Kevin Chau
23816929
CS70-zh

Homework 5

Problem 1

1. $(569)^{570^{571}} \bmod 571 = (569)^{(570^{571}) \bmod 570} \bmod 571$

since 571 is prime, we can use Fermat's little Theorem.

$570^{571} = \underbrace{(570)(570)\cdots(570)}_{571 \text{ times}}$ is divisible by

~~by~~ 570, so $570^{571} \bmod 570 = 0$

$\Rightarrow 569^{570^{571} \bmod 570} \bmod 571 = 569^{0} \bmod 571$

$= \boxed{1 \bmod 571}$

2. $5^{8^{11}} \pmod 9$

using a calculator, we can brute force calculate the following:

$5^6 \bmod 9 = 15625 \bmod 9 = 1 \bmod 9$

$8^{11} \bmod 6 = 8589934592 \bmod 6 = 2 \bmod 6$

$\Rightarrow 8^{11} = 6k+2$ for some $k \in \mathbb{Z}$

$\Rightarrow 5^{8^{11}} = 5^{6k+2}$

$\Rightarrow 5^{6k+2} \bmod 9 = 5^{6k} 5^2 \bmod 9$

$= (5^{6k} \bmod 9)(25 \bmod 9) \bmod 9$

$= (5^6 \bmod 9)^k (7 \bmod 9)$

$= (1^k \bmod 9)(7 \bmod 9)$

$= 1 \cdot 7 \bmod 9 = \boxed{7 \bmod 9}$

3. $7^{2014} \bmod 31 = 7^{2014 \bmod 30} \bmod 31$

$= 7^4 \bmod 31$

$= 2401 \bmod 31$

$= \boxed{14 \bmod 31}$

Problem 2

1. $N = p$ where p is 1024 bits

$E(x) = x^e \bmod p = y$

$D(y) = D(E(x)) = y^d \bmod p = x$

e is relatively prime to $p-1$, so $\gcd(e, p-1) = 1$

d is still the inverse of $e \bmod p-1 \Rightarrow d = e^{-1} \bmod p-1$

$\Rightarrow de = 1 \bmod p-1$

You can find d with egcd $(N-1, e)$, which has to equal 1 since $N-1$ & e are coprime

$$\text{egcd}(N-1, e) = 1 = a(N-1) + be \pmod{N-1}$$
$$1 = be \pmod{N-1}$$
$$\Rightarrow b = e^{-1} = d \checkmark$$

Theorem: $D(E(x)) = x \bmod N$

Proof: We need to show $(x^e)^d = x \bmod N$ for all $x \in [0, N-1]$

since $ed = 1 \bmod (p-1)$

$ed = 1 + k(p-1)$ for some $k \in \mathbb{Z}$

$\Rightarrow x^{ed} - x = x^{1+k(p-1)} - x = x(x^{k(p-1)} - 1)$ should equal 0

Case: x is a multiple of p. So $x(x^{k(p-1)} - 1) \bmod p = 0 \checkmark$

Case: x is not a multipe of p. so $x \equiv 0 \bmod p$.

But $x^{p-1} = 1 \bmod p$ (Fermat's little Theorem)

$\Rightarrow (x^{p-1})^k \equiv 1 \bmod p$

$$\Rightarrow (x^{p-1})^k - 1 \equiv 0 \bmod p \checkmark$$

2. Yes. Eve can compute d easily with just the egcd, as explained above. Since e and $N-1$ are coprime, $\text{egcd}(e, N-1) = 1 = a(N-1) + be$

$$1 \equiv be \bmod (N-1)$$
$$\Rightarrow d = e^{-1} = b \text{ from EGCD algorithm.}$$

The EGCD algorithm, like the GCD, decreases the problem size at least by a factor of two every two recursive calls (see Note 5), so it takes at most $2n$ calls to stop. For p with $n = 1024$ bits, it would take about 2000 calls, which is $O(n)$.

