RSA Encryption

10 February 2012

"Putting your text in Pig Latin isn't the same as encrypting."

We saw some methods of encryption Wednesday, but none of these is good enough to be used in actual situations. Today we'll talk about one method, RSA Encryption, which is good enough to be used, and is used.

We saw some methods of encryption Wednesday, but none of these is good enough to be used in actual situations. Today we'll talk about one method, RSA Encryption, which is good enough to be used, and is used.

To be useful a method of encryption must be easy and quick to use but very difficult to decrypt unless you are the intended recipient.

RSA Encryption

This encryption method, published by Rivest, Shamir and Adleman in an article published in 1978, allows for data to be easily encrypted and decrypted, yet can keep the data secure.

Data is assumed to be in numerical form; a message consists of a number or a series of numbers. For illustrative purposes we assume a message is a single number, for example, a credit card number.

To use RSA, you begin by choosing two prime numbers, which we will call p and q. Prime numbers are whole numbers greater than 1 which cannot be factored into smaller whole numbers, such as $5, 7$, and 13.

To use RSA, you begin by choosing two prime numbers, which we will call p and q. Prime numbers are whole numbers greater than 1 which cannot be factored into smaller whole numbers, such as $5, 7$, and 13.

We set

$n = pq$

To use RSA, you begin by choosing two prime numbers, which we will call p and q. Prime numbers are whole numbers greater than 1 which cannot be factored into smaller whole numbers, such as $5, 7$, and 13.

We set

$$n = pq$$
$$m = (p-1)(q-1)$$

To use RSA, you begin by choosing two prime numbers, which we will call p and q. Prime numbers are whole numbers greater than 1 which cannot be factored into smaller whole numbers, such as $5, 7$, and 13.

We set

$$n = pq$$
$$m = (p-1)(q-1)$$

We choose a number e, which we will call the encoding number, and we calculate a number d, called the decoding number, to satisfy the equation

$$ed \equiv 1 \pmod{m}$$

To use RSA, you begin by choosing two prime numbers, which we will call p and q. Prime numbers are whole numbers greater than 1 which cannot be factored into smaller whole numbers, such as $5, 7$, and 13.

We set

$$n = pq$$
$$m = (p-1)(q-1)$$

We choose a number e, which we will call the encoding number, and we calculate a number d, called the decoding number, to satisfy the equation

$$ed \equiv 1 \pmod{m}$$

We must choose e appropriately for there to be such a d. Technically, we have to choose e be able to divide by e modulo m.

Suppose M is a message. To encrypt M we compute

$$M^e \pmod{n}$$

Suppose M is a message. To encrypt M we compute

$$M^e \pmod{n}$$

To decrypt, we take the received message N, which is $M^e \pmod{n}$, and compute

$$N^d \pmod{n}$$

Suppose M is a message. To encrypt M we compute

$$M^e \pmod{n}$$

To decrypt, we take the received message N, which is $M^e \pmod{n}$, and compute

$$N^d \pmod{n}$$

The amazing thing is we recover the original message!

Examples of using RSA

The following calculations are done in a computer program that can handle both numerical and symbolic computation. Each of the calculations we'll see were done almost instantaneously.

$$p := 7$$

$$7$$

$$q := 11$$

$$11$$

$$n := p \cdot q$$

$$77$$

$$m := (p-1) \cdot (q-1)$$

$$60$$

$$e := 13$$

$$13$$

$$d := e^{-1} \bmod m$$

$$37$$

$M := 17$

$$17$$

$N := M^e$

$$9904578032905937$$

$N \bmod n$

$$73$$

So the message 17 is encrypted as 73.

N^d

$$8768914275535665941006178673203588185690865716840426568655730913845553$$

$N^d \ \mathbf{mod}\ n$

$$17$$

The number $N^d = 73^{37}$, which amounts to multiplying 73 times itself 37 times.

A little more realistic example

$p := nextprime(10^{200})$

100000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000000000000000000000000000000000000000000000000000\
000000000000000000000000000000000000000357

$q := nextprime(9^{212})$

199256272249431221328603033054645678897075327295912250213439044806853800883809353\
096726339601918790294815253340993739842744300688098446575123760666990589913606\
79021040866132270885145863928226389293864513

These are numbers with at least 200 digits!

