THE MAGIC WORDS ARE SQUEAMISH OSSIFRAGE

Brian Hayes

You are given two integers, a and b, and asked to compute their product, $ab = c$. An algorithm for this task is taught in the early primary grades. For those of us who were daydreaming in class that day, a computer implementation of the algorithm yields an answer in microseconds, even if a and b are rather large numbers, say 60 or 70 decimal digits. Now suppose you are given the number c and asked to discover the two factors a and b, which you may assume are prime numbers (that is, they have no factors of their own, apart from 1 and themselves). This is a much harder assignment. If a and b are in the 60-digit range, so that c has more than 120 digits, finding the factors is definitely not elementary-school homework.

The dramatic asymmetry between multiplication and factorization is the basis of an important cryptographic system: the RSA public-key cryptosystem, named for the initials of its inventors, Ronald L. Rivest of the Massachusetts Institute of Technology, Adi Shamir of the Weizmann Institute in Israel and Leonard M. Adleman of the University of Southern California. When a message is encoded in the RSA system, the legitimate recipient, who knows the decryption key, can recover the original text through a computational process roughly as easy as multiplying several large numbers. For an eavesdropper trying to break the code, however, decrypting the message is thought to be as hard as factoring a very large composite integer into its two large prime factors.

In 1977 the RSA system was brought to public attention in Martin Gardner's "Mathematical Games" column in *Scientific American*. The column included a short message encoded with a 129-digit key, and offered a prize of $100 to anyone who could decrypt the message, presumably by factoring the published number. The challenge stood for 17 years. A campaign to factor the number, which has come to be known as RSA-129, was launched last August, and it came to a successful conclusion after eight months of effort by more than 600 contributors (and their computers). The factors were announced on April 26, along with the decoded message: "The magic words are squeamish ossifrage." (An ossifrage is a lammergeier, a "bone-breaking" vulture, which seems to suggest the character of the factoring problem, but Rivest says the secret text was chosen at random.)

The factoring of RSA-129 is interesting for at least three reasons. First, it calls attention to a substantial improvement in the art and the technology of factoring, and to progress in computational number theory more generally. Second, because the factoring was done by an ad hoc network of widely dispersed volunteers, it offers lessons on the logistics of organizing large-scale cooperative computing projects. Third, the success of the factoring effort emphasizes the strong connections that have grown up between cryptology and areas of discrete mathematics and theoretical computer science. It is well known that modern cryptology rests on a foundation of ideas from these fields, but it is not so widely appreciated that cryptological problems have stimulated deep theoretical work.

The Trials of Division

The most direct method of factoring is another grammar-school algorithm: division. Given a number n to be factored, you first try dividing it by 2, which is the smallest of all primes. If the division yields a remainder of 0—or in other words if 2 divides n exactly—then you have found a factor. Actually you have found a pair of factors: a small one (2) and a large co-factor ($n/2$). If 2 doesn't work, try 3, then 5, then 7, then 11, then each of the larger prime numbers in sequence. If you reach a prime greater than the square root of n without finding a factor, you have proved that n is prime.

Trial division is a brute-force algorithm—a description that always suggests a hint of scorn. (Clever people shouldn't need brute force.) Factoring a 100-digit number by trial division could require roughly 10^{50} operations, which could not possibly be completed even by an ad hoc network of all the computers on earth. Nevertheless, trial division should not be dismissed as totally useless. As a matter of fact, it is the best known factoring algorithm for almost all integers! To be more specific, it is the best method of finding a factor when n happens to have a small factor, which is true of

Brian Hayes, a former editor of American Scientist, *has been writing about computing for more than 10 years. Address: 211 Dacian Avenue, Durham, NC 27701. Internet: bhayes@mercury.interpath.net.*

most n's. After all, half of all integers are divisible by 2, a third of them are divisible by 3, a fifth by 5, and so on. If 100-digit numbers are selected at random, more than 95 percent will have at least one factor smaller than a million. On the other hand, numbers used as keys in RSA cryptography are not selected at random; they are known to have only very large factors, so that for these selected integers trial division is certain to fail.

One response to this problem is simply to work from the "other end" of the number. Instead of starting with 2 and progressing toward the square root of n, start with the largest integer less than the square root and work downward toward 2. If a 100-digit n is created by multiplying two 50-digit primes, then the prime factors must be comparatively close to the square root of n. By starting with the square root, it seems one must quickly discover the factors. But there is a fallacy in this reasoning. Knowing that two numbers are both 50 digits long offers little help in finding them. Among all integers with 50 or fewer decimal digits, 90 percent have exactly 50 digits.