3. Given d, Eve can recover x using the mod exponent algorithm from lecture note 5. The decryption function $D(y) = y^d \bmod p$ could take a very long time to compute for very large d, so we can exploit repeated squaring in the mod-exp algorithm. For repeated squaring, the recursive call reduces the exponent by a factor of 2, so the iterations count is equal to the bits n in d. The computation is $O(n)$

4. Since the total computation is just finding d & recovering x with egcd and mod-exp, it takes $O(n_p) + O(n_d)$ time for eve to decrypt the message, where n_p = bits in p and n_d = bits in d. Assuming d and p are roughly the same bit length, the total computation scales with the problem size. So for $p = 1024$, it would take about 1000 operations, which on any computer would not take very long (certainly not as long as 2^{1024}). Eve can recover the message quickly

Problem 3 $\phi(n) = |\{i : 1 \le i \le n, \gcd(n,i) = 1\}|$

For m, n such that $\gcd(m,n) = 1$, $\phi(mn) = \phi(m) \cdot \phi(n)$

1. Let p be a prime number.

$\phi(p) = |\{i : 1 \le i \le p, \gcd(p,i) = 1\}|$

Since all integers less than p are coprime with p, and there are $p-1$ numbers less than p and greater than or equal to 1,

$\boxed{\phi(p) = p-1}$

3. Let p be a prime number and $a \in \mathbb{Z}^+$, $a < p$.

What is $a^{\phi(p)} \bmod p$? From part 1, $\phi(p) = p-1$

$a^{\phi(p)} \bmod p = a^{p-1} \bmod p = \boxed{1 \bmod p}$

2. Let p be prime and k be a positive integer.

What is $\phi(p^k)$?

This is just all the numbers up to p^k minus all the multiples of p. For p^k, there are p^{k-1} multiples of p less than p^k.

So $\boxed{\phi(p^k) = p^k - p^{k-1}}$

4. b with prime factors $p_1, p_2, p_3, \cdots, p_k$. $b = p_1^{\alpha_1} \cdot p_2^{\alpha_2} \cdots p_k^{\alpha_k}$

$\gcd(a,b) = 1$ given, show $a^{\phi(b)} \equiv 1 \bmod p_i$.

Since a is coprime with b, it is also coprime with all factors of b, and all $p_i^{\alpha_i}$ are coprime with $p_j^{\alpha_j}$ for $j \ne i$. Using the theorem mentioned earlier:

$$a^{\phi(p_1^{\alpha_1} \cdot p_2^{\alpha_2} \cdots p_k^{\alpha_k})} = a^{\phi(p_1^{\alpha_1})\phi(p_2^{\alpha_2})\cdots\phi(p_k^{\alpha_k})}$$
$$= a^{(p_1^{\alpha_1} - p_1^{\alpha_1 - 1})(p_2^{\alpha_2} - p_2^{\alpha_2 - 1})\cdots(p_k^{\alpha_k} - p_k^{\alpha_k - 1})}$$
$$= a^{(p_1^{\alpha - 1})(p_1 - 1)\cdots(p_k^{\alpha_k - 1})(p_k - 1)}$$
$$= a$$

For any i, this can be rewritten $a^{k(p_i - 1)}$ for $k \in \mathbb{Z}$

$$a^{k(p_i-1)} = (a^{p_i - 1})^k \equiv 1^k \bmod p_i \equiv \boxed{1 \bmod p_i} \text{ using FLT}$$

Problem 4

1. No matter what you flip, your friend can make you stand in line. If you flip heads, then your friend can decide heads means you have to stand in line. If you flip tails, your friend can also decide tails means you stand in line. It's not a fair methood since your frend decides after you toss.

2. Before you toss the coin, your friend can decide on the rules first, and then encrypt them with RSA, using a public key of his choosing. Your friend can then send you the encrypted message and public key, but you won't know the rules before or after you toss the coin. So you toss the coin and tell your friend the actual result, since you have no incentive to lie without knowing the assignment of heads and tails. Once your friend knows what the coin toss result was, he can send you the private key to decrypt the message. Since RSA encryption is a bijection, there's no way for the decryption to give you anything other than the original message, and since the rules were decided before the toss, there's no way your friend is deciding unfairly, even if you choose to lie about the toss result.

