MATH 470: Communications and Cryptography

Homework 6

Due date: 12 October 2023 *Name: Huy Lai*

Problem 1. Let $N = pq$ be the product of two distinct primes. Let e, d be integers such that $ed \equiv 1 \mod \phi(N)$. Prove that for every integer m, we have $m^{ed} \equiv m \mod N$. (note: If $\gcd(m, N) = 1$, then this follows from the Proposition from class that $m^{\phi(N)} \equiv 1 \mod N$. The purpose of this problem is to handle the other cases, thus proving that RSA decryption always returns the original message.)

Solution:
The proof is as follows

Proof. We can use the Chinese Remainder Theorem to show that the two congruences

$$m^{ed} \equiv m \mod p \quad \text{and} \quad m^{ed} \equiv m \mod q$$

hold.
We can use the fact that $\gcd(p, q) = 1$ and the Chinese Remainder Theorem to show that $m^{ed} \equiv m \mod N$.

By proposition, $ed \equiv 1 \mod \phi(N) \to \phi(N) \mid (ed - 1)$
By definition of divisibility, $\exists k \in \mathbb{Z}$ such that $k \cdot \phi(N) = ed - 1 \to ed = k \cdot \phi(N) + 1$

If $m \equiv 0 \mod p$, then certainly $m^{ed} \equiv m \mod p$.
If $m \not\equiv 0 \mod p$, then by Fermat's Little Theorem, $m^{p-1} \equiv 1 \mod p$.
From this we get the following:

$$m^{ed} \equiv m^{k \cdot \phi(N)+1} \equiv m(m^{p-1})^{k(q-1)} \equiv m \cdot 1^{k(q-1)} \equiv m \mod p$$

Therefore, $m^{ed} \equiv m \mod p$ holds for all integers m.
Replacing p with q in the previous argument shows that $m^{ed} \equiv m \mod q$ holds for all integers m. $\square$

Problem 2. For each part, use the data provided to find values of a and b satisfying $a^2 \equiv b^2 \mod N$, and then compute $\gcd(N, a-b)$ in order to find a nontrivial factor of N, as we did in Examples 3.37 and 3.38. $N = 2525891$

Solution:
First we try

$$\begin{aligned}1591^2 \cdot 3182^2 &\equiv (2 \cdot 5 \cdot 7^2 \cdot 11)(2^3 \cdot 5 \cdot 7^2 \cdot 11) \mod N \\ &= (2^2 \cdot 5 \cdot 7^2 \cdot 11)^2 \\ &= 10780^2 \\ \gcd(N, 1591 \cdot 3182 - 10780) &= 2525891\end{aligned}$$

Next we try

$$\begin{aligned}1591^2 \cdot 4773^2 &\equiv (2 \cdot 5 \cdot 7^2 \cdot 11)(2 \cdot 3^2 \cdot 5 \cdot 7^2 \cdot 11) \mod N \\ &= (2 \cdot 3 \cdot 5 \cdot 7^2 \cdot 11)^2 \\ &= 16170^2 \\ \gcd(N, 1591 \cdot 4773 - 16170) &= 2525891\end{aligned}$$

Next we try

$$\begin{aligned}3182^2 \cdot 4473^2 &\equiv (2^3 \cdot 5 \cdot 7^2 \cdot 11)(2 \cdot 3^2 \cdot 5 \cdot 7^2 \cdot 11) \\ &\equiv (2^2 \cdot 3 \cdot 5 \cdot 7^2 \cdot 11)^2 \\ &\equiv 32340^2 \mod N \\ \gcd(2525891, 3182 \cdot 4473 - 32340) &= 2525891\end{aligned}$$

Finally we try

$$\begin{aligned}1591^2 \cdot 5275^2 \cdot 5401^2 &\equiv (2 \cdot 5 \cdot 7^2 \cdot 11)(2^3 \cdot 3^6 \cdot 7)(2^4 \cdot 3^2 \cdot 5^3 \cdot 7 \cdot 11) \mod N \\ &= (2^4 \cdot 3^4 \cdot 5^2 \cdot 7^2 \cdot 11)^2 \\ &= 17463600^2 \\ \gcd(N, 1591 \cdot 5275 \cdot 5401 - 17463600) &= 1637\end{aligned}$$

Problem 3. Let $L(N) = e^{\sqrt{\ln N \ln \ln N}}$ as ususal. Suppose that a computer does one billion operations per second.

1. How many seconds does it take to perform $L(2^{100})$ operations?
2. How many hours does it take to perform $L(2^{250})$ operations?
3. How many days does it take to perform $L(2^{350})$ operations?
4. How many years does it take to perform $L(2^{500})$ operations?
5. How many years does it take to perform $L(2^{750})$ operations?
6. How many years does it take to perform $L(2^{1000})$ operations?
7. How many years does it take to perform $L(2^{2000})$ operations?

