

Timothy Leung

1 a

Let $a, m > 0$ be n-bit integers such that $\gcd(a, m) = 1$. How can we use Euler's theorem to compute the inverse of a modulo m

Euler's theorem states that if a and m are coprime then:

$$a^{\phi(m)} \equiv 1 \pmod{m}$$

let $b = a^{\phi(m)-1} \pmod{m}$.

Then

$$ab = a^{\phi(m)-1}a = a^{\phi(m)} \equiv 1 \pmod{m}$$

Thus, b is the inverse to a .

1 b

Since $\gcd(m, a) = 1$, the extended Euclidean algorithm would produce integers s, t such that

$$\gcd(m, a) = sm + ta = 1$$

Since s is an integer, $sm \pmod{m} \equiv 0$. Since addition is closed under modulus, $sm + ta = 0 + ta \equiv ta \equiv 1 \pmod{m}$. Thus t must be $a^{-1} \pmod{m}$

1c

The inverse I found was

4558498324486549181452716451987626804846091211559485277181551163601073764035252832
13075123888664381192832794777034012202625978250778619258394374862217

the extended Euclidian algorithm was significantly faster.

Euler's algorithm took 2.062000000023545 ms.

Extended Euclidean took 0.7098999999470834 ms.

2 a

if x is small enough, then $x^e < n$. This results in $x^e \pmod{n} \equiv x^e$. So we no longer need to solve the discrete algorithm problem to find x , instead we can just take the e 'th root of x .

For this case $x = 38745745356349$.

$y = x^e = 131089570084977332238073581941947577721291475320895338144511363627800743530233098$

$y < n$ so $y \pmod{n} = y$

so $x = y^{\frac{1}{e}}$

2b

```
n =
508281196310201376192554864656699346831575429768465482788715190735760361687281
737746563113895010157

e = 7
y =
406648847744033968991151413850899861366296698228754391905600624292459633536428
6584668317646217011

expected_x = round(pow(y,1/e))

print(expected_x)

print(pow(expected_x,e) == y)
```

$x = 63287374328731$

3

We have access to y_1, y_2, e_1, e_2, n

$$y_1 = x^{e_1} \pmod{n}$$

$$y_2 = x^{e_2} \pmod{n}$$

Since $\gcd(e_1, e_2) = 1$,

we use Extended Euclidean algorithm to calculate unique integers s, t such that $se_1 + te_2 = 1$

$$y_1 = x^{e_1} \pmod{n}$$

$$y_1^s = x^{se_1} \pmod{n}$$

$$y_2 = x^{e_2} \pmod{n}$$

$$y_2^t = x^{te_2} \pmod{n}$$

$$y_1^s y_2^t = x^{se_1} x^{te_2}$$

$$y_1^s y_2^t = x^{se_1 + te_2}$$

$$y_1^s y_2^t = x^1 = x$$

Thus we can extract the original message from the two cipher texts.