3. The problem is that since everyone knows the set only has two elements, its easy for any one to figure out the message even without doing a proper decryption using the private key. Here's how: Suppose I'm trying to send my friend the result of the coin toss - "heads" - with a public key that he knows and you know. Even without the private key, my friend can guess the coin toss and encrypt his guess with the public key. If his encrypted message is the same as the one I sent, he correctly guessed heads. If its diffcrent, he knows he incorrectly guessed tails, and <u>knows</u> the message is heads. Anyone with the public key can "decrypt" the message if there are only two possible messages.
This can be fixed by making the message into a string, such as "I tossed heads", which increases the bits in the message and makes the set of possible messages so large that you can't use a brute force guess/check decryption.

4. According to professor Sahai on piazza, we can interpret this question as saying there are 2 groups, where 1 representative from either group must stand in line. For part 2, it would have been the same if the coin tosser encrypts the toss result. For $n > 2$, we can have the coin tosser encrypt the message and send it to everyone, including the people on his side. Since there are no rules yet, he has no reason to lie, and even if he does it won't affect the rules. All n people can verify that they have the same encrypted message, so the coin tosser isn't saying one thing to his group and another thing to the other. Now the otherside can make a decision on the rules, and once they do, the coin tosser can reveal the private key.

Problem 5. Write Your Own Problem -
Modular arithmetic & Basic algebra in mod math
The Chinese remainder theorem says that there exists a number number n when divided by given sequence of integers gives another sequence of given integers. In other words, suppose $n_1, n_2, \dots, n_k$ are positive integers and all coprime to eachother, Then given a sequence of integers $a_1, a_2, a_3, \dots, a_k$, there exists an x such that x solves:

$$x \equiv a_1 \mod n_1$$
$$x \equiv a_2 \mod n_2$$
$$\vdots$$
$$x \equiv a_k \mod n_k$$

Show that such a number exists for an example of $K=3$, and generate an algorithm that can determine x.

Solution: Consider $n_1 = 2$ $n_2 = 5$ $n_3 = 7$
and $a_1 = 2$ $a_2 = 3$ $a_3 = 6$
(chosen randomly)

$x \equiv 2 \mod 2$
$x \equiv 3 \mod 5$
$x \equiv 6 \mod 7$

Congruence classes:
$x \in \{0, 4, 6, 8, 10, 12, 14, 16, 18 \dots\}$
$x \in \{3, 8, 13, 18, 23, 28, \dots\}$
$x \in \{6, 12, 18, 24, 30, \dots\}$

$\boxed{x \equiv 18 \mod 70}$

Using algebra:
$x = 2 + 2t$
$x = 3 + 5s$
$x = 6 + 7u$

these can be substituted into the mod equations to get the same result.

Algorithm

We want to solve $x \equiv a_i \pmod{n_i}$ for $i = 1, \dots, k$

- Define $N = n_1 n_2 \dots n_k$
- n_i and N/n_i are always coprime. Using EGCD, we can find a_i and b_i such that $GCD(n_i, N/n_i) = 1 = a_i n_i + b_i N/n_i$

call $c_i = b_i N/n_i$

$\Rightarrow a_i n_i + c_i = 1$

so $c_i \bmod n_i = 1$. But for n_j where $j \neq i$

$c_i \bmod n_j = 0$

$$\Rightarrow c_i \equiv 1 \bmod n_i$$
$$c_i \equiv 0 \bmod n_j \quad j \neq i$$

Then x is just is just the sum of $a_i c_i$

$$x = \sum_{i=1}^{k} a_i c_i$$

6. Midterm question 3: For $p > 1$, prove $p-1$ is its own inverse

Direct Proof: $(p-1)(p-1) \bmod p$

$\equiv (-1)(-1) \bmod p$

$\equiv (-1)^2 \bmod p$

$\equiv 1 \bmod p$

$\Rightarrow (p-1)^2 \equiv 1 \bmod p$

Hence $p-1$ is its own inverse

7. Midterm Question 4: $21^{-1} \bmod 31$?

$21 \cdot a \equiv 1 \bmod 31 = \{1, 32, 63, 94, 125, 156\}$

$21 \times 3 = 63 \equiv 1 \bmod 31$

I got the inverse by listing multiples of 21, and looked for the first one that is congruent mod 31

$$\boxed{21^{-1} \bmod 31 = 3 \bmod 31}$$

8. Midterm Question 5

a)

