AN EXAMPLE OF THE RSA ALGORITHM

Each character of the message corresponds to an ASCII number in the range 32–126. For example:

(space) $\to 32$, ! $\to 33, \ldots, 0 \to 48, \ldots, 9 \to 57, A \to 65, \ldots, Z \to 90, \ldots, a \to 97, \ldots, z \to 122, \ldots,$~ $\to 126$.

- Alice juxtaposes ASCII numbers in groups of 4 to get strings of numbers each less than pq.

- Alice encrypts M by calculating $C = M^e \pmod{pq}$ and sends a file of encrypted numbers to Bob.

- Bob decrypts each number C from the file by calculating $M = C^d \pmod{pq}$.

$p = 123456789234543234567893$

$q = 132345434567623456789153$

$pq =$
163389424215691216400529225322155884871644646 29

$e = 35,\ d =$
144716347162469363097609333894212964552761810 03

Message: **Proofs are fun?**

Corresponding ASCII string:

80 114 111 111 102 115 32 97 114 101 32 102 117 110 63

Now juxtapose every 4 ascii numbers to get the following numbers:

$M_1 = 80114111111,\ M_2 = 1021153297,$

$M_3 = 11410132102,\ M_4 = 11711063.$

Then encrypt M_1, M_2, M_3, M_4 to get C_1, C_2, C_3, C_4.

For example:

$C_1 = (80114111111)^{35} \pmod{pq}$

$=162784882194094963102102219420406867370675445 97.$

$C_2 =$
401063883997071061715861124151219622826040719 6

$C_3 =$
906287009589909424930273564722222905526160116 4

$C_4 =$
148543508591069430412131339701946874211022010 41

To decrypt C_1, we recover M_1 by modular exponentiation:

$M_1 =$
$(162784882194094963102102219420406867370675445 97)^d \pmod{pq}$

$= 80114111111.$

We read off ASCII numbers $80, 114, 111, 111$ and characters P r o o.