Pierre de Fermat, the great 17th-century number theorist, suggested another factoring strategy that begins with the square root of n. Fermat observed that if n could be expressed as the difference between two squares, the factorization would be known directly. If n is equal to $x^2 - y^2$, then n is also equal to $(x + y)(x - y)$, and so $(x + y)$ and $(x - y)$ are factors of n. For example, the composite number 187 can be written as the difference $14^2 - 3^2$, or $196 - 9$, and the numbers $14 - 3 = 11$ and $14 + 3 = 17$ are indeed factors of 187.

To turn Fermat's idea into a factoring algorithm, all that is needed is a procedure for finding suitable values of x and y. The straightforward solution begins by setting x equal to the smallest integer greater than or equal to the square root of n and calculating $x^2 - n$. If the result is a perfect square, a pair of factors has been found; otherwise, increase x by 1 and try again. This tactic works well if n happens to have two nearly equal factors, but in general the number of operations needed is no better than it is with trial division.

The trouble with Fermat's algorithm is that for each pair of factors in n there is only one pair of positive integers satisfying $x^2 - y^2 = n$; finding those integers is not necessarily any easier than finding the factors directly. A way around this limitation derives from the work of the French mathematicians Adrien-Marie Legendre (in the 1820s) and Maurice Kraitchik (a century later). The idea is to search for values of x and y that satisfy not a quadratic *equation* but a quadratic *congruence* of the form $x^2 \equiv y^2$ modulo n. Thus one looks for values of x and y whose squares leave the same remainder when they are divided by n. When this relation holds, $x + y$ and n must have a factor in common (and the co-factor is common to $x - y$ and n); the factor is found by calculating the greatest common divisor of $x + y$ and n, for which there is an efficient algorithm. In some cases the factor found is a trivial one—either 1 or n itself—but if n is not prime, there is at least a 50–50 chance of finding a nontrivial factor. What is more, for a large composite n there are many x, y pairs that satisfy the congruence, so that they are potentially easier to find than the unique solutions to the equation $x^2 - y^2 = n$.

Consider again the factoring of $n = 187$. The values $x = 14$ and $y = 3$ discovered through Fermat's method satisfy the congruence formula, since $14^2 = 196$ and $3^2 = 9$ are both congruent to 9 modulo 187. But the values $x = 25$ and $y = 8$ also work in the congruence, since 625 and 64 both leave a remainder of 64 when divided by 187. The sum $25 + 8 = 33$ has the factor 11 in common with 187; the difference $25 - 8 = 17$ gives the co-factor. The same result could be obtained with $x = 72$ and $y = 38$, with $x = 535$ and $y = 246$ and with innumerable other pairs. Stumbling onto any of these solutions is enough to factor the number. Most of the hot new factoring algorithms rely in one way or another on a search for quadratic congruences.

3490529510		3276913299	11438162575788886766
8476509491		3266709549	92357799761466120102
4784961990	×	9619881908	18296721242362562561
3898133417		3446141317	84293570693524573389
7646384933		7642967992	78305971235639587050
8784399082		9425397982	58989075147599290026
0577		88533	879543541

Figure 1. Prime factors of the 129-digit number known as RSA-129.

Sieving for Smooth Numbers

The first of the modern factoring algorithms was a continued-fraction method, devised in the 1930s by D. H. Lehmer and R. E. Powers but not put into practice until 1970 by Michael A. Morrison and John Brillhart. In 1974 the British mathematician John M. Pollard introduced the $p - 1$ algorithm, which produces a quick factorization of $n = pq$ if either $p - 1$ or $q - 1$ happens to be an easily factored number; this works surprisingly often for numbers of 20 or 30 digits. Pollard also created a second algorithm, the ρ method, which works over a similar range. In 1986 Hendrik W. Lenstra, Jr., of the University of California at Berkeley devised a quite different factoring method based on elliptic curves; it works well for finding factors up to about 40 decimal digits, regardless of the size of n. When the smallest factor is still larger, the elliptic-curve method has been surpassed by the quadratic sieve, an algorithm published in 1981 by Carl Pomerance of the University of Georgia; the quadratic sieve will be described in greater detail below. In the meantime, yet another algorithm of Pollard's, the number-field sieve, appears ready to supplant the quadratic sieve in dealing with still larger numbers.