Pokemon	Trainer		
A	2	3	1
B	1	3	2
C	3	2	1

Trainer	Pokemon		
1	A	B	C
2	C	A	B
3	B	C	A

Pokémon-optimal: {(A, 2), (B, 1), (C, 3)} (pokemon propose)
Trainer-optimal: {(1, A), (2, C), (3, B)} (trainers propose)

b)

A	2	3	1	4
B	1	3	2	4
C	3	2	1	4
D	4	2	1	3

1	~~D~~	A	B	C
2	~~D~~	C	B	A
3	~~D~~	B	C	A
4	~~A~~	D	B	C

Day		Proposals
1	A	4
	B	
	C	
	D	~~1~~ 2 ~~3~~
2	A	1 ~~4~~
	B	3
	C	
	D	2
3	A	1
	B	3
	C	
	D	~~2~~ 4
4	A	1
	B	3
	C	2
	D	4

Everyone gets rejected once

9. <u>Midterm Question 6</u>

$H_j = \sum_{i=1}^{j} \frac{1}{i} = 1 + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{j}$

<u>Prove by Induction</u>: $\sum_{j=1}^{n} H_j = H_1 + H_2 + \cdots + H_n = (n+1)H_n - n$

<u>Base Case</u>: $n=1 \Rightarrow \sum_{1}^{1} H_j = H_1 = (1+1)H_1 - 1$

$= 2H_1 - 1 = 2 \cdot 1 - 1 = \boxed{1}$ ✓

<u>Induction Hypothesis</u>: Assume true for $n=k$. So

$$\sum_{j=1}^{k} H_j = H_1 + H_2 + \cdots + H_k = (k+1)H_k - k$$

<u>Induction step</u>: Show that $k+1$ is true following the hypothesis

$$\sum_{j=1}^{k+1} H_j = \underbrace{\sum_{j=1}^{k} H_j} + H_{k+1} = \underbrace{(k+1)H_k - k} + H_{k+1}$$

$$= (k+1)H_k - k + \left(H_k + \frac{1}{k+1}\right)$$

$$= kH_k + H_k + H_k - k + \frac{1}{k+1}$$

$$= kH_k + 2H_k - k + \frac{1}{k+1}$$

$$= (k+2)H_k - \frac{k(k+1)}{k+1} + \frac{1}{k+1}$$

$$= (k+2)H_k + \frac{k+2}{k+1} - \frac{k(k+1)}{k+1} + \frac{1}{k+1} - \frac{k+2}{k+1}$$

$$= (k+2)\left(H_k + \frac{1}{k+1}\right) + \frac{1 - k^2 - k - k - 2}{k+1}$$

$$= (k+2)H_{k+1} - \frac{k^2 + 2k + 1}{k+1}$$

$$= (k+2)H_{k+1} - \frac{(k+1)\cancel{(k+1)}}{\cancel{k+1}}$$

$$= \boxed{(k+2)H_{k+1} - (k+1)}$$

Thus by induction $\forall n$ $\sum_{1}^{n} H_j = (n+1)H_n - n$

10. Midterm Question 7

If $P \Rightarrow Q$, then $Q \Rightarrow P$ $\boxed{\text{False}}$

Proof by counterexample:

truth table

P	Q	$P \Rightarrow Q$	$Q \Rightarrow P$
T	T	T	T
F	T	T	F
T	F	F	T
F	F	T	T

When P is false and Q is true, $P \Rightarrow Q$ but $Q \not\Rightarrow P$, so the statement is false

11. Midterm Question 8: If p is prime, then $(p-1) \equiv (p-1)! \bmod p$

Since p is prime, all numbers less than p are coprime with p, so $\{1, 2, \ldots, p-1\}$ all have gcd with p equal to 1. Hence all elements in $\{1, 2, \ldots, p-1\}$ have a multiplicative inverse mod p. We know 1 is its own inverse (in any mod), and we proved already that $p-1$ is its own inverse. Now since every integer less than p has an inverse mod p, we can reduce $(p-1)! = (p-1)(1)(p-1)(p-1)^{-1}(p-2)(p-2)^{-1}\ldots$

In order to pull $p-1$ & 1 out of $(p-1)!$,

$$= (p-1)(1)(1)(1)$$
$$= (p-1)(1) = p-1 \checkmark$$

We need to prove $(p-1)$ and 1 are the only numbers mod p that are there own inverse: suppose there is a number k such that

$$k^2 \equiv 1 \bmod p$$
$$\Rightarrow k^2 - 1 = 0 \bmod p$$
$$(k-1)(k+1) = 0 \bmod p$$
$$(k-1) \bmod p = 0 \quad (k+1) \bmod p = 0$$
$$\Rightarrow k = 1 \bmod p \qquad \Rightarrow k = -1 \equiv p-1 \pmod p$$

hence 1 and $p-1$ are the only numbers mod p that are self inverses.

