Jon Sonderman & Geoff Appelbaum  
CS2911  
Lab 9

Question 1: In this lab, Trudy is able to find the private key from the public key. Why is this not a problem for RSA in practice?

Mostly because RSA isn’t using 8 bits, so the numbers involved get astronomically huge to break it by brute force. It becomes nearly impossible to do so when we get to 64 bit encryption, and thus the message involved become safe.

Question 2: How is the RSA code able to run using such large numbers?

The modulus is a one way operator, where many inputs result can result in one output, thus allowing a shadow to pass over the identity of the real answer. Further, because a large amount of primes could be the answer, and modulusing by such primes is a much simpler operation than turning them into their respective right multiples

Excellent Credit:

Bob Step 1

Public key:   
(17, 41989)

Private key:   
(22013, 41989)

Message: Bob owes Trudy $100.99

Hash: 069e  
Encrypted Hash: 22d9

Trudy step 2

Forged Message: Bob owes Trudy $199.00  
Encrypted Hash: 22d9

Alice step 3

Message checked, verified as true. Oh no! Bamboozled

Step 4

To actually decrypt the message, Trudy would need to brute force it, so a higher encryption would prevent her from accessing the message content and thus tampering with it. Further, the checksum could be adapted to be computed from length of message, or form combinations of other characters, or any much more complicated hash computation