COMP5631: Cryptography and Security
2025 Spring – Written Assignment Number 1
Handed out: on Feb. 4, 2025
Due: on Feb. 18 by 23:30

Please upload your solution paper into Canvas by 23:30 on Feb. 18. If you have difficulty to solve some of the problems, you can search the Internet and get some textbooks on discrete mathematics and elementary number theory. Solving these problems are essential for this course. My lecture slides at https://www.cse.ust.hk/faculty/cding/COMP2711H/ may be helpful. Most of you learnt these topics during your undergraduate study, but you may have forgotten them. The objective of this assignment is to ask you to review them as early as possible.

Q1. Solve the equation $1009 \otimes_{21111} x = 2$ to find the unique solution $x \in \mathbb{Z}_{21111}$. Please use the extended Euclidean algorithm to compute the multiplicative inverse of 1009 modulo 21111 first, and then solve the equation. You are asked to write down all the details of your computation step by step. 20 marks **Solution:** To solve $1009 \otimes_{21111} x = 2$, we can rewrite it as:

$$1009 \cdot x \equiv 2 \pmod{21111} \tag{1}$$

Let's solve this step by step:

1. **Extended Euclidean Algorithm** to find the multiplicative inverse of 1009 modulo 21111:

$$\begin{aligned}
21111 &= 20 \times 1009 + 931 && \text{(divide)}\\
1009 &= 1 \times 931 + 78 && \text{(divide)}\\
931 &= 11 \times 78 + 73 && \text{(divide)}\\
78 &= 1 \times 73 + 5 && \text{(divide)}\\
73 &= 14 \times 5 + 3 && \text{(divide)}\\
5 &= 1 \times 3 + 2 && \text{(divide)}\\
3 &= 1 \times 2 + 1 && \text{(divide)}\\
2 &= 2 \times 1 + 0 && \text{(terminate)}
\end{aligned}$$

2. **Back-substitution** to express 1 as a linear combination:

$$\begin{aligned}
1 &= 3 - 1 \times 2\\
&= 3 - (5 - 3) = 2 \times 3 - 5\\
&= 2(73 - 14 \times 5) - 5 = 2 \times 73 - 29 \times 5\\
&= 2 \times 73 - 29(78 - 73) = 31 \times 73 - 29 \times 78\\
&= 31(931 - 11 \times 78) - 29 \times 78 = 31 \times 931 - 370 \times 78\\
&= 31 \times 931 - 370(1009 - 931) = 401 \times 931 - 370 \times 1009\\
&= 401(21111 - 20 \times 1009) - 370 \times 1009\\
&= 401 \times 21111 - 8390 \times 1009
\end{aligned}$$

3. Therefore, $1009^{-1} \equiv -8390 \equiv 12721 \pmod{21111}$
4. **Final solution:** Multiply both sides of equation 1 by 12721:

$$x \equiv 2 \times 12721 \equiv 25442 \equiv 4331 \pmod{21111}$$

5. **Verification:** We can verify that:

$$1009 \times 4331 \equiv 2 \pmod{21111}$$

Therefore, the unique solution is $x = 4331$ in $\mathbb{Z}_{21111}$.

Q2. This problem is about modular arithmetic.

1. Let a and b be two integers and $n \geq 2$ be an integer. Prove that the following equality holds: (10 marks)

$$(a + b) \bmod n = ((a \bmod n) + (b \bmod n)) \bmod n.$$

Proof:

Let's prove this step by step.

1. By the division algorithm, for any integers a and b, we can write:

$$a = q_1 n + r_1, \quad \text{where } 0 \leq r_1 < n \text{ and } r_1 = a \bmod n$$

$$b = q_2 n + r_2, \quad \text{where } 0 \leq r_2 < n \text{ and } r_2 = b \bmod n$$

2. Adding these equations:

$$\begin{aligned} a + b &= (q_1 n + r_1) + (q_2 n + r_2) \\ &= (q_1 + q_2)n + (r_1 + r_2) \\ &= kn + (r_1 + r_2), \text{ where } k = q_1 + q_2 \end{aligned}$$

3. Therefore:

$$(a + b) \bmod n = (r_1 + r_2) \bmod n$$

4. On the other hand:

$$((a \bmod n) + (b \bmod n)) \bmod n = (r_1 + r_2) \bmod n$$

5. Thus:

$$(a + b) \bmod n = ((a \bmod n) + (b \bmod n)) \bmod n$$

The equality is proved. □

2. Let a and b be two integers and $n \geq 2$ be an integer. Prove that the following equality holds: (10 marks)

$$(ab) \bmod n = ((a \bmod n)(b \bmod n)) \bmod n.$$

Proof:

Let's prove this step by step.

