# CSEP590 : Applied Cryptography: Homework 4

Karuna Sagar Krishna

May 6, 2023

## Task 1a

```
{ 1, 2, 3, 4, 6, 8, 9, 11, 12, 13, 16, 17,
 18, 19, 22, 23, 24, 26, 27, 29, 31, 32, 33, 34 }
```

As seen in class, $Z_{35}^*$ is defined as $\{a \in Z_{35} : gcd(a, 35) = 1\}$. So we need to remove numbers that are not coprime with 35. This can also be done by removing multiples of 5 or 7 since $35 = 5 * 7$. Further, Euler's function $\phi(35) = (5 - 1)(7 - 1) = 4 * 6 = 24$ provides the order of $Z_{35}^*$.

## Task 1b

Since 37 is odd prime, we know that $Z_{37}^*$ is cyclic. This follows from the lemma discussed in class - group $Z_n^*$ is cyclic iff $n = 2, 4, p^k, 2p^k$ where $p$ is an odd prime.

2 and 5 are generators of $Z_{37}^*$. I used the following code to check if $a \in Z_{37}^*$ is a generator. Note, the below code only works for prime $p$.

```python
def isgen(x, p):
    l = [1]
    a = 1
    for i in range(0, p-2):
        a = (a * x) % p
        l.append(a)
    print(l)
    l.sort()
    print(l)
    for i in range(0, p-1):
        if l[i] != i+1:
            return False
    return True

```

## Task 1c

Given a cyclic group $G = \langle g \rangle$ of order $n$, we discussed Euler's theorem in class which states that $g^n = e$ where $e$ is the identity element in $G$. We can write

$G = \{g^0, g^1 \ldots g^{n-1}\}$. Lets multiply each element of the group with $g$, then we'll have $G' = \{g.g^0, g.g^1 \ldots g.g^{n-1}\} = \{g^1, g^2 \ldots g^{n-1}, g^n\}$. Notice, that elements are shifted by one. So compared to $G$, we are missing $g^0 = e$ and we have a new element $g^n$. Since $G$ is a group lets say $g^i \in G$ is inverse of $g$. Suppose $i \neq n-1$ i.e. $0 \leq i < n-1$, then $e = g.g^i = g^{i+1}$, but this cannot be true because in $G$, there is a unique identity element $g^0$ and $i+1 \neq 0$. So the only choice is $i = n-1$, then $e = g.g^i = g.g^{n-1} = g^n$. This makes sense because $g^n$ is not part of $G$ and we lost $g^0$ in $G'$. So $g^n = e$ implies that $G = G'$. Hence $g^n = e$.

Next, using the provided hint, given a cyclic group $G = \langle g \rangle$ of order $n$, we want to prove that $g^y \in G$ where $y \in Z_n$ is a generator if and only if $y$ is coprime with $n$ i.e. $gcd(y, n) = 1$. **Case 1: $g^y$ is a generator** - lets assume that $y$ is not coprime with $n$ i.e. there exists $d > 1$ such that $y = dy'$ and $n = dn'$. We know $g^n = e$, so $g^{dn'} = e$. Raising both sides by $y'$, we get $g^{dn'y'} = e^{y'}$ which can be simplified to $(g^y)^{n'} = e$. But this is not possible because $g^y$ is a generator, and from above proof, we cannot have $(g^y)^{n'} = e$ for $n' < n$. Hence, $y$ is coprime with $n$. **Case 2: $y$ is coprime with $n$** - then we have $ya + nb = 1$ from the lemma introduced in class for some $a, b \in Z_n$. So $g^1 = g^{ya+nb} = g^{ya}g^{nb} = (g^y)^a(g^n)^b = (g^y)^a e^b = (g^y)^a$ i.e. we have $g = (g^y)^a$. For any $i \in Z_n$, we can write $g^i = (g^y)^{ai}$. So $g^i \in \langle g^y \rangle$. In other words, $G = \langle g \rangle = \langle g^y \rangle$. Hence $g^y$ is a generator of G.

Any generator $g' \in G = \langle g \rangle$. So $g' = g^i$ for some $i \in Z_n$. And from above we saw that $g^i$ is a generator iff $gcd(i, n) = 1$. By definition of Euler's function $\phi(n)$ is the number of elements that are coprime with $n$ in $Z_n$. Hence $\phi(n)$ is the number of generators in $G$.

A group with prime order implies that $|G| = n$ is prime. So from above, we know there are $\phi(n)$ generators. Since $n$ is prime, all numbers in $Z_n$ except 0 are coprime with $n$ i.e. $\phi(n) = n - 1$. Hence there are $n - 1$ generators in $G$. So any element $g \in G$ except $e$ is a generator of $G$.

## Task 1d

Given $n = pq$ for some primes $p$ and $q$ unknown to us and $\phi(n)$, the question how to we find out the factors $p$ and $q$. In class, we saw that $\phi(n) = \phi(pq) = (p-1)(q-1) = pq - p - q + 1$. This can be rewritten as $p + q = n - \phi(n) + 1$. If we can somehow find $p - q$ then we can easily solve for $p$ and $q$. Consider $(p+q)^2 = p^2 + q^2 + 2pq$, so we can compute $p^2 + q^2 = (p+q)^2 - 2n$. Now, $(p-q)^2 = p^2 + q^2 - 2pq = (p+q)^2 - 2n - 2n = (p+q)^2 - 4n$. So $p - q = \pm\sqrt{(p+q)^2 - 4n}$. Now we know $p + q$ and $p - q$ which can solved to get $p$ and $q$. Hence, we can find the factors in polynomial time.

## Task 2a

10 is $46^{-1} \mod 51$ i.e. $46 * 10 = 460 = 1 \mod 51$.

