## The RSA encryption algorithm

## David S. Gunderson

**Introduction:** Throughout the ages, there have been many methods for encoding (or encrypting) messages. A key is a method for encryption. For example, if the message is "I  $\heartsuit$  MATH!", one way to encrypt it gives the code "J  $\heartsuit$  NBUI!"; the key was to shift letters by one and leave symbols as is. Sometimes a key is called a cipher. The above cipher, called a *shift cipher*, is rather easy to crack. Most ciphers that use a simple translation table are also easy to figure out if the message is long enough, for example, as in cryptograms. The encryption technique we describe below is not only hard to crack, but it has a special, incredible property.

In ancient times, to ensure secrecy, often an encrypted message and a key were sent by separate messengers, or a key was agreed upon first. Could you imagine an encryption technique where the key for encryption is made public? It turns out that the RSA encryption algorithm is a technique which allows me to advertise my key, then using that key, you send me an encrypted message—and even if the enemy intercepts this message, they can't reproduce the original message! The whole trick relies on some simple properties of numbers, those involving primes and a special type of arithmetic. It turns out that the method is relatively secure because large numbers can not be factored easily. On the other hand, the encryption and decryption are surprisingly simple, so the technique is very popular.

Here is a puzzle from Dudeney (1907), called the riddle of St. Edmondsbury based on factoring: They were so overrun with mice that cats from all over the country were brought in to exterminate the vermin. Records were kept, and it was found that every cat killed an equal number of mice, and 1,111,111 mice in all were killed. How many cats were there? Before we solve this, we recall some basic properties of numbers.

**Primes and factoring** A number p is prime if  $p \ge 2$  and its only factors are itself and 1; here are the first few:

$$2, 3, 5, 7, 11, 13, 17, 19, \ldots$$

Euclid showed that there are infinitely many primes. The largest known prime (found in 2001) is  $2^{13466917} - 1$ , over 4 million digits long! For the RSA encryption algorithm, large primes are required; how does one find them? This is a subject that number theorists are hard at work on.

To factor a number n into primes, you only need to divide n by primes 2, 3, 5, 7, ..., up to  $\sqrt{n}$ . Why? If n = pq and both  $p > \sqrt{n}$  and  $q > \sqrt{n}$ , then  $n > \sqrt{n} \cdot \sqrt{n} = n$ , impossible. Is 151 prime? Since  $12 < \sqrt{151} < 13$ . Check 2,3,5,7,11. None go into 151, so 151 is prime.

Let's return to the riddle. Cats  $\times$  mice/cat = total mice. So to find the possible numbers of cats, we factor 1,111,111. Since  $1054 < \sqrt{1,111,111} < 1055$ , check all possible primes 2,3,5,7,11,...,1051, and discover that 1,111,111 =  $239 \times 4649$ , where each of 239 and 4649 is prime. [This "discovery" takes a little work, doesn't it?] So, there was either 1 cat (with a huge appetite), or there were 239, 4649, or 1,111,111 cats.

**Modular arithmetic:** To perform the computations for the encryption algorithm presented here, we need a special kind of arithmetic, one which is in many ways simpler that standard arithmetic. We are all familiar with what is called *clock addition*:  $11 + 5 = 16 \equiv 4$  in the afternoon; throw away 12, get 4. If one throws away any multiple of 12, we write, for example,  $11 + 42 = 53 \equiv 5 \pmod{12}$   $6 \cdot 4 = 24 \equiv 0 \pmod{12}$ .

If instead of 12, we use, say, 5's, then,  $11 + 5 = 16 \equiv 1 \pmod{5}$ ,  $11 + 42 \equiv 3 \pmod{5}$ , and  $6 \cdot 4 \equiv 4 \pmod{5}$ . The number 5 is called a *modulus*. Note that  $5 \equiv 0 \pmod{5}$ .

If  $a \equiv b \pmod{n}$ , we say *a* is *congruent* to *b* modulo *n* (both *a* and *b* have same remainder upon division by *n*, or, equivalently, a - b is divisible by *n*). Most calculators today have a "mod" function button (even the calculator in Windows XP has one!).

In standard arithmetic, the *multiplicative inverse* of 8 is  $8^{-1} = \frac{1}{8}$  since  $8 \times \frac{1}{8} = 1$ . In modular arithmetic, inverses also arise:  $8 \times 3 \equiv 1 \pmod{29}$ , so, modulo 23,  $8^{-1} = 3$ . If the modulus is a prime p, all integers which are not zero (or a multiple of p) have an inverse modulo p. For example, let p = 13. Check that the only number  $m \pmod{13}$  so that  $4m \equiv 1 \pmod{13}$  is m = 10, so, modulo 13,  $4^{-1} = 10$ . The *Euclidean division algorithm* easily finds inverses modulo p. (You'll learn this simple trick early in university math.) If x and n have no common factors, then  $x^{-1} \pmod{n}$  exists.

