

The RSA Cryptosystem

In most coding schemes, coding information and decoding encrypted messages are similar processes, so anyone who knows how to send a code knows how to decode encrypted messages. In 1976, it was proposed that it might be possible to have what are called asymmetric encryption/decryption schemes--ones in which coding and decoding are considerably different. This allows for the idea of a **Public Key code**, one in which the scheme for encrypting messages is made public, so everyone has access to it. But never-the-less, only people "in the know" will be able to decode encrypted messages. An example where this might be useful is in banking, where a large number of people might wish to send information confidentially, but only the bank needs to know how to decode messages.

In 1977, three mathematicians proposed just such an asymmetric coding scheme. The mathematicians were L. Adelman, A. Shamir, and R. Rivest, and the method they devised is called the RSA Cryptosystem.

The method works as follows. In private, you select two large primes, say for example $p = 2447$ and $q = 4651$. You then calculate $n = pq$ and $\varphi(pq)$, using the formula $\varphi(pq) = (p - 1)(q - 1)$. In our case, these would be $n = 11380997$ and $\varphi(n) = 11373900$. Next, you pick a number e for use in encryption. This number should be relatively prime to $\varphi(n)$. For the above, I will arbitrarily set $e = 23$. Now solve the congruence $de \equiv 1 \pmod{\varphi(n)}$ for d . As mentioned previously, this is accomplished by solving $de + x\varphi(n) = 1$ by the Euclidean algorithm. In our case, the calculations are as follows:

$$\left. \begin{array}{l} 11373900 = 494517 \cdot 23 + 9 \\ 23 = 2 \cdot 9 + 5 \\ 9 = 1 \cdot 5 + 4 \\ 5 = 1 \cdot 4 + 1 \end{array} \right\} \text{backsubstituting: } \left\{ \begin{array}{l} 1 = 5 - 4 \\ = 5 - (9 - 5) \\ = 2 \cdot 5 - 9 \\ = 2(23 - 2 \cdot 9) - 9 \\ = 2 \cdot 23 - 5 \cdot 9 \\ = 2 \cdot 23 - 5(11373900 - 494517 \cdot 23) \end{array} \right.$$

from which it follows that $d = 2 + 5 \cdot 494517 = 2472587$ works.

Now you publish the numbers n and e to the world, while keeping d , p , q and $\varphi(n)$ private. In fact, at this point, all you need are d and n , so you can get rid of p , q , $\varphi(n)$.

Encoding:

Given a message to code, such as “Math is fun!”, you must first convert the message to numerical format. Here is one way: using the ASCII system, which assigns a number to each character, given any character c , assign it the number $\text{ASCII}(c) - \text{ASCII}(\text{space}) + 10$. This will do the following:

character	space	!	“	#	\$	%	&	‘	()
number	10	11	12	13	14	15	16	17	18	19
character	*	+	,	-	.	/	0	1	2	3
number	20	21	22	23	24	25	26	27	28	29
character	4	5	6	7	8	9	:	;	<	=
number	30	31	32	33	34	35	36	37	38	39
character	>	?	@	A	B	C	D	E	F	G
number	40	41	42	43	44	45	46	47	48	49

character	H	I	J	K	L	M	N	O	P	Q
number	50	51	52	53	54	55	56	57	58	59
character	R	S	T	U	V	W	X	Y	Z	[
number	60	61	62	63	64	65	66	67	68	69
character	\]	^	_		a	b	c	d	e
number	70	71	72	73	74	75	76	77	78	79
character	f	g	h	i	j	k	l	m	n	o
number	80	81	82	83	84	85	86	87	88	89
character	p	q	r	s	t	u	v	w	x	y
number	90	91	92	93	94	95	96	97	98	99

As you can see, this method has the drawback of not being able to represent z with a 2-digit number. This will be fine for our course: let's not use any codes containing a lower case z . "Math is fun!" now translates into 55 75 94 82 10 83 93 10 80 95 88 11. Usually, stuff is stuck together to form larger numbers. The rule of thumb is that you can stick things together in groups where all numbers stay less than n . In this case, we can combine 3 numbers at a time to get the message 557594 821083 931080 958811.

So far, there has been no encryption--we have only changed the message into a useful form for transmission. Now comes the encryption: For each part, M , of the message, we calculate $E \equiv M^e \pmod{n}$. For our case, we must calculate 557594^{23} , 821083^{23} , 931080^{23} , 958811^{23} , all modulo 11380997. Our encrypted message (calculations done with Maple) is

1448130 8608216 2995328 5853993.