As we saw in class, the multiplicative inverse exists iff $gcd(46, 51) = 1$.

Assuming this is true, we can then compute the inverse using the extended Extended Euclidean algorithm since $51x + 46y = gcd(51, 46)$

$$
\begin{aligned}
&\text{iteration 1} \\
&\qquad a = 51, b = 46 \\
&\qquad a \mod b = 5 \\
&\text{iteration 2} \\
&\qquad a = 46, b = 5 \\
&\qquad a \mod b = 1 \\
&\text{iteration 3} \\
&\qquad a = 5, b = 1 \\
&\qquad a \mod b = 0 \\
&\qquad \text{return}(x = 0, y = 1) \\
&\text{back to iteration 2} \\
&\qquad \text{return}(x = 1, y = 0 - 1\lfloor 46/5 \rfloor) \\
&\qquad \text{return}(x = 1, y = -9) \\
&\text{back to iteration 1} \\
&\qquad \text{return}(x = -9, y = 1 + 9\lfloor 51/46 \rfloor) \\
&\qquad \text{return}(x = -9, y = 10)
\end{aligned}
$$

So $y = 10$ is the inverse of $46 \mod 51$. Further, note $gcd(51, 46) = gcd(46, 5) = gcd(5, 1) = gcd(1, 0) = 1$

## Task 2b

$x = 31$ is the solution to $46x = 49 \mod 51$. Since we saw above that $gcd(51, 46) = 1$, we know that there exists an inverse of $46 \mod 51$. And by definition of inverse $aa^{-1} = a^{-1}a = 1$

$$
\begin{aligned}
46x &= 49 \mod 51 \\
46^{-1}46x &= 46^{-1}49 \mod 51 \\
x &= 46^{-1}49 \mod 51 \\
x &= 10 * 49 \mod 51 \\
x &= 490 \mod 51 \\
x &= 31
\end{aligned}
$$

We can verify this by plugging $x = 31$ in the given equation to see if it satisfied. $46 * 31 = 1426 = 49 \mod 51$

## Task 2c

$$
\begin{aligned}
gcd(45,51) &= gcd(51, 45 \mod 51) \\
&= gcd(51,45) \\
&= gcd(45, 51 \mod 45) \\
&= gcd(45,6) \\
&= gcd(6, 45 \mod 6) \\
&= gcd(6,3) \\
&= gcd(3, 6 \mod 3) \\
&= gcd(3,0) \\
&= 3
\end{aligned}
$$

Since, $gcd(45, 51) = 3 \neq 1$, the inverse of $45 \mod 51$ does not exist as discussed in class.

## Task 2d

$45 * 2 = 90 = 39 \mod 51$, so indeed $x = 2$ is a solution for $45x = 39 \mod 51$. Since $45^{-1}$ does not exist as we saw above, we cannot use the technique we used to solve Task 2b above. This is not a contradiction with Task 2c; it only means we need to find a different way to solve the equation if a solution exists. One alternative is to brute force search $x \in Z_{51}$

## Task 3

As discussed in class, the RSA encryption scheme publishes public key $(n, e)$ and the encryption algorithm is $c = x^e \mod n$ where $x$ is the plain text. The new proposal constructs the plain text as $x = m.r$ where $m \in Z_n$ is the original message to communicate and $r \in Z_n$ is random. The decryption algorithm uses the secret key $(n, d)$ to retrieve the plain text as $c^d \mod n = x$. However, note plain text is $x = m.r$ and $r$ is not shared with the receiver. So it not possible for the receiver to retrieve $m$ from $x$. Hence, the new proposal is not correct.

Further, since $r$ is chosen from $Z_n$ and $m \in Z_n$, we know that $Z_n$ is not necessarily a group under multiplication. So there may not be an inverse of $r$ and hence it may not be possible to retrieve $m$ even if $r$ somehow shared with the receiver.

## Task 4a

RSA encryption security is dependent on the hardness of finding the factors of $n$. Though $n = pq$ for some primes $p$ and $q$, these primes are not known and finding them is hard. The proposal generates $p = (r - i)$ and $q = (r + j)$

starting from a single random number $r$. As the question describes primes are dense and hence $p$ and $q$ are close to $r$. An attacker can roughly guess $r$ as $r' = \sqrt{pq} = \sqrt{n}$. Then attacker tries to guess $p$ by computing $r' - 1, r' - 2, \ldots$ until a prime $p'$ is found and similarly guess $q$ by computing $r' + 1, r' + 2, \ldots$ until a prime $q'$ is found. If $p'q' = n$ then we found the factors. If not, we consider other candidates for $p'$ and $q'$. Since $p$ and $q$ are close to $r$, primes are dense and checking for prime can be done efficiently, we can find the factors in reasonable amount of time. Hence, this leads a insecure encryption scheme.

## Task 4b

Found the factors using the following python code which encodes the idea described above.

```
Found factors
p: 35123014591230139123011933120312
     223198716238123918231119382061,
q: 35123014591230139123011933120312
     223198716238123918231119382447
```

```python
from decimal import *
from sympy import *

def factor(n):
    with localcontext() as ctx:
        ctx.prec = 124/2
        r = int(Decimal(n).sqrt())

        pl = []
        for i in range (1, 1000):
            p = r - i
            if isprime(p):
                pl.append(p)

        ql = []
        for i in range (1, 1000):
            q = r + i
            if isprime(q):
                ql.append(q)

        for p in pl:
            for q in ql:
                if p * q == n:
                    print("Found factors p: {}, q: {}".
format(p, q))
                    return

factor(
123362615397576525683206910571962544945300500765564700092323333
```

```
    67120767290238588667397052161653352801437540471197470570083267)
```