Here are a couple more facts from number theory that we use for the encryption algorithm (these are only required for the proof, and so can be skipped): If p and q are primes, and  $a \equiv M \pmod{p}$  and  $a \equiv M \pmod{q}$ , then  $a \equiv M \pmod{pq}$ . Fermat's little theorem states that if p is a prime and a is any number which is not a multiple of p, then  $a^p \equiv a \pmod{p}$ . or  $a^{p-1} \equiv 1 \pmod{p}$ . For example, with p = 3, and a = 4,  $4^3 = 64 = 3 \cdot 20 + 4 \equiv 4 \pmod{3}$ . With p = 7, and a = 3,  $3^6 = 729 \equiv 1 \pmod{7}$ .

The RSA public key encryption: We now have the tools to completely describe the technique. The idea is that I publicly advertise a key, you use it to encode a message to me. Here is the simple recipe for encryption.

I pick two large primes, p and q, and put n = pq. I pick d (my secret key) and e so that

$$ed \equiv 1 \mod (p-1)(q-1)$$

In practise, I pick d large at random; most of the time such an e will exist—if not, try another d. I make the pair n, e the public key.

You take a message, convert it into numbers (like A=01, B=02,...), and cut it into blocks. Make sure that blocks are small enough so that  $M^3 < n$  for any M. For a block M of these numbers, you compute  $C = M^e$  and send me C—that's it!

**Decryption:** I compute  $C^d \pmod{n}$  and the answer, remarkably, is M. For the interested reader, here is the proof (you can skip this, and proceed directly to the example which follows): For some integer k, write

$$C^{d} = (M^{e})^{d} = M^{ed} = M^{1+kn}.$$

If  $M \equiv 0 \pmod{p}$ , then  $M^{1+kn} \equiv M \pmod{p}$ . If  $M \equiv 0 \pmod{p}$ , then by Fermat's little theorem,  $M^{p-1} \equiv 1 \pmod{p}$ , and so

$$M^{1+kn} = M \cdot M^{k(p-1)(q-1)}$$
  
=  $M \cdot (M^{p-1})^{k(q-1)}$   
=  $M \cdot 1^{k(q-1)} \pmod{p}$   
=  $M \pmod{p}.$ 

Similarly,  $C^d \equiv M \pmod{q}$ . Remainders are same relative to p and q, so

$$C^d \equiv M \pmod{pq}$$

and since M < n = pq,  $C^d = M$ . This completes the proof.

Visit http://oregonstate.edu/dept/honors/makmur/ where applets do all these steps for you.

**Example:** I pick p = 17, q = 11,  $n = 17 \times 11 = 187$ . I pick e = 3, d = 107 ( $ed = 321 = 2 \cdot 16 \cdot 10 + 1$ ). I post 187, 3. You encode the letter J as 10, and put M = 10; then  $C \equiv M^e \equiv 10^3 \equiv 1000 \equiv 65 \pmod{187}$  and so you send me 65. I compute  $C^d \pmod{n}$ , and find  $C^d \equiv 65^{107} \equiv 10 \pmod{187}$ .

A little history: The RSA algorithm is named after Rivest, Shamir, and Adleman, based on a theory proposed by Diffie and Hellman in 1975 (called *trapdoor functions*). It is now recognized that James Ellis came up with the idea of trapdoor functions in 1969 and Clifford Cocks found the number theory necessary in 1973. Cocks seemed very humble about the math, essentially saying that it was really nothing at all!

In 1977, Martin Gardner (in *Scientific American*), and gave the following value for n (called RSA-129 because it had 129 digits):

An encrypted message was given, and a prize of 100 dollars offered for cracking it. A team of 600 volunteers factored n, a mere 17 years later into the primes:

## 3490529847650949147849619903898133417764638493387843990820577,

## 32769132993266709549961988190834461413177642967992942539798288533.

The first message was "the magic words are squeamish ossifrage." (An ossifrage is an osprey-like bird.) In, 1999, RSA-155 was finally factored into two 78 digit primes. Banks are now using many hundreds of digits. Theoretically, these too can be cracked, however the information will likely be a bit outdated by the time you read it.

What else? The RSA technique can be used to publicly decide on keys going both ways! It can also be used to establish identity (since the enemy might crack the code, then send something else). In all of these aspects to encryption, it is amazing that the same simple number theory can be used.

Are there weaknesses to this algorithm? If someone finds an easy way to factor large numbers, the encryption can theoretically be broken. Larger primes make the encryption more secure; other conditions on the primes also help, but no one can prove that this encryption can't be cracked easily! Mathematicians are now developing other algorithms, for example, some based on elliptic curves, or quantum theory, in case someone cracks this one. Despite potential weakness of the RSA algorithm, it still seems to be the most popular encryption device for today's e-business.

Most of the material here is on the web [I had over 80,000 hits for "RSA encryption algorithm"!] The actual algorithm is now public domain—you can download copies of a program that does everything for you.