Most of these algorithms draw on a common body of underlying principles and techniques. As noted above, quadratic congruences are central to several of them. Here are some other shared ideas:

First, all of the recent algorithms have an ele-

ment of randomness in their operation. Trial division is a fully deterministic procedure: Nothing is left to chance, and if a trial-division program terminates without finding a factor, you can be sure that no factor exists. The newer algorithms cannot offer such a guarantee.

Second, some of the algorithms adopt a strategy that seems perverse and evasive: Given a number that is too hard to factor, they set about searching for other numbers that are especially easy to factor, namely numbers composed entirely of quite small primes. Numbers of this kind are said to be "smooth," because there are no wide gaps or jagged peaks in their list of factors. The hard part, of course, is finding a way to apply information gained from the smooth numbers to the single nonsmooth number whose factors are needed.

Third, several of the algorithms make use of "sieves," which bring a kind of mass-production technology to number theory. The most famous number sieve is that of Eratosthenes, an Alexandrian who lived circa 250 B.C.E. To strain some numbers through Eratosthenes' sieve, start by writing down the integers from 2 to some limit. Circle the 2 and cross off every second number after it, then circle the 3 and cross off every third number; at this point 4 has already been crossed off, so circle the 5 and cross off every fifth number. Continue in this way until there are no numbers left unmarked, at which point the circled numbers are the primes. In factoring, a sieve is often used in reverse, so that the numbers of interest are not the primes (which never get crossed off) but the smooth numbers (which are crossed off many times).

The algorithm used to factor RSA-129 was a version of the quadratic sieve. It incorporates all of the features mentioned above: a focus on quadratic congruences, randomness, a search for smooth numbers and a sieve. I shall describe it in a simplified form, but even without all the latest embellishments, it is an admirably ingenious algorithm.

The Quadratic Sieve

To factor n with a quadratic sieve, first choose some integers r not too far from the square root of n, then for each r calculate $Q \equiv r^2 \bmod n$. Now try to factor each of the Q's, using only the factors in a finite set of small primes called the factor base. Since the primes in the factor base are small, the factoring will be easy. (The best way to do it will be described below.) A factored value of Q can be represented as a list of the primes in the factor base with their accompanying exponents, which indicate how many times each factor appears in Q. For example, if the factor base consists of the primes 2, 3, 5 and 7, the number 504 can be represented as $2^3 \times 3^2 \times 5^0 \times 7^1$.

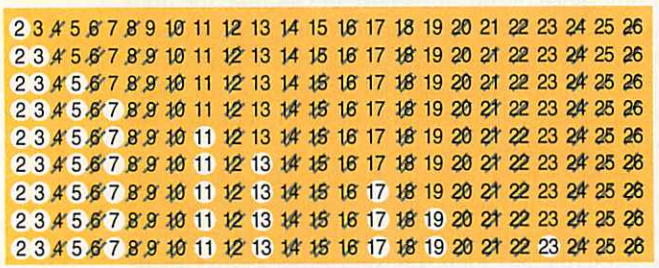

Figure 2. The sieve of Eratosthenes separates primes from composites.

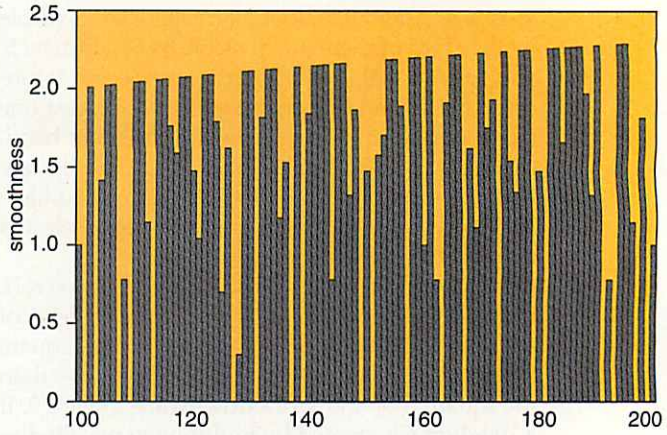

Figure 3. "Smooth" numbers are those with many small factors.

As you are accumulating factored Q values, you might keep an eye out for any list of factors in which all the exponents are even; if such a number should turn up, it is a perfect square and hence a solution to the congruence $x^2 \equiv y^2 \bmod n$. An example is 400, with the factorization $2^4 \times 3^0 \times 5^2 \times 7^0$, which is the square of 20. Finding a square Q immediately leads to a factor of n. Unfortunately, when n is large, the odds of finding a factor in this way are minuscule.

