Homework 6

1. **RSA Assumption (5+12+5).** Consider RSA encryption scheme with parameters $N = 35 = 5 \times 7$.

   (a) Find $\varphi(N)$ and $\mathbb{Z}_N^*$.

   $\mathbb{Z}_N^* = \{ 1, 2, 3, 4, 6, 7, 9, 11, 12, 13, 16, 17, 18, 19, 22, 23, 24, 26, 27, 29, 31, 32, 33, 34\}$

   And since $\varphi(N) = \left|\mathbb{Z}_N^*\right|$, $\varphi(N) = 24$.

   (b) Use repeated squaring and complete the rows X, X^2, X^4 for all $X \in \mathbb{Z}_N^*$ as you have seen in the class (slides), that is, fill in the following table by adding as many columns as needed.

   Solution.

X	1	2	3	4	6	8	9	11	12	13	16	17
X^2	1	4	9	16	1	29	11	16	4	29	11	9
X^4	1	16	11	11	1	1	16	11	16	1	16	11

X	18	19	22	23	24	26	27	29	31	32	33	34
X^2	9	11	29	4	16	11	29	1	16	9	4	1
X^4	11	16	1	16	11	16	1	1	11	11	16	1

(c) Find the row X^5 and show that X^5 is a bijection from $\mathbb{Z}_N^*$ to $\mathbb{Z}_N^*$.
Solution.

X	1	2	3	4	6	8	9	11	12	13	16	17
X^4	1	16	11	11	1	1	16	11	16	1	16	11
X^5	1	32	33	9	6	8	4	16	17	13	11	12

X	18	19	22	23	24	26	27	29	31	32	33	34
X^4	11	16	1	16	11	16	1	1	11	11	16	1
X^5	23	24	22	18	19	31	27	29	26	2	3	34

X^5 is clearly a bijection since every element of X shows up exactly once.

2. **Answer to the following questions (7+7+7+7):**

   (a) Compute the three least significant (decimal) digits of $6251007^{1960404}$ by hand.

   The three least significant decimal digits of $6251007^{1960404}$ **is equivalent to** $6251007^{1960404} \pmod{1000}$. **We can further reduce the complexity of this problem by observing that** $6251007^{1960404} \pmod{1000} = 007^{1960404} \pmod{1000} = 7^{1960404} \pmod{1000}$.

   We learned about Euler's totient in class, and one theorem that will help us in this problem.
   For x and N that are relatively prime, $x^{\varphi(N)} = 1 \mod N$

   $\varphi(1000) = \varphi(2^3 * 5^3) = (2^{3-1} * 5^{3-1})(2-1)(5-1) = (2^2 * 5^2)(1)(4) = (100)(1)(400) = 400$
   $\varphi(1000) = 400$

   Therefore,
   $x^{400} = 1 \mod 1000$

   And since
   $1960404 \mod 400 = 4$

   We can conclude that $7^{1960404} \mod 1000 = 7^4 \mod 1000$.
   Using repeated squares, we can solve the problem ($\mod 1000$ **is abstracted out for simplicity):**

   $7^1 = \alpha_0 = 7^{2^0} = 7$
   $7^2 = \alpha_1 = 7^{2^1} = \alpha_0 * \alpha_0 = 49$
   $7^4 = \alpha_2 = \alpha_1 * \alpha_1 = 49 * 49 = 2401 = 401 \mod 1000$

   Therefore, the three least significant decimal digits of $6251007^{1960404} = 401$.

(b) **Is the following RSA signature scheme valid?(Justify your answer)**
$(r||m) = 24, \sigma = 196, N = 1165, e = 43$
Here, m denotes the message, and r denotes the randomness used to sign m and σ denotes the signature. Moreover, $(r||m)$ denotes the concatenation of r and m. The signature algorithm $Sign(m)$ returns $(r||m)^d \mod N$ where d is the inverse of e modulo $\varphi(N)$. The verification algorithm $Ver(m, \sigma)$ returns $((r||m) == \sigma^e \mod N)$.

When we run the $Ver(m, \sigma)$ function, we get the following:
$Ver(m, \sigma) = ((r||m) == \sigma^e \mod N)$
$= (24 == 196^{43} \mod 1165)$
$= (24 == 676)$
$= FALSE$

Since the verification function fails, this RSA signature scheme is INVALID.

(c) **Remember that in RSA encryption and signature schemes, $N = p \times q$ where p and q are two large primes. Show that in a RSA scheme (with public parameters N and e), if you know N and $\varphi(N)$, then you can find the factorization of N i.e. you can find p and q.**
From class, we know that:
$N = pq$
$\varphi(N) = (p-1)(q-1)$

Simplifying, $\varphi(N) = pq - p - q + 1$. Notice that we can substitute $N = pq$ into that equation. Simplifying further,

$\varphi(N) = N - p - q + 1$
$p + q = N - \varphi(N) + 1$

And since the entire right side of that final equation is known, p and q become much easier to find.

(d) **Consider an encryption scheme where $Enc(m) := m^e \mod N$ where e is a positive integer relatively prime to $\varphi(N)$ and $Dec(c) := c^d \mod N$ where d is the inverse of e modulo $\varphi(N)$. Show that in this encryption scheme, if you know the encryption of m_1 and the encryption of m_2, then you can find the encryption of $(m_1 \times m_2)^5$.**

First, we know that:
$(x \mod N * y \mod N) \mod N = xy \mod N$

Therefore, if we set the encryption of $m_1 = x$ and the encryption of $m_2 = y$, we can get the encryption of $(m1 \times m2)$ by calculating $(x * y) \mod N$. To clearly illustrate:

$((m_1^e \mod N) * (m_2^e \mod N)) \mod N = (m_1 * m_2)^e \mod N$

Therefore, we can find the encryption of $(m_1 \times m_2)^5$ by doing the following:

$$((m_1^e \mod N)^5 * (m_2^e \mod N)^5) \mod N$$

$$= ((m_1^{5e} \mod N) * (m_2^{5e} \mod N)) \mod N$$

$$= (m_1 * m_2)^{5e} \mod N$$

3. **Programming Assignment: Compute the Cube Root of a Large Integer.** (50 points)
   This problem requires you to find the cube-root of very large perfect cube integers (each number is roughly 30K bits in binary representation). The inputs shall be given using a text file `inputs.txt` containing five (roughly) 30K-bit numbers in binary representation. These numbers are separated by a new line character. Your program must output the cube roots of the five numbers, represented as binary and separated by a new line character, to a text file named as `outputs.txt`. Make sure that you follow the conventions, otherwise you will get zero credit. You can use Java, Python, C, or SageMath. Turn in your code and the output file via Gradescope.

Collaborators: