MA4220: Number Theory (Spring 2011)
Encoding via RSA*

1 The setup

To receive encrypted messages using RSA, Alice announces the following information.

- The product pq, where p and q are distinct, large primes (*Note:* the identities of p and q themselves are kept private).
- A dictionary for converting characters into numbers. We'll use

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16
Q	R	S	T	U	V	W	X	Y	Z		.	?	,	!	'
17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32

  Note that 27 is a space.

- A "block length" B that tells one how to divvy an initial message into smaller sub-messages. The block length should be chosen smaller than the number of digits in pq. Since our dictionary gives a two digit number for each character, we'll need block lengths to be at most the largest even number smaller than the number of digits in pq. Given our dictionary, we can simplify things and just assume that the block length is 2. (*Note:* If using something like `Sage` to help encode and decode, letters like A should just be converted to 1 instead of 01.)
- An encoding key E, which is a natural number satisfying $(E, (p-1)(q-1)) = 1$.

2 Encoding

If Bob wants to send a message to Alice, then Bob follows these steps.

1. The original message is converted into a string of numbers via the dictionary. This string of numbers will be denoted W.
2. The string W is split into substrings W_i of length B. If the final substring doesn't have enough characters, it is padded with 0's until it has the correct length.
3. For each i, the quantity $W_i^E \pmod{pq}$ is computed. A computer can do this very quickly.
4. The encoded messages $W_1^E, W_2^E, \cdots, W_n^E$ are sent in order.

3 Decoding

To decode, Alice needs to compute the decryption key D, which is a natural number satisfying $ED = 1 + y(p-1)(q-1)$. Notice this is easy to compute if you know $(p-1)(q-1)$. In particular, you can use Euler's Theorem to find the inverse of $E \pmod{\phi(pq)}$:

$$D \equiv E^{\phi((p-1)(q-1))-1} \pmod{(p-1)(q-1)}.$$

*This work is a modification of work written by Andrew Schultz of Wellesley College.

 Written by D.C. Ernst

The security of RSA rests in the fact that it's hard for anyone but Alice to determine $(p-1)(q-1)$ by looking at the product pq alone — one also needs to know what p and q are individually, which means one has to be able to factor pq. If p and q are large enough, this factorization is impractical.

To decode a message, Alice follows these steps.

1. For each received message W_i^E, one computes $(W_i^E)^D \pmod{pq}$. Theorem 5.4 says this is the same as W_i.
2. One then concatenates the strings W_i to reconstitute W.
3. The original message is translated using the original alphanumeric dictionary.