What if you don't get lucky and find a square Q? Suppose you have two Q's that would be perfect squares except that one includes 3^1 as a factor and the other includes 3^3? If you multiply these Q values—and thus add the exponents of their prime factors—the product includes 3^4 among its factors, and all the other exponents also remain even. What is more, if you also multiply the corresponding r's, the quadratic congruence will still be satisfied. In other words, if you cannot find a square Q, you can create one out of two Q's that are "almost" square. Indeed, you can combine not just two Q's but any number of them to convert odd-order factors into even-order ones. If you have enough Q's to work with, there is sure to be a combination of them that is a perfect square; furthermore, there is an efficient procedure for finding that combination. How many Q's are enough? You need at least as many Q's as there are primes in the factor base.

The procedure for constructing a square combination of Q's is an exercise in linear algebra. Assemble all the data into a matrix with one column for each prime and one row for each Q; a matrix element at column i and row j is the exponent to which the ith prime is raised in Q_j. In combining Q's to form a perfect square, all that matters is whether an exponent is odd or even, and so each exponent can be represented by a single bit of information—0 for even, 1 for odd. Two rows of the matrix are combined by adding them modulo 2.

The selection of which rows to add can be made systematically through the technique called Gaussian elimination, which eventually generates a row of all 0's. If you keep track of the operations that lead to this result, you can then go back to the actual Q and r values that the matrix rows represent and carry out the corresponding multiplications. The resulting numbers produce a factor.

How are all those Q's and r's found in the first place? That's where the sieve comes in. The naive approach would be to select r's either sequentially or randomly, calculate $Q \equiv r^2 \bmod n$, and try factoring Q by trial division. If Q is smooth, the factoring will go quickly; if not, you can simply abandon the attempt and skip to the next r. A sieve improves on this method by testing thousands of Q's at once. When a range of Q values is set up in an appropriate arithmetic progression, divisibility by the primes in the factor base is determined merely by counting off and crossing out, as it is in the sieve of Eratosthenes. In practice, rather than simply counting how many primes divide a given Q, the algorithm sums the logarithms of the successful divisors, thereby giving greater weight to larger factors. Each Q for which this sum exceeds a threshold is likely to be smooth, and so it can be factored quickly by trial division.

Factoring by E-mail

Factoring is one of the most computationally demanding tasks in mathematics, and yet it has traditionally been done on a shoestring budget. Factorers have scavenged idle time on computers; they have offered to test and exercise the hardware of new machines; they have rescued discarded computers and kept them running in the basement. There is also a tradition of homebrewing, seen most notably in the marvelous mechanical number sieves of D. H. Lehmer, built out of punched paper tapes or bicycle chains.

In recent years the most popular approach to low-budget factoring has been exploiting the idle nights and weekends of workstations that are left on 24 hours a day but often have nothing better to do than run a screen-saver program. The pioneer of this technique has been Robert Silverman of the MITRE Corporation, who has factored more than a thousand large numbers with a few dozen workstations on a local-area network. Arjen K. Lenstra of Bellcore and Mark S. Manasse of the Digital Equipment Corporation have cast a wider net, writing software that allows a factoring job to be distributed among distant workstations whose only connection is electronic mail.

The factoring of RSA-129 was the largest yet of the distributed factoring projects; indeed, it is probably one of the largest single computations ever undertaken. The four principal organizers were Arjen Lenstra, Paul Leyland of the University of Oxford, Derek Atkins of MIT and Michael Graff of Iowa State University. They employed the Lenstra-Manasse software, tuned for the 129-digit number and adapted to run on a wider variety of computers. The call for volunteers last August elicited help from 600 people in 24 countries, many of whom were able to contribute time on multiple machines.

The algorithm chosen for the project is known as the multiple-polynomial quadratic sieve, in which the single congruence $Q \equiv r^2 \bmod n$ is replaced by a large set of polynomial relations. The multiple-polynomial variant is particularly well adapted to distributed computing, since each machine can work on its own polynomial. Another enhancement allows the program to accept Q values that do not quite factor fully within the factor base but have either one or two larger prime factors. Such single and double "partials" can be combined in various ways to yield full congruences.

All through the fall and winter, congruences poured into a central collecting site at MIT. The factor base consisted of 524,338 primes, and so at least as many full congruences were needed to guarantee a factorization. By the end of March the threshold had been reached. The far-flung sievers had returned more than 100,000 full congruences and 8 million partials; the latter were combined to create the equivalent of another 425,000 fulls.

This was the end of the networked phase of the project, but there was still a substantial computation ahead: The analysis of a matrix with 524,338 columns and 569,466 rows. The Gaussian elimination took 45 hours on a MasPar computer with 16,384 processors. The first three factorizations found were trivial ones, but the fourth solution yielded the factors shown in Figure 1.

