TCSS 581 Cryptology

Homework 5

Tianyi Li
Student ID: 1827924

Tianyi Li
Student #:1827924

Describe the extended Euclidean algorithm. Implement it in a programming language of your choice. Explain how one can use the Euclidian algorithm for computing multiplicative inverses in modular arithmetic. Use your implementation to compute the multiplicative inverse of 2 modulo 7919.

- The standard **Euclidean algorithm** is a method of computing the greatest common divisor (GCD) of two integers a and b, such that
$$a = b \cdot q + r, \qquad where\ 0 \le r < b$$
- And the q is called the quotient and r is the remainder.
- The GCD of the integers a and b, denoted by
$$gcd\ (a, b)$$
- which is the largest integer that can divide both a and b at the same time, without remainder.
- Using repeated application of the above divisions, the GCD of a and b can be found by, repeatedly divide the divisor by the remainder r until the remainder is 0.
- For example,
  - Let $a = 102$ and $b = 38$,
  - Then,
$$102 = 2 \cdot (38) + 26$$
$$38 = 1 \cdot (26) + 12$$
$$26 = 2 \cdot (12) + 2$$
$$12 = 6 \cdot (2) + 0$$
  - Since the last non-zero remainder that appeared is 2,
  - Therefore, the GCD of $a = 102$ and $b = 38$ is 2.

- There is a way to push the Euclidean algorithm a little further to achieve something more.
- It is called the **extended Euclidean algorithm**, by given two integers a and b, it can find the integers x and y such that
$$a \cdot x + b \cdot y = gcd\ (a, b)$$
- This is done by reversing the steps in the above Euclidean algorithm.
- Start by finding the GCD, it uses the numbers as variables until it ends up an expression that is a linear combination of our initial numbers.
- To illustrate from the previous example,
  - Start by the second last line,
$$26 = 2 \cdot (12) + 2$$
  - Rewrite it becomes,
$$2 = 26 - 2 \cdot (12) \qquad (*)$$
  - Replace 12 by its previous line (third last line) and rewriting it in the form just like $(*)$,
$$2 = 26 - 2 \cdot (38 - 1 \cdot 26)$$

- Collect like terms,

$$2 = 3 \cdot 26 - 2 \cdot 38$$

- Repeat the step for the next line,

$$2 = 3 \cdot (102 - 2 \cdot 38) - 2 \cdot 38$$

- Then,

$$2 = 3 \cdot 102 - 8 \cdot 38$$

- Therefore, in this case, $x = 3$ and $y = -8$.

- This algorithm can also produce **<u>modular multiplicative inverse</u>** of b modulo a.
- Continue on after using the extended Euclidean algorithm
- It will obtain the final equation

$$a \cdot x + b \cdot y = gc\,d(a, b) = r, \qquad where\ 0 \leq r < b$$

- Then, find a special case during the steps of the Euclidean algorithm, where

$$a \cdot x + b \cdot y = r = 1$$

- From here, we can deduce

$$1 \equiv b \cdot y \ \ mod\ a$$

- Thus, the integer y is the modular multiplicative inverse of b modulo a.
- To demonstrate with an example,
  - Let $a = 11$ and $b = 8$,
  - Then,

$$11 = 1 \cdot (8) + 3$$
$$8 = 2 \cdot (3) + 2$$
$$3 = 1 \cdot (2) + 1$$
$$2 = 2 \cdot (1)$$

  - Reversing from the second last steps,

$$1 = 3 - 1 \cdot (2)$$

$$\begin{aligned} 1 &= 3 - 1 \cdot \big(8 - 2 \cdot (3)\big) \\ &= 3 - \big(8 - 2 \cdot (3)\big) \\ &= 3(3) - 8 \end{aligned}$$

$$\begin{aligned} 1 &= 3 \cdot \big(11 - 1 \cdot (8)\big) - 8 \\ &= 11(3) - 8(4) \\ &= 11(3) + 8(-4) \end{aligned}$$

  - Therefore,

$$1 \equiv 8 \cdot (7) \ \ mod\ \ 11$$

  - Thus, the modular multiplicative inverse of $b = 8$ modulo $a = 11$ is $y = 7$.

- Here is the implementation of the extended Euclidean algorithm to compute modular multiplicative inverse:

```python
def ext_euc(a, b):

    if a == 0:
        return (b, 0, 1)

    else:
        gcd, y, x = ext_euc(b % a, a)
        x = x - (b // a) * y

        return (gcd, x , y)

def mod_multi_inv(b, a):

    gcd, x, y = ext_euc(a, b)

    if gcd != 1:
        print ("There is no inverse.")

    else:
        multi_inv = x % b

        print ("The Greatest Common Divisor of %s and %s is %s" % (a, b, gcd))
        print ("The Modular inverses of %s modulo %s is %s" % (a, b, multi_inv))
```

- Using above, the solution to the multiplicative inverse of $b = 2$ modulo $a = 7919$ is $\underline{3960}$.