12. Midterm Question 9

```
EGCD(x,y):
    if y>x:
        return EGCD(y,x)
    if y=0:
        return (x,1,0)
    else:
        (d,a,b) = ECGD(y,x-y)
        return (d, a, a-b)
```

<u>Proof of Termination</u>: If $y=0$, then the algorithm returns the correct base case, since $y=0$ is divisible by x $\forall x$ so $d = x(1) + 0(0)$. Any other input will make a recursive call. EGCD is intended for $x \geq y$ such that the recursive call always reduces the problem size or returns the base case. If $y=x$, then the next recursive call is the base case. If $y<x$, we already know that the recursive calls first argument y is less than x and $x-y$ is less than x. Since we are working with natural numbers, $y \leq x-1$, so the recursion reduces the first argument by atleast 1 every recursive call. Thus it takes at most x recursions to return to a base case. Since we have an upper bound in computations, the algorithm ends. The algorithm will also terminate for $y>x$ since the first "if" clause will return EGCD with switched arguments, and so by the proof above it ends.

<u>Correct EGCD</u>: The triple integers $(d, a, a-b)$ are the final return of the algorithm. It is correct by direct proof:

$$(d, b, a-b) \longrightarrow d = ax + (a-b)y$$

and since $EGCD(x,y) = EGCD(y, x-y)$

$$(d,a,b) \longrightarrow d = a(y) + b(x-y)$$
$$= ay + bx - y$$
$$= (a-b)y + bx$$

13. <u>Midterm Question 10</u>

$31x + 21y = 1010$ since $\gcd(31,21) = 1 = 21a + 31b$

for some $x, y \in \mathbb{Z}^+$

$\Rightarrow 21a + 31b = 1$

$\Rightarrow 21(3) + 3(-2) = 1$

$\times 1010 \Rightarrow 21(3030) + 3(-2020) = 1010$

$+k(21(-31) + 31(21) = 0)$

$21(\underline{y}) + 31(\underline{x}) = 1010$

k has to be less than 100... try a few values:

$$\begin{array}{r} 31 \\ \times 99 \\ \hline 279 \\ 2790 \\ \hline 3069 \end{array} \qquad \begin{array}{r} 31 \\ \times 98 \\ \hline 3038 \end{array} \qquad \begin{array}{r} 31 \\ \times 97 \\ \hline 3007 \checkmark \end{array} \qquad \begin{array}{r} 21 \\ \times 97 \\ \hline 147 \\ 1890 \\ \hline 2037 \end{array}$$

$21(3030) + 3(-2020) = 1010$

$+ 21(-3007) + 31(2037) = 1010$

$\Rightarrow \boxed{y = 23 \quad x = 17}$

$\boxed{21(23) + 31(17) = 1010}$

Midterm

14. Problem 11:

<u>Improvement Lemma for Troll Women:</u>

IF on day k troll W receives a proposal from M, on all days after k W has someone M^* that she likes at least as much as M on that day k itself.

<u>Proof by Contradiction:</u> Suppose for the sake of contradiction on day $j > k$ that W has M^- (or no one) that she scored less than M on day k. j is the first such day that W has such M^-. By the well ordering principle, on day $j-1$, someone better than M^-, call him M^*, who is at least as good as M (he could be M as well) proposed to W. Since W said "maybe" to M^*, her score for him is higher than any other proposer, and if there were others that proposed, M^*'s score increases by 5 points. If so, any man she likes less than M^* on day $j-1$ can never be with her, since there score can never increase as they automatically are rejected because on day j M^* will propose again. Thus M^- could never actually receive a maybe. Contradiction.