Ignorance and Security

The factoring of RSA-129 was not an isolated landmark but rather a milestone along a route where mathematicians and computer scientists have been making steady progress. In 1990 RSA Data Security—the company formed by Rivest, Shamir and Adleman—published a list of challenge numbers designated RSA-100, RSA-110, RSA-120 and so forth up to RSA-500 (the names indicate the length of the number in decimal dig-

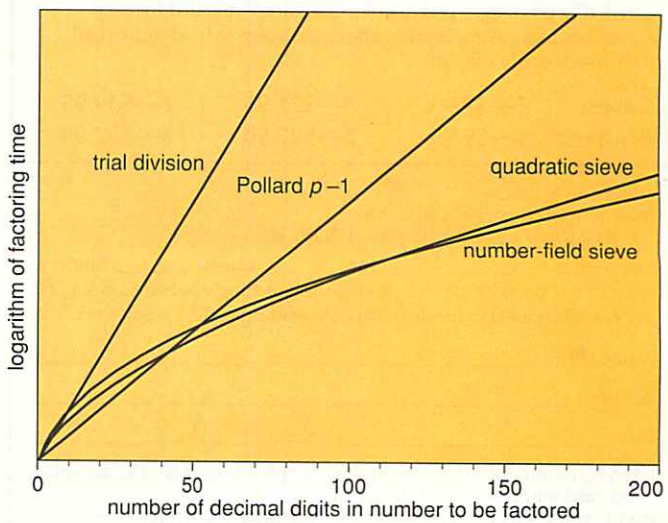

Figure 4. Approximate running time of some factoring algorithms.

its). The first three of these numbers have now been factored, all by quadratic-sieve methods and all by Arjen Lenstra either alone or in collaboration with others. Now with RSA-129 factored, RSA-130 is obviously in jeopardy.

Do these accomplishments put the security of the RSA cryptosystem in doubt? Both the cryptologists and the number theorists think not. RSA Data Security recommends a minimum key size of about 150 digits, which is still beyond the practical range of the programs employed in the RSA-129 effort. A rule of thumb suggests that adding 10 decimal digits to the length of a number makes it from five to 10 times as hard to factor. By this crude measure, factoring RSA-150 by the methods and machines applied to RSA-129 would take more than a decade, though less than a century.

Of course methods and machines will continue to improve, but in the contest between key-maker and key-breaker, the advantage clearly lies with the maker. It is easier to add another 10 digits to the key than it is to make a factoring algorithm 10 times faster. Keys as long as 300 digits are in common use; factoring numbers this large is beyond any forseeable extrapolation of present techniques.

On the other hand, there is reason for caution in the long run. The true difficulty of factoring remains unknown and somewhat controversial. Although the best algorithms devised so far do not threaten the RSA system, some totally new and astonishingly clever method might be invented tomorrow. Or it might have been invented yesterday by someone working for a publicity-shy government agency in the suburbs of Washington, D.C. No law of nature or theorem of mathematics forbids efficient factoring; perhaps it is as easy as multiplication if only you know the trick. Thus the security of RSA ciphers (and many others) depends more on ignorance than on knowledge—it depends on what we don't know how to do, rather than on what we know cannot be done.

Bibliography

Bressoud, David M. 1989. *Factorization and Primality Testing*. New York: Springer-Verlag.

Denny, Thomas, Bruce Dodson, Arjen K. Lenstra and Mark S. Manasse. 1993. On the factorization of RSA-120. In *Advances in Cryptology, Crypto '93*. New York: Springer-Verlag, pp. 166–174.

Gardner, Martin. 1977. Mathematical games: a new kind of cipher that would take millions of years to break. *Scientific American* 237 (August): 120–124.

Lenstra, Arjen K., and Mark S. Manasse. 1990. Factoring by electronic mail. In *Advances in Cryptology, Eurocrypt '89*. New York: Springer-Verlag, pp. 355–371.

Lenstra, Arjen K., and Hendrik W. Lenstra, Jr., eds. 1993. *The Development of the Number Field Sieve*. New York: Springer-Verlag.

Rivest, R. L., A. Shamir and L. Adleman.1978. A Method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM* 21 (February):120–126.

Silverman, Robert D. 1987. The multiple polynomial quadratic sieve. *Mathematics of Computation* 48:329–339.

Silverman, Robert D. 1991. Massively distributed computing and factoring large integers. *Communications of the ACM* 34(11):95–103.