The Diffie-Hellman Key Exchange

April 26, 2023

0.1 The Diffie-Hellman Key Exchange

First, we are required to functions for a prime number **p** and the group $\mathbb{Z}_{\mathbb{p}}^*$ ={1,2,3,...,p-1}.

The first function is called **generator(g,p)** and checks if g is a generator of $\mathbb{Z}_{\mathbb{p}}^*$. That means

$$g^n\ mod\ p = 1, 2, ..., p-1 - \ for\ some\ n$$

For doing it we are going to use the **Primitive Root Theorem**. Let p be a prime number. Then there is some $g \in \mathbb{F}_p^*$, the multiplicative group of nonzero elements of $\mathbb{F}_p$, so that all alements of $\mathbb{F}_p$ are powers of g. For such an element g, we have that:

$$g^{p-1} \equiv\ 1\ (mod\ p)$$

$$g^r \not\equiv\ 1\ (mod\ p)\ for\ 1 \leq r \leq p-2$$

[1]:
```
def generator(g,p):
    if p<2:
        return False

    if pow(g,p-1,p)!=1:
        return False

    for i in range(2,p-1):
        if pow(g,i,p)==1:
            return False

    return True
```

1. The first prime numer is 2, so if p<2 we can return False.
2. If $p \geq 2$, we first check if g is relatively prime to p. If they are not relatively prime, then g cannot be generator of $\mathbb{Z}_{\mathbb{p}}^*$, so we can return false (First condition from theorem).
3. Finally, we check if g is a generator of $\mathbb{Z}_{\mathbb{p}}^*$ by computing `pow(g,i,p)` $\forall\ i \in$ {2,...,p-1}. If any of these values are equal to 1, then g is not a generator of $\mathbb{Z}_{\mathbb{p}}^*$ because it would be generaiting a cyclic subgroup, so we can return false (Second condition from theorem).
4. We return true in case g pass the previous conditions.

Note: We do not check if p is a prime number. We suppose the user introduces it correctly.

[2]:
```
generator(2,47)
```

[2]: False

[3]:
```
generator(5,47)
```

[3]: True

The second function is called **euklid(a,p)** and allows us to calculate a^{-1} in $\mathbb{Z}_p^*$. We know that

$$a \cdot a^{-1}\ (mod\ p) \equiv 1$$

[4]:
```
def euklid(a,p):
    if a%p==0:
        raise ValueError("a is divisible by p")

    for i in range(1,p):
        if pow(a*i,1,p)==1:
            return i

    raise ValueError("not multiplicative inverse found")
```

1. First, we check if a is divisible by p, because, in that case, a does not have multiplicative inverse in $\mathbb{Z}_p^*$
2. Using that the third parameter or `pow` is the module, we compute $(a \cdot i)^1\ (mod\ p) \equiv a \cdot i\ (mod\ p)\ for\ 1 \leq i \leq p-1$. If it is equal to 1, then i=a^{-1} and we can return i.
3. If we do not find the inverse, we return a error.

Note: We do not check if p is a prime number. We suppose the user introduces it correctly.

[5]:
```
euklid(2,5)
```

[5]: 3

[6]:
```
euklid(4,5)
```

[6]: 4

[7]:
```
euklid(3,5)
```

[7]: 2

Now, we are going to use this functions to find x in **Diffie-Hellman Key Exchange** for prime p=1117, g=6(generator in $\mathbb{Z}_p^*$) and key h=527($h = g^x\ mod\ p$)

So, we are required to solve

$$527 = 6^x\ mod\ 1117$$

First, we need to check if 6 is a generator of $\mathbb{Z}^*_{1117}$. This condition is necessary, since otherwise we cannot guarantee that the power 6 raised to x can take the value 527. For this, we use the function `generator`:

[8]:
```
generator(6,1117)
```

[8]: `True`

Since we are working in $\mathbb{Z}^*_{1117}$, we know that every element has multiplicative inverse, so we can multiply each member by the multiplicative inverse of 527 in $\mathbb{Z}^*_{1117}$:

$$527 \cdot 527^{-1} = 6^x \cdot 527^{-1} \; mod \; 1117$$

and we would get that:

$$1 = 6^x \cdot 527^{-1} \; mod \; 1117$$

Using the `euklid` function, we can get the multiplicative inverse of 527 in $\mathbb{Z}^*_{1117}$:

[9]:
```
euklid(527,1117)
```

[9]: `195`

So, the equation we have to solve is:

$$1 = 6^x \cdot 527^{-1} \; mod \; 1117 \implies 1 = 6^x \cdot 195 \; mod \; 1117$$

Since 1117 is a small prime number, we can solve it by looping through elements of $\mathbb{Z}^*_p$ and checking if the value verifies the equation.

[10]:
```
for i in range(1,1117):
    a=pow(6,i) #a=6^i
    if pow(a*195,1,1117)==1: #(195*6^i)^1 mod 1117
        print(i)
```

```
123
```

So, the solution is **x=123**.