CSCI-4116
Assignment 8

Anas Alhadi

B00895875

March 26, 2025

Note: all code was written to run on Sage Cell Server (sagecell.sagemath.org), with the language option set to Sage

---

Question 1

Part A

Code:

```
R = IntegerModRing(3511)
_,x,y = xgcd(65537,3511)

print("(x,y) =", (x,y))
```

Output:

```
(x,y) = (-1405, 26226)
```

Part B

Code:

```
p, m = 1234567, 100000    # defining the modulo to acquire the last 5 digits
R = IntegerModRing(m)     # (Z/mZ)
a = R(3)                  # 3 ∈ (Z/mZ)

n = euler_phi(m)          # φ(m)
N = IntegerModRing(n)     # (Z/φ(m)Z)
p_mod_n = N(p)            # calculating the congruence: p_mod_n ≡ p (mod φ(m))

a^p_mod_n                 # 3^φ(m) (mod m)
```

Output:

```
40587
```

Part C

Code:

```
a, r, m = 314, 271, 11111       # For the form a*x ≡ r (mod m)
R = IntegerModRing(m)

if gcd(a,m) == 1:
  a_inv = inverse_mod(a, m)
  x = r*a_inv                   # We multiply both sides by the inverse
  x = R(x)                      # reducing x to modulo m
  print(x)
else:
  print("No unique solution")
```

Output:

```
10298
```

Part D

We first check if a solution exists. Given the form:

$$ax + by = d$$

A solution exists if and only if $gcd(a, b)|d$. By corollary 5.1 (in the notes)

In our case the equation is:

$$216x + 606y = 66$$

And

$$gcd(216, 606) = 6, \qquad 6|66$$

We now reduce the equation by the gcd (divide it), So:

$$36x + 101y = 11$$

Such that the $gcd(a, b) = 1$ and we can use the Extended Euclidean Algorithm to find the multiplicative inverse of 36 in modulo 101. I do this using the bellow code in Sage:

Code:

```
inverse_mod(36, 101)
```

Output

```
87
```

So $x = 87$ is a solution for $36x + 101y = 1$ But we want to solve for 11. Thus we multiply both sides by 11 (I disregard the value of y since it disappears when we apply mod 101):

$$x = 87 * 11 \equiv 48(\text{mod } 101)$$

So:

$$36(48) + 101y = 11$$

And we have it that $x = 48$ is a solution to the congruence $216x \equiv 66(\text{mod } 606)$

We acuired this solution by restricting the modulo to 101 from 606 (which makes it so that the congruence is solvable). To extend the modulo back to 606 and find the remaining possible values of x we add multiples of 101 to x.

$$x = \{48 + 101n, \quad n = \{0, 1, 2, 3, 4, 5\}\}$$

So the solution is:

$$x = \{48, 149, 250, 351, 452, 553\}$$

Question 2

Part A

Given: $n = 899$ $e = 11$ $c = 468$. We compute $\varphi(n) = 840$

We first check the $gcd(e, \varphi(n)) = gcd(11, 840) = 1$ So the key is valid. To find the decryption key d we need to find the mutiplicative inverse of e in $\mathbb{Z}/\varphi(n)\mathbb{Z}$.

Using the Extended Euclidean Algorithm, we get that $d = 611$

It follows then that the plaintext m is:

$$m = c^d (\text{mod } n)$$
$$m = 468^{611} (\text{mod } 899)$$

$$m = 13$$

The decryption function in Sage is:

Code:

```
def decrypt(n,e,c):
  R = IntegerModRing(n)
  c = R(c)                  # redefing the cipher text so that: c ∈ (Z/nZ)

  phi_n = euler_phi(n)
  d = inverse_mod(e, phi_n)

  m = c^d
  m = R(m)                  # Same as for c, this is equivalent to: m (mod n)

  return m
```

Calling the function:

```
decrypt(899,11,468)
```

returns the same value as before. $m = 13$

Part B

We repeat the exact same steps, this time calling:[1]

```
decrypt(11413,7467,5859)
```

We get $m = 1415$

[1] In both parts a and b, we easily check by raising $m^e (\text{mod } n)$ which returned c so the function works correctly for the given inputs

Question 3

Given: $n = 642401$, $516107^2 \equiv (\text{mod } n)$ and $187722^2 \equiv 2^2 \times 7(\text{mod } n)$

We make use of the **Basic Princliple** mentioned in appendix A:

If $x^2 \equiv y^2(\text{mod } n)$ and $x \not\equiv \pm y(\text{mod } n)$. Then $gcd(x-y,n)|n$ and is non-trivial (niether 1 nor n)

Observe that:

$$(516107)^2 \times (187722)^2 \equiv 2^2 \times 7^2(\text{mod } 642401)$$

Since the Ring $(\mathbb{Z}/m\mathbb{Z}, +, \cdot)$ is commutative, we can write it as:

$$(516107 \times 187722)^2 \equiv (14)^2(\text{mod } 642401)$$

$$(96884638254)^2 \equiv (14)^2(\text{mod } 642401)$$

We can confirm using Sage that:

$$\begin{aligned} 95884638254 \quad &\equiv \quad 289038 \quad \not\equiv \quad 14 \\ &\not\equiv \quad -14 \quad \equiv 642387 \end{aligned}$$

So the $gcd(x-y,n)$ is a non-trivial factor:

$$gcd(96884638254 - 14, 642401) = 1129$$

And n can be factored as:

$$n = 642401 = 1129 \times 569$$

Question 4

Given: $n = 537069139875071$ $x = 85975324443166$ $y = 462436106261$

We first check that $x \not\equiv \pm y (\text{mod } n)$. Which is trivial since both x and y $<$ n. Then since $x \neq y$ and $x \neq -y + n$, we know that the condition holds.

Now we can use the "gcd" function in Sage math to find the factors. By $gcd(x - y, n)$ We get the factor 9876469. so:

$$n = 537069139875071 = 9876469 \times 54378659$$

Question 5

We first implement the Fermat Factorization method which gives us both q and p. Then use them to obtain $\varphi(n)$ to find the inverse.

Code:

```
n = 152415787501905985701881832150835089037858868621211004433
R = IntegerModRing(n)

e = 9007
c = R(14107746176556950024119950561785467338839857433334142352 5)


def factor(n):
  for i in range(1, 1000000):
    cur = n + pow(i, 2)
    if cur.is_square():
      return (cur.sqrt() + i, cur.sqrt() - i)


p,q = factor(n)
phi_n = (p-1)*(q-1)
d = inverse_mod(e, phi_n)

m = c^d
print(m)
```

Output:

```
200809190014211302051800230119001415200019050321 1805
```