$n := p \cdot q$

199256272249431221328603033054645678897075327295912250213439044806853800883809353\
096726339601918790294815253340993739842744300688098446575123760666990589913606\
790210408661322708851458639282263892939356474891930469460143112828005085073662\
558918446406733261977389960468069155199390555313032378850081352490454427347651\
238597153456511454273191825581156405991576241051158920922070599707342237682097\
7909631141

$$m := (p-1)\cdot(q-1)$$

1992562722494312213286030330546456788970753272959122502134390448068538008838093533\
0967263396019187902948152533409937398427443006880984465751237606669905899136063\
7902104086613227088514586392822638929393544723292079751479298267976745386168733\
5881651734476107598429995123995311463612970243457689828308934495423018939377133\
8401697104496304698074405879744865000924401731490548343088435111927558448594583\
8615766272

$$e := nextprime\left(10^{10}\right)$$

$$10000000019$$

$$d := e^{-1} \bmod m$$

43695188012875698000982720932550555985318121438526344668728793360407261047480799\
86318053604533949972954686466778810360433876911555509683919305701225183418596\
03061054385360031311580950667050673678308234980674465273321105688488426295567\
94511427605988718415321137342297457875995438488008577268960187975297081644999\
89310975673421504125013337960866494607643995786216090192872760436977921403690\
554872859

$M := 1234567890$

$$1234567890$$

$N := M^e \bmod n$

```
Error, numeric exception: overflow
```

$$M := 1234567890$$

$$1234567890$$

$$N := M^e \bmod n$$

```
Error, numeric exception: overflow
```

If we could write out M^e, it would have about 100 billion digits. At 10 characters per inch, writing it in one long string, it would be more than 5 times the distance between the earth and moon, or printing it would take about 50 million pages!

```
N := M&^e mod n;
```

```
118354142379010621989167711214088091289399779111605628193080230845868276129926738\
    132277508698788350724800183060965690417377880538449289524631012371060928125968\
    862556474797001533495948389105599815014139966716373280043391109841172792668003\
    966062747532332173533979896331254308856186785593262593151588974967086207785027\
    686937724224712746513368535486792780600720389737969121247126578888487529915911\
    0487147197
```

```
N &^d mod n;
```

```
1234567890
```

Why does RSA work?

There is a result, called Fermat's Little Theorem, which says that if p is a prime number, and a is not evenly divisible by p, then

$$a^{p-1} \equiv 1 \pmod{p}.$$

Why does RSA work?

There is a result, called Fermat's Little Theorem, which says that if p is a prime number, and a is not evenly divisible by p, then

$$a^{p-1} \equiv 1 \pmod{p}.$$

A generalization, called Euler's Theorem, says in our case that if M is not divisible by p or q, then $M^{(p-1)(q-1)} \equiv 1 \pmod{n}$.

The choice of e and d says that $ed = 1 + km$ for some k. Encrypting, then decrypting yields

$$M \longrightarrow M^e \longrightarrow (M^e)^d = M^{ed}$$

The choice of e and d says that $ed = 1 + km$ for some k. Encrypting, then decrypting yields

$$M \longrightarrow M^e \longrightarrow (M^e)^d = M^{ed}$$

Modulo n, we have

$$\begin{aligned} M^{ed} &= M^{1+km} = M \cdot (M^m)^k \\ &\equiv M \cdot 1 \equiv M \pmod{n} \end{aligned}$$

The choice of e and d says that $ed = 1 + km$ for some k. Encrypting, then decrypting yields

$$M \longrightarrow M^e \longrightarrow (M^e)^d = M^{ed}$$

Modulo n, we have

$$\begin{aligned} M^{ed} &= M^{1+km} = M \cdot (M^m)^k \\ &\equiv M \cdot 1 \equiv M \pmod{n} \end{aligned}$$

Because $M^m \equiv 1 \pmod{n}$ by Euler.

In using RSA, one must keep p, q, m, d private, although n and e can be public. Knowing any one of p, q, m, d is enough to break the code.

In using RSA, one must keep p, q, m, d private, although n and e can be public. Knowing any one of p, q, m, d is enough to break the code.