Decoding:

Now, suppose someone sends us an encrypted message such as the above. How do we decode it? The answer is as follows: If $E \equiv M^e \pmod{n}$, then $M \equiv E^d \pmod{n}$. For the proof of this, recall Euler's Theorem: if $(a, n) = 1$, then $a^{\varphi(n)} \equiv 1 \pmod{n}$. We have $E^d \equiv (M^e)^d \equiv M^{ed} \pmod{n}$. Now d was picked so that $ed \equiv 1 \pmod{\varphi(n)}$. Thus, for some integer k , $ed = 1 + k\varphi(n)$. Thus, $E^d \equiv M^{ed} \equiv M^{1 + k\varphi(n)} \equiv M \cdot M^{k\varphi(n)} \equiv M \cdot (M^{\varphi(n)})^k \equiv M \cdot 1 \equiv M \pmod{n}$. So to decode, we simply calculate $E^d \pmod{n}$ for each code word E .

To check this against our code, we must calculate $1448130^{2472587}$, $8608216^{2472587}$, $2995328^{2472587}$, $5626717^{2472587}$ all modulo 11380997. Again, complements of Maple (yes, I actually did check my answers!), this gives

557594 821083 931080 958811,

Just as it should.

Breaking the code:

Now suppose you don't know the code, but you intercept an encrypted message. Since this is a public code, you also know e and n . To decode the message, one method is to find d . There might be other methods--it is an open question as to whether there might be other ways to decode the message. But certainly, if you know d , you can decode the message. Let us assume that this is the only way to break the code. How does one find d ? One way is the same way the code maker found d : solve the

congruence $de \equiv 1 \pmod{\varphi(n)}$. But to do this, you have to know $\varphi(n)$. (Again, there might possibly be other ways to calculate d . Let's ignore this issue as well.) Now if you can find the factorization $n = pq$, then you can find $\varphi(n)$ by using the formula $\varphi(pq) = (p - 1)(q - 1)$. In fact, knowing $\varphi(n)$ is equivalent to knowing the factorization of n . By this I mean the following: If you know the factorization of n , you can easily find $\varphi(n)$. In addition, if you know $\varphi(n)$, you can easily find the factorization of n . Here is how: Since $\varphi(n) = (p-1)(q-1)$ and $(p - 1)(q - 1) = pq - p - q + 1 = n + 1 - p - q$, if we know both n and $\varphi(n)$, then we can calculate $p + q$. In particular, we get that

$$\begin{cases} pq = n \\ p + q = n + 1 - \varphi(n). \end{cases}$$

These two equations can be solved for p and q as follows: write the second as $p + q = m$ (so $m = n + 1 - \varphi(n)$). Then $m^2 = (p + q)^2 = p^2 + 2pq + q^2$ so $m^2 - 4n = p^2 + 2pq + q^2 - 4pq = p^2 - 2pq + q^2 = (p - q)^2$. Thus, assuming that $p \geq q$, we have

$$\begin{cases} p + q = m \\ p - q = \sqrt{m^2 - 4n} \end{cases} \quad \text{So} \quad \begin{cases} p = \frac{m + \sqrt{m^2 - 4n}}{2} \\ q = \frac{m - \sqrt{m^2 - 4n}}{2} \end{cases}.$$

In our ongoing example, $n = 11380997$ and $\varphi(n) = 11373900$. Thus, $m = 11380997 + 1 - 11373900 = 7098$; $\sqrt{m^2 - 4n} = 2204$ and

$$\begin{cases} p = \frac{7098 + 2204}{2} = 4651 \\ q = \frac{7098 - 2204}{2} = 2447, \end{cases}$$

as expected.

Some Coded Messages

Here are some messages coded using the RSA cryptosystem. They are of varying degrees of difficulty to crack ranging from crackable via a calculator to probably not crackable for people in this class. Each of them uses the scheme employed in these notes for changing text to numbers. In all codes, $e = 101$. Also, the size of the unencrypted code words is given. In each case, the size of the unencrypted code words will be multiples of 2 since each character takes two digits to express. Enjoy.

Code 1. $N = 24503$, code words are of size 4 (2 characters each) the message is:

20481	2486	4347	8113	15980	12383
14367	19816	4347	23410	16945	3506
20070	9949	10326	131	12383	2052
16017	18249	12298	18233		

Code 2: $N = 617686377693546512802931776089$, code words are of size 26 (13 characters each) the message is:

304084172656628563288671501596	502227429711472113479835441149
4410617979560884741888869338	536398929623731509782526102551

Code 3: $N = 16247340161155710461285128279343888221248228063089$, code words are of size 40 (20 characters each) the message is:

10973097990262172534649502233754367294018604349367
13179334637681554871302149733792103122670212441993
12627193325235355634022555287492923621333942824475
8132419086499219567815180056661335216969695196582

Code 4:

$N = 334875926730102513053308763219080292886660060244626155920228665358301351880157$

Code words are of size 50 (25 characters each) the message is:

164088027364964517301388028503864978715459543430011895201520116092421461392280
188606128968363260312173732640110779634980605825566626960146341514647455285111