MATH 470: Communications and Cryptography

Homework 9

Due date: 15 November 2023 *Name: Huy Lai*

Problem 1. Solve the discrete logarithm problem $10^x = 106$ in the finite field $\mathbb{F}_{811}$ by finding a collision among the random powers 10^i and $106 \cdot 10^i$ that are listed in Table 5.17.

Solution:
From the table we see that

$$10^{234} = 106 \cdot 10^{399} = 304 \quad \text{in} \quad \mathbb{F}_{811}$$

Hence

$$10^{234} \cdot 10^{-399} = 10^{-165} = 10^{645} = 106 \quad \text{in} \quad \mathbb{F}_{811}$$

Problem 2. Program Pollard's ρ algorithm with $f(x) = x^2 + 1$ and $x_0 = y_0 = 0$, and use it to factor the following numbers. In each case, give the smallest value of k such that g_k is a non trivial factor of N and print the ratio $\frac{k}{\sqrt{N}}$

1. $N = 2201$
2. $N = 9409613$
3. $N = 1782886219$

Solution:

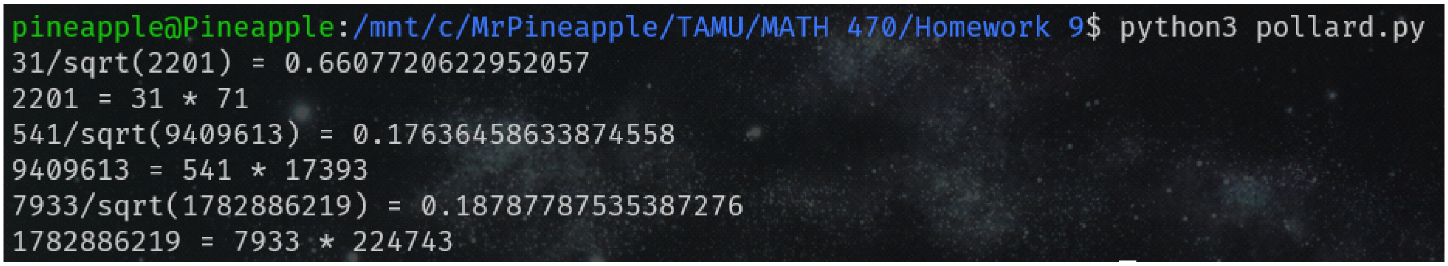
```python
from numpy import gcd
from typing import Callable


def pollard_rho(n: int, seed: int = 2, f: Callable[[int], int] = lambda x: x * x
    x, y = seed, seed
    d = 0
    while (not (1 < d < n)):
        x = f(x) % n
        y = f(f(y)) % n
        d = gcd(x - y, n)
    return d


def main():
    nums = [2201, 9409613, 1782886219]

    for N in nums:
        factor = pollard_rho(N, 0, lambda x: x * x + 1)
        other_factor = N // factor
        print(f"{factor}/sqrt({N}) = {factor * pow(N, -0.5)}")
        print(f"{N} = {factor} * {other_factor}")


if __name__ == "__main__":
    main()
```



```
pineapple@Pineapple:/mnt/c/MrPineapple/TAMU/MATH 470/Homework 9$ python3 pollard.py
31/sqrt(2201) = 0.6607720622952057
2201 = 31 * 71
541/sqrt(9409613) = 0.17636458633874558
9409613 = 541 * 17393
7933/sqrt(1782886219) = 0.18787787535387276
1782886219 = 7933 * 224743
```

Figure 1: Output

For Question 3 through 6 consider the following protocol (Schnorr Signature):

Let p be a large prime of the form $p = 2q + 1$ where q is prime.
Let $g \in (\mathbb{Z}/p\mathbb{Z})^*$ be an element of order q.
Let $\langle g \rangle$ denote the set $\{1, g, g^2, g^2, \cdots, g^{q-1} \mod p\}$. (note that g is not a primitive root $\mod p$, and $\langle g \rangle$ only contains half the numbers in $(\mathbb{Z}/p\mathbb{Z})^*$)
Assume that discrete logarithm problem for p, g is hard. In other words, assume that there is no efficient algorithm that can compute, for any given $A \in \langle g \rangle$, an integer a such that $g^a \equiv A \mod p$.
Let $A \in \langle g \rangle$.
Peggy wants to convince Victor that she knows an integer a such that $g^a \equiv A \mod p$.

