Decryption

known

512

C ≡ Cipher Text → string of bits

e ≡ public key

d ≡ private key

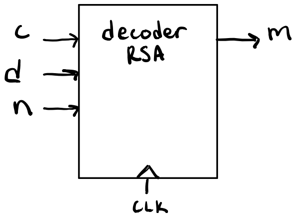
n ≡ P * Q → product of prime numbers.

* Clk ≡ Common system clock (assumed)

Unknown

m ≡ message Text (output)

Top Level



decryption equation

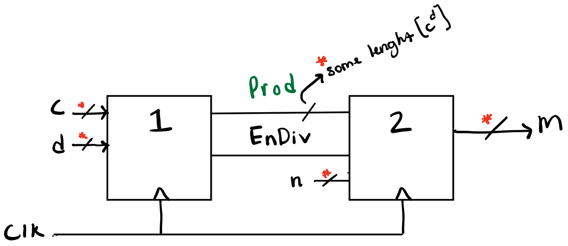
$$m = \underline{\underline{c}}^{d} \; \underline{\underline{(mod(n))}}$$

→ -division
-write a module that performs mod

Steps to decrypt

1. - calculate c^d → *- okay -maybe custom
   - store into register Prod
2. - remainder of div. $\frac{Prod}{n}$ → Chinese remainder theorem
   - output m (string to calculate)

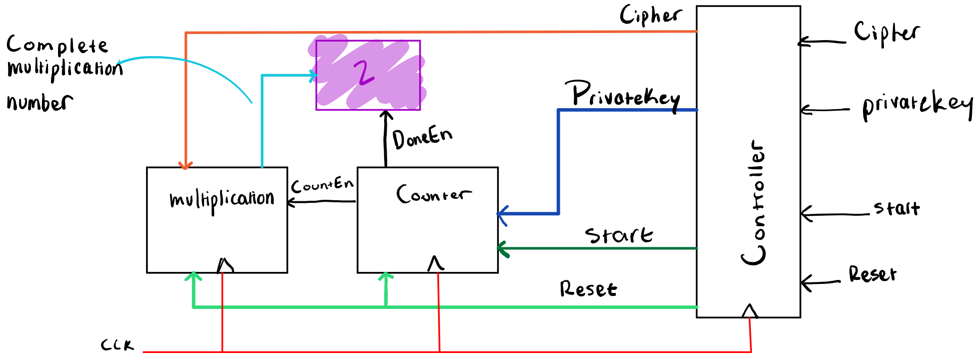
EnDiv ≡ Enable signal sent to 2 when prod has been calculated.



- are keys / product of prime constant lenght?
  - if not, what is max lenght?

* lenght of input/output not determined.

1)



Counter - Counts to privatekey, sets CountEn high to signal the

DoneEn - pulse sent when Counter is done, signaling modulus step to start

CountEn - high when Counter is counting to enable multiplication module