CS161–Fall 2014 — Project0 Write-Up

Erik Bartlett, `cs161-en`

September 22, 2014

1.

cert2.crt

2.

Given m, d, n **- run the following algorithm:**

```
$result = 1$
while $d \geq 0$
if $d\ mod\ 2 = 1 \rightarrow result = result * m$
$m* = m$
$d = d \div 2$
return $result$
```

This algorithm is essentially finding whether the lowest value bit of the exponent is 1 **or** 0**, and if it is one, multiplying it into the result. Then it divides by 2, shifting the exponent by one bit to the right, and increases the base by what it was squared (repeatedly squaring the message).**
This algorithm is used to compute $m^{ed}\ mod\ p$ **in the RSA security algorithm - where** ed **is a large number.**

3.

Given that raising m **to the power of** e **means multiplying** m e **times, we can conclude that without repeated squaring we will do** e $O(1)$ **computations, meaning our run time will be in** $O(e)$

4.

Because we are calculating out the binary form of the exponent and using it to know which values to multiply by - we can do at most as many multiplications

as there are bits in e. There are at most $log_2\ e$ bits in e. Therefore the runtime must be $O(log_2\ e)$

5.

The job of certificate authorities is to verify that the public key advertised by a given server is for that servers company - that the server isn't faking to be someone that it is not. If a CA signs a certificate without making sure the recipient is who they think it is then people can pretend to be whoever the CA said they were and take users unaware - as they can use SSL/RSA for connections and trick the user into giving them private information.