28192750	43	305761713237
16	6542	30	54400028	44	597116381732
17	12251	31	105097565	45	1166746786182

Table 4, these ratios vary smoothly, with no saltatory behavior near a power of 2.

It would be possible to compute $\pi(2^m)$ exactly for m up to about 64 or 65 (or even 70), but this calculation would be close to the limit of modern algorithms and computers. Thus, with much effort, we could find $r(m)$ and $s(m)$ for m near 64, the first power of 2 beyond Table 4. However, there is no way we could extend the calculations to 128, the next power of 2. How can we possibly settle the question of whether there are enough 512-bit and 1024-bit primes to provide safe cryptography?

4 The Answer

Rosser and Schoenfeld [4] proved explicit inequalities for $\pi(x)$ and other functions related to prime numbers. For example, they proved that $x/\ln x < \pi(x) < 1.25x/\ln x$ for all $x \geq 114$. Corollary 3 of Theorem 2 of [4] has just the inequality we need. It says that

$$0.6x/\ln x < \pi(2x) - \pi(x) < 1.4x/\ln x$$

for all $x \geq 20.5$. If we take $x = 2^{m-1}$, we find that

$$\begin{aligned} \pi(2^m) - \pi(2^{m-1}) &> 0.3 \cdot 2^m/((m-1)\ln 2) \\ &\approx 0.43 \cdot 2^m/(m-1) \end{aligned}$$

for $m \geq 6$. This shows that the number of 512-bit primes is

$$\pi(2^{512}) - \pi(2^{511}) > 0.3 \cdot 2^{512}/(511 \ln 2) \approx 11.36 \times 10^{150},$$

which is smaller than our first approximation (1) of 18.85×10^{150}, but still large enough so that there will be plenty of them for safe cryptography. Just don't try to choose a 512-bit prime with exactly two 0 bits.

This answer ignores the number of 0 bits or 1 bits in the 512-bit primes. What if we wish to speed our cryptographic algorithm by choosing 512-bit primes having only a few 0 bits (but more that two of them) or only a few 1 bits? For example, there are $\binom{510}{10}$ odd numbers with 512 significant bits and exactly ten 0 bits. How many of them are prime? Theorem 2 forces many of these numbers to be composite. I don't know how many primes there are of this type. But one can thwart the action of Theorem 2 simply by specifying that a candidate prime have an *odd* number of 0 bits. There should be plenty of 512-bit primes having exactly nine or exactly eleven 0 bits. These are the ones to use for secure and efficient cryptography. An alternative simple solution to the possible prime shortage would be to change the requirements of cryptographic algorithms to prescribe that the length in bits of secret primes not be a power of 2. Then Theorems 1 and 2 would not apply.

Table 4: Values of the ratios $r(m)$ and $s(m)$ for $7 \leq m \leq 36$

m	$r(m)$	$s(m)$	m	$r(m)$	$s(m)$
7	1.722222	1.857143	22	1.901839	1.907102
8	1.741935	1.769231	23	1.906297	1.911242
9	1.796296	1.869565	24	1.910567	1.915277
10	1.773196	1.744186	25	1.914597	1.919024
11	1.796512	1.826667	26	1.917832	1.921369
12	1.825243	1.861314	27	1.921152	1.924769
13	1.822695	1.819608	28	1.924211	1.927532
14	1.848249	1.879310	29	1.926940	1.929891
15	1.848421	1.848624	30	1.929575	1.932418
16	1.862756	1.879653	31	1.931940	1.934483
17	1.872669	1.884158	32	1.934205	1.936635
18	1.877398	1.882817	33	1.936321	1.938587
19	1.886522	1.896921	34	1.938284	1.940380
20	1.890413	1.894801	35	1.940137	1.942112
21	1.897117	1.904646	36	1.941890	1.943756

Acknowledgement

This work was supported in part by grants from the CERIAS Center at Purdue University and from the Lilly Endowment Inc.

References

[1] R. Crandall and C. Pomerance, *Prime Numbers: A Computational Perspective*, Springer-Verlag, New York, 2001.

[2] M. Deléglise and J. Rivat, "Computing $\pi(x)$: the Meisel, Lehmer, Lagarias, Miller, Odlyzko method," *Mathematics of Computation*, vol. 65, pp. 235-245, 1996.

[3] E. Rescorla, *SSL and TLS, Designing and Building Secure Systems*, Addison-Wesley, Boston, Massachusetts, 2001.

[4] J. B. Rosser and L. Schoenfeld, “Approximate formulas for some functions of prime numbers,” *Illinois Journal of Mathematics*, vol. 6, pp. 64-94, 1962.

[5] S. S. Wagstaff, Jr, “Prime numbers with a fixed number of one bits or zero bits in their binary representation,” *Experimental Mathematics*, vol. 10, pp. 267-273, 2001.

Samuel S Wagstaff, Jr received a B.S. from the Massachusetts Institute of Technology, Cambridge, Massachusetts, USA and a Ph.D. from Cornell University, Ithaca, New York, USA, both in Mathematics. He is a professor of Computer Science at Purdue University, West Lafayette, Indiana, USA. Before coming to Purdue, he taught at the Universities of Rochester (New York), Illinois (Urbana) and Georgia (Athens). He worked at the Institute for Advanced Study, Princeton, New Jersey, in 1971–1972. He is a member of the AMS, MAA and UPE. His research interests include primality testing, integer factorization, cryptography, secure patch distribution and watermarking. He is the leader of the Cunningham Project, which factors numbers of the form $b^n \pm 1$. He has supervised three Ph.D. theses and published five books and more than 60 research papers. An algorithm that he and R. Baillie invented and published in 1980 was selected as ANSI Standard X9-80 for choosing industrial-grade primes for use in cryptography. It is used worldwide as part of the Secure Socket Layer.