- Peggy chooses a random integer $c \in \mathbb{Z}/q\mathbb{Z}$, computes $C = g^c \mod p$, and sends **commitment** C.
- Victor chooses a random integer $h \in \mathbb{Z}/q\mathbb{Z}$, and sends **challenge** h.
- Peggy computes $r = c + ah \mod q$, and sends **response** r.
- Victor **accepts** if $g^r \equiv C \cdot A^h \mod p$, and **rejects** otherwise.

Problem 3. Let x be an integer with $x \not\equiv 0 \mod q$. Prove that there exists an integer y such that $g^{xy} \equiv g \mod p$

Solution:
The proof is as follows

Proof. Since g has order q, we get that

$$g^q \equiv 1 \mod p$$

We can rewrite $g^{xy} \equiv g \mod p$ as

$$g^{xy-1} \equiv 1 \mod p$$

Since q is the order, by proposition we know that $q \mid (xy-1) \to \exists k \in \mathbb{Z}, xy = kq + 1$.
We can plug this into the original equation we get

$$\begin{aligned} g^{xy-1} &\equiv 1 \mod p \\ g^{(kq+1)-1} &\equiv 1 \mod p \\ (g^q)^k &\equiv 1 \mod p \\ 1^k &\equiv 1 \mod p \end{aligned}$$

Therefore we can find a y such that $g^{xy} \equiv g \mod p$.
This y can be found by solving the congruence $xy \equiv 1 \mod q$ □

Problem 4. Prove that the above protocol satisfies **completeness**; that is, prove that if Peggy does know a and if both parties follow the protocol, then Victor always accepts.

Solution:
We can verify that g^r is sufficient enough for Victor to verify as follows

$$\begin{aligned} g^r &\equiv g^{c+ah} \\ &\equiv g^c \cdot (g^a)^h \mod p \\ &\equiv C \cdot A^h \mod p \end{aligned}$$

Since the discrete log problem is assumed hard for g, p, Victor can have confidence that Peggy did not calculate a and instead knows a.

Problem 5. A transcript is a tuple (C, h, r) where $C \in \langle g \rangle, h \in \mathbb{Z}/q\mathbb{Z}$. A transcript (C, h, r) is valid if $g^r \equiv C \cdot A^h \mod p$. Prove that if you are given two valid transcripts (C, h_1, r_1) and (C, h_2, r_2) with $h_1 \not\equiv h_2 \mod q$, then you can efficiently compute a. (This property is called special **soundness**).

Solution:
From the valid certificates we can generate the following system of congruence

$$\begin{aligned} g^{r_1} &\equiv C \cdot A^{h_1} \mod p \\ g^{r_2} &\equiv C \cdot A^{h_2} \mod p \end{aligned}$$

Dividing these two equivalences gives

$$\begin{aligned} g^{r_1 - r_2} &\equiv A^{h_1 - h_2} \mod p \\ &\equiv (g^a)^{h_1 - h_2} \mod p \\ r_1 - r_2 &= a(h_1 - h_2) \end{aligned}$$

Solving this equation gives

$$a = \frac{r_1 - r_2}{h_1 - h_2}$$

Problem 6. Prove that the above protocol satisfies honest verifier zero knowledge; that is, prove that there is an efficient algorithm which, without knowledge of a, can produce transcripts that are indistinguishable from transcripts of honest interactions between Peggy (who knows a) and Victor.

Solution:
To generate transcripts that are indistinguishable from honest interactions between Peggy and Victor, we need to generate tuples (C, h, r) that are distributed identity to those send between Peggy and Victor.

We can generate this transcript by running the protocol algorithm in reverse.

1. Let $r \in \mathbb{Z}/q\mathbb{Z}$ be a uniformly random integer.
2. Let $h \in \mathbb{Z}/q\mathbb{Z}$ be a uniformly random integer.
3. Let $C = \dfrac{g^r}{A^h}$

We can verify the following this process, a verify could compute the following

$$\begin{aligned} g^r &\equiv C \cdot A^h \mod p \\ &\equiv \frac{g^r}{A^h} \cdot A^h \mod p \\ &\equiv g^r \end{aligned}$$

This means that with an exchange constructed with this method, a reader of a transcript (C, h, r) can be independently verified as accepted relations. However, this does not prove to the transcript reader that Peggy actually knows the secret.