Why isn't it easy to break RSA? Can't we just factor n to get p and q, and then compute d?

Clicker Question

Suppose $n = 91$. Can you find the two prime factors p and q for which $pq = 91$?

If you can find p and/or q, enter one of them into your clicker and send. If you cannot, enter 0.

An RSA Challenge

In 1977 Martin Gardner issued a challenge in his Scientific American column. He printed some encrypted text, using RSA with a 129 digit modulus, which was the product of two unknown prime numbers, and he challenged people to decrypt the text. The modulus was

An RSA Challenge

In 1977 Martin Gardner issued a challenge in his Scientific American column. He printed some encrypted text, using RSA with a 129 digit modulus, which was the product of two unknown prime numbers, and he challenged people to decrypt the text. The modulus was

$$\begin{aligned}\text{RSA-129} \quad = \quad & 114381625757888867669235779976146612010218\,29\\ & 6721242362562561842935706935245733897830597\\ & 1235639587050589890751475992900268795435 41\end{aligned}$$

An RSA Challenge

In 1977 Martin Gardner issued a challenge in his Scientific American column. He printed some encrypted text, using RSA with a 129 digit modulus, which was the product of two unknown prime numbers, and he challenged people to decrypt the text. The modulus was

$$
\begin{aligned}
\text{RSA-129} \;=\;& 11438162575788886766923577997614661201021829 \\
& 6721242362562561842935706935245733897830597 \\
& 123563958705058989075147599290026879543541
\end{aligned}
$$

He claimed it would take millions of years to break.

Gardner was a little off.

Gardner was a little off.

RSA-129 was factored in April 1994 by a team led by Derek Atkins, Michael Graff, Arjen K. Lenstra and Paul Leyland, using approximately 1600 computers from around 600 volunteers connected over the Internet. It only took 17 years!

Gardner was a little off.

RSA-129 was factored in April 1994 by a team led by Derek Atkins, Michael Graff, Arjen K. Lenstra and Paul Leyland, using approximately 1600 computers from around 600 volunteers connected over the Internet. It only took 17 years!

$$\begin{aligned}\mathrm{RSA\text{-}129} &= 3490529510847650949147849619903898\\ &\quad\ 133417764638493387843990820577\\ &\times\ 3276913299326670954996198819083446\\ &\quad\ 1413177642967992942539798288533\end{aligned}$$

Finding Large Primes

Even though factoring large numbers takes a great deal of time, it turns out that checking if a number is prime is relatively easy. For example, if we wish to test if a number b is prime, we can choose various values for a not divisible by b and check if $a^{b-1} \equiv 1 \pmod{b}$. According to Fermat, this must be true if b is a prime.

Finding Large Primes

Even though factoring large numbers takes a great deal of time, it turns out that checking if a number is prime is relatively easy. For example, if we wish to test if a number b is prime, we can choose various values for a not divisible by b and check if $a^{b-1} \equiv 1 \pmod{b}$. According to Fermat, this must be true if b is a prime.

For example,

$$2^5 = 32 \equiv 2 \pmod{6},$$

so 6 cannot be prime.

Finding Large Primes

Even though factoring large numbers takes a great deal of time, it turns out that checking if a number is prime is relatively easy. For example, if we wish to test if a number b is prime, we can choose various values for a not divisible by b and check if $a^{b-1} \equiv 1 \pmod{b}$. According to Fermat, this must be true if b is a prime.

For example,

$$2^5 = 32 \equiv 2 \pmod{6},$$

so 6 cannot be prime.

There are other primality tests, which can tell if a number is not prime. By using several, one can check with high probability that a number is prime. This is what the computer did in the RSA computations we showed earlier.

So, when the computer says a large number is prime, what it really says is that the probability is very high that the number is prime. While this may seem unsatisfactory, no example has ever been found of a number being reported as prime but failing to be prime.

So, when the computer says a large number is prime, what it really says is that the probability is very high that the number is prime. While this may seem unsatisfactory, no example has ever been found of a number being reported as prime but failing to be prime.

Factoring a large integer, however, is another story. It turns out that factoring a large number takes an enormous amount of time, compared to checking if a number is prime.

