Homework 6 (Solution)

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

   (a) Find $\varphi(N)$ and $\mathbb{Z}_N^*$.
   Solution.
   Recall that for $N = p \cdot q$, we have $\varphi(N) = (p-1)(q-1)$, where p and q are prime numbers. Thus $\varphi(35) = (5-1)(7-1) = 24$.

   $\mathbb{Z}_{35}^* = \{1, 2, 3, 4, 6, 8, 9, 11, 12, 13, 16, 17, 18, 19, 22, 23, 24, 26, 27, 29, 31, 32, 33, 34\}$

   (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

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

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

$$6251007 \equiv 7 \pmod{1000}$$

Since $\gcd(7, 1000) = 1$, we have

$$7^{\phi(1000)} \equiv 1 \pmod{1000}$$

where

$$\Phi(1000) = 5^3 \times 2^3 \times (1 - \frac{1}{5})(1 - \frac{1}{2}) = 400$$

$$1960404 \equiv 4 \pmod{400}$$

$$6251007^{1960404} \equiv 7^4 \pmod{1000} \equiv 2401 \pmod{1000} = 401$$

(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 $((r||m) == \sigma^e \mod N)$.
Solution.
$\sigma^e \equiv 196^{43} \pmod{1165}$. Since 1165 is divisible by 5, then if we had $196^{43} \equiv 24 \pmod{1165}$, then we should have had $196^{43} \equiv 24 \equiv 4 \pmod 5$. But $196 \equiv 1 \pmod 5$ and so $196^{43} \equiv 1^{43} = 1 \pmod 5$. So, this signature is not valid.

(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.
Solution.
Suppose $N = pq$, then $\varphi(N) = (p-1)(q-1) = pq - p - q + 1 = N - (p+q) + 1$, so $pq = N$ and $p + q = N - \varphi(N) + 1$. This means that we know both the multiplication and summation of p and q, so p and q are roots of equation $(x - p)(x - q) = x^2 - (p + q) + pq = x^2 - (N - \varphi(N) + 1)x + N$.

(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 $(m_1 \times m_2)^5$.

Solution.
Suppose $m_1^e \equiv c_1 \pmod N, m_2^e \equiv c_2 \pmod N$, then $(m_1^5)^e \equiv c_1^5 \pmod N, (m_2^5)^e \equiv c_2^5 \pmod N, (m_1^5 m_2^5)^e \pmod N = c_1^5 c_2^5 = (c_1 c_2)^5 \pmod N$.

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 new line characters. Your program must output the cube roots of the five numbers, represented as binary and separated by new line characters, to a text file named as yourLastName-yourFirstName.txt. Make sure that you follow the naming convention. You can use Java, Python, C, or SageMath. Turn in your code and the output file via Blackboard.

Collaborators :