Solution:
The length of the operations is as follows

1. $2.780 \cdot 10^{-2}$ sec
2. $2.657 \cdot 10^{0}$ hr
3. $8.224 \cdot 10^{1}$ days
4. $1.129 \cdot 10^{3}$ years
5. $1.833 \cdot 10^{8}$ years
6. $5.548 \cdot 10^{12}$ years
7. $9.846 \cdot 10^{35}$ years

Problem 4. Implement Pollard's $p-1$ algorithm on a computer to factor N:

$$N = 340510176929609558738506407941198102081020749940944635553628097992090306553579338501$$

Solution:
The algorithm is as follows

```
from numpy import gcd


def pollard(N):
    a, i = 2, 2

    # iterate till a prime factor is obtained
    while (True):
        a = pow(a, i, N)
        g = gcd(a - 1, N)
        if (g > 1):
            return g
        i += 1


def main():
    N = 34051017692960955873850640794119810208102074994094463555362809799209030

    factor = pollard(N)
    other_factor = N // factor
    print(f"Factor found: {factor}")
    print(f"Other factor: {other_factor}")


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

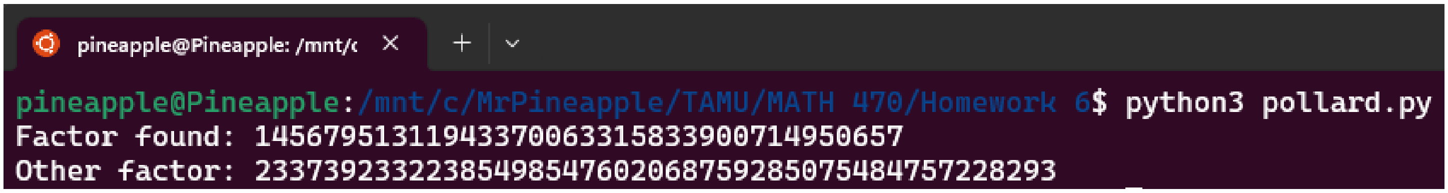

Figure 1: Output

Problem 5. Let p be an odd prime and let g be a primitive root modulo p. Then any number a is equal to some power of $g \mod p$, say $a \equiv g^k \mod p$. Prove that a has a square root modulo p if and only if k is even.

Solution:
We first prove that if k is even, then a has a square root modulo p.

Proof. Assume k is an even integer such that $k = 2j$. Then

$$a \equiv g^k \equiv g^{2j} \equiv (g^j)^2 \mod p$$

$\therefore$ a has a square root modulo p □

Next we prove that if a has a square root modulo p then k is even.

Proof. Assume a is a square such that $a \equiv b^2 \mod p$
Since g is a primitive root, $b \equiv g^i \mod p$ for some integer i. Then

$$g^k \equiv a \equiv b^2 \equiv (g^i)^2 \equiv g^{2i} \mod p$$

Multiplying both sides by the inverse of g^{2i} gives us:

$$g^{k-2i} \equiv 1 \mod p$$

The fact that g is a primitive root implies that $p-1 \mid (k-2i)$.
We know that $p-1$ is even. Therefore $2 \mid (p-1)$
By proposition, if $2 \mid (p-1)$ and $p-1 \mid (k-2i)$, then $2 \mid (k-2i)$.
Additionally, by proposition, $2 \mid (k-2i) \to 2 \mid k$ and $2 \mid 2i$.
Therefore, k is even. □

Problem 6. Let k be a positive integer. What is the number of square roots of $1 \mod 2^k$? In order words, determine the number integer x with $0 \leq x \leq 2^k - 1$ that satisfy $x^2 \equiv 1 \mod 2^k$.

Solution:
When $k = 1$, there is only one solution: $x = 1$
When $k = 2$, there are two solutions: $x = 1, x = 3$

We will now handle when $k \geq 3$
Let x be am integer such that $x^2 \equiv 1 \mod 2^k$.
Rearranging gives us $(x-1)(x+1) \equiv 0 \mod 2^k$.

Since $x \equiv 1 \mod 2$, $x \pm 1 \equiv 0 \mod 2$.
Because of this $\gcd(x-1, x+1) = 2$, which implies that one of $x \pm 1 \equiv 2 \mod 4$.

Since $2^k \mid (x-1)(x+1)$, one of $x \pm 1$ is divisible by 2^{k-1}, which means $x \pm 1 \equiv 0$ or $2^{k-1} \mod 2^k$.
If $x \pm 1 \equiv 0 \mod 2^k$, then we get two solutions $x \equiv 1 \mod 2^k$ and $x \equiv -1 \mod 2^k$
If $x \pm 1 \equiv 2^{k-1} \mod 2^k$, then we get two more solutions $x \equiv 2^{k-1} - 1 \mod 2^k$ and $x \equiv 2^{k-1} + 1 \mod 2^k$

Therefore, when $k \geq 3$, there are four solutions: $x = 1, x = 2^{k-1} - 1, x^{k-1} + 1, x = 2^k - 1$

Problem 7. Let p be an odd prime and let b be an integer not divisible by p. Prove that for every positive integer e, the congruence $X^2 \equiv b \mod p^e$ has either 0 or 2 solutions in $\mathbb{Z}/p^e\mathbb{Z}$.

Solution:
The proof is as follows

Proof. Base Case: $e = 1$
$X^2 \equiv b \mod p^1$ has 0 or 2 solutions is proven by Homework 3 Question 4.

If $X^2 \equiv b \mod p^1$ has a solution, then $X^2 \equiv b \mod p^2$ also has a solution is proven by is proven by Homework 4 Question 5.

Inductive Hypothesis
Assume that if $X^2 \equiv b \mod p^e$ has a solution in $\mathbb{Z}/p^{e+1}\mathbb{Z}$ then $X^2 \equiv b \mod p^{e+1}$ also has a solution $\forall e \geq 1$.
Additionally, assume that $X^2 \equiv b \mod p^e$ has two unique solutions.

Inductive Step
Without loss of generality, let α be a solution to the congruence $X^2 \equiv b \mod p^e$.
By the inductive hypothesis, we also have a solution β that solve the congruence $X^2 \equiv b \mod p^{e+1}$.
From homework 4 question 5, we know we can generate the solution β from α by adding a multiple of p^e.

A similar logic can be applied to the other solution to the congruence $X^2 \equiv b \mod p^e$. □