So, when the computer says a large number is prime, what it really says is that the probability is very high that the number is prime. While this may seem unsatisfactory, no example has ever been found of a number being reported as prime but failing to be prime.

Factoring a large integer, however, is another story. It turns out that factoring a large number takes an enormous amount of time, compared to checking if a number is prime.

On the RSA.com webpage, they recommend using moduli of around 300 digits.

Will we run out of prime numbers for RSA as computers get faster?

Will we run out of prime numbers for RSA as computers get faster?

Euclid proved, over 2,000 years ago, that there are infinitely many primes.

Will we run out of prime numbers for RSA as computers get faster?

Euclid proved, over 2,000 years ago, that there are infinitely many primes.

G. H. Hardy, one of the most important mathematicians of the 20th century and who did work in number theory, published a book, A Mathematician's Apology, in 1940. He gave the following proof of Euclid, what he calls is one of the most beautiful results of mathematics.

G. H. Hardy

The first is Euclid's proof of the existence of an infinity of prime numbers. The prime numbers or primes are the numbers

$$2, 3, 5, 7, 11, 13, 17, 19, 23, 29, \ldots \tag{A}$$

which cannot be resolved into smaller factors. Thus 37 and 317 are prime. The primes are the material out of which all numbers are built up by multiplication: thus $666 = 2 \cdot 3 \cdot 3 \cdot 37$*. Every number which is not prime itself is divisible by at least one prime (usually, of course, by several). We have to prove that there are infinitely many primes, i.e. that the series (A) never comes to an end.*

Let us suppose that it does, and that $2, 3, 5, \ldots, P$ is the complete series (so that P is the largest prime); and let us, on this hypothesis, consider the number Q defined by the formula

$$Q = (2 \cdot 3 \cdot 5 \cdots P) + 1.$$

It is plain that Q is not divisible by and of $2, 3, 5, \ldots P$; for it leaves the remainder 1 when divided by any one of these numbers. But, if not itself prime, it is divisible by some prime, and therefore there is a prime (which may be Q itself) greater than any of them. This contradicts our hypothesis, that there is no prime greater than P; and therefore this hypothesis is false.

The proof is by reductio ad absurdum, and reductio ad absurdum, which Euclid loved so much, is one of a mathematicians finest weapons. It is a far finer gambit than any chess gambit: a chess player may offer the sacrifice of a pawn or even a piece, but a mathematician offers the game.

A quote from A Mathematician's Apology

The following quote is referring to two theorems, one by Fermat and Euclid's theorem proving there are infinitely many prime numbers.

There is no doubt at all, then, of the seriousness of either theorem. It is therefore the better worth remarking that neither theorem has the slightest practical importance. In practical application we are concerned only with comparatively small numbers; only stellar astronomy and atomic physics deal with large numbers, and they have very little more practical importance, as yet, than the most abstract pure mathematics.

I do not know what is the highest degree of accuracy ever useful to an engineer—we shall be very generous if we say ten significant figures. Then 3.14159265 *(the value of* π *to eight places of decimals) is the ratio*

$$\frac{314159265}{1000000000}$$

of two numbers of ten digits. The number of primes less than 1,000,000,000 is 50,847,478 : that is enough for an engineer, and he can be perfectly happy without the rest.

Hardy's book was published in 1940. He didn't foresee the internet! We are using primes with hundreds of digits today.

Hardy's book was published in 1940. He didn't foresee the internet! We are using primes with hundreds of digits today.

Because there are prime numbers of unlimited size, RSA can be used even as computers get faster, by choosing larger and larger primes. As long as factoring is a relatively slow process, this method will be useful.

Hardy's book was published in 1940. He didn't foresee the internet! We are using primes with hundreds of digits today.

Because there are prime numbers of unlimited size, RSA can be used even as computers get faster, by choosing larger and larger primes. As long as factoring is a relatively slow process, this method will be useful.

New technology, such as quantum computing, could change this, however. We would then have to come up with even more clever methods of encryption.

Next Time

We will begin a three week discussion of probability. Next week will focus on a little history of the subject and some of the principal ideas. In particular, we'll conduct some probability experiments to help understand the meaning of probability and to see some common misconceptions.

Quiz Question

There are useful applications that involve having very large prime numbers.

A True

B False