1. By the division algorithm, for any integers a and b, we can write:

$$a = q_1 n + r_1, \quad \text{where } 0 \leq r_1 < n \text{ and } r_1 = a \bmod n$$

$$b = q_2 n + r_2, \quad \text{where } 0 \leq r_2 < n \text{ and } r_2 = b \bmod n$$

2. Multiplying these equations:

$$\begin{aligned} a \cdot b &= (q_1 n + r_1)(q_2 n + r_2) \\ &= q_1 q_2 n^2 + (q_1 r_2 + q_2 r_1) n + r_1 r_2 \\ &= kn + r_1 r_2, \text{ where } k = q_1 q_2 n + (q_1 r_2 + q_2 r_1) \end{aligned}$$

3. Therefore:

$$(ab) \bmod n = (r_1 r_2) \bmod n$$

4. On the other hand:

$$((a \bmod n)(b \bmod n)) \bmod n = (r_1 r_2) \bmod n$$

5. Thus:

$$(ab) \bmod n = ((a \bmod n)(b \bmod n)) \bmod n$$

The equality is proved. □

Q3. For each positive integer n, let $\phi(n)$ be the total number of integers i with $1 \leq i \leq n-1$ and $\gcd(i, n) = 1$. This function $\phi(n)$ is called the *Euler totient function.* Prove that

$$\phi(pq) = (p-1)(q-1)$$

for a pair of distinct primes p and q. 20 marks

Proof: Let's prove that for distinct primes p and q, $\phi(pq) = (p-1)(q-1)$.

1. **Preliminary Understanding**: For a number $n = pq$ where p and q are distinct primes, $\phi(n)$ counts the positive integers less than n that are coprime to n.

2. **Analysis of Numbers from 1 to pq**: Let's analyze the structure of numbers from 1 to pq:

- Total numbers: pq
- Numbers divisible by p: q numbers $(p, 2p, ..., qp)$
- Numbers divisible by q: p numbers $(q, 2q, ..., pq)$
- Numbers divisible by both p and q: 1 number (pq)

3. **Using the Inclusion-Exclusion Principle**: To find $\phi(pq)$, we:

$$\begin{aligned} \phi(pq) &= \text{Total numbers } < pq - \text{ Non-coprime numbers} \\ &= (pq-1) - [(q+p-1) - 1] \\ &= pq - 1 - (p+q-2) \\ &= pq - p - q + 1 \\ &= (p-1)(q-1) \end{aligned}$$

4. **Verification**: This result makes intuitive sense because:

- The formula is symmetric in p and q
- When either p or q increases, $\phi(pq)$ increases
- The result is always positive for primes $p, q > 1$

Therefore, we have proved that $\phi(pq) = (p-1)(q-1)$ for distinct primes p and q. □

Q4. Euler's Theorem: For any positive integer a and n with $\gcd(a, n) = 1$, we have

$$a^{\phi(n)} \bmod n = 1.$$

If $n = p$ is prime, we have **Fermat's Theorem**:

$$a^{p-1} \bmod p = 1.$$

Prove Euler's theorem above in detail. (20 marks)

Proof: Let's prove this theorem step by step.

1. **Setup and Notations**: Let $r_1, r_2, \ldots, r_{\phi(n)}$ be the complete set of residues coprime to n. By definition, for each r_i:

- $1 \leq r_i \leq n-1$
- $\gcd(r_i, n) = 1$

2. **Key Observation**: Consider the set $S = \{ar_1, ar_2, \ldots, ar_{\phi(n)}\}$ where each element is taken modulo n. Since $\gcd(a, n) = 1$, we can prove that:

- Each ar_i is still coprime to n
- The elements in S are all distinct modulo n

3. **Main Argument**: Therefore, S must be a permutation of the original residue set modulo n. This means:

$$\{ar_1, ar_2, \ldots, ar_{\phi(n)}\} \equiv \{r_1, r_2, \ldots, r_{\phi(n)}\} \pmod{n}$$

4. **Product Comparison**: Taking the product of both sets:

$$(ar_1)(ar_2)\cdots(ar_{\phi(n)}) \equiv r_1 r_2 \cdots r_{\phi(n)} \pmod{n}$$

This can be rewritten as:

$$a^{\phi(n)}(r_1 r_2 \cdots r_{\phi(n)}) \equiv r_1 r_2 \cdots r_{\phi(n)} \pmod{n}$$

5. **Final Step**: Since $\gcd(r_1 r_2 \cdots r_{\phi(n)}, n) = 1$, we can cancel this product from both sides to obtain:

$$a^{\phi(n)} \equiv 1 \pmod{n}$$

Therefore, we have proved Euler's Theorem. When $n = p$ is prime, $\phi(p) = p - 1$, which gives us Fermat's Theorem as a special case. □

Q5. Let p be a prime. A positive integer α is called a *primitive root* of p if ever integer a with $1 \leq a \leq p-1$ can be expressed as

$$a = \alpha^i \bmod p$$

for a unique i with $0 \leq i \leq p-2$. It is known that every prime has at least one primitive root.

The exponent i is referred to as the **discrete logarithm**, or **index**, of a for the base α, and is denoted by $\log_\alpha(a)$ or index(a). The *discrete logarithm problem* is to compute the unique exponent i (i.e., $\log_\alpha(a)$), given p, α and a. If p is large (say, p has 130 digits), people believe that it is computationally very hard to solve the discrete logarithm problem.

Prove that 2 is a primitive root of 13. Find out $\log_2(10)$. (10 marks)

Show that it is easy to compute a, given p, α and i. To this end, you need to describe an efficient algorithm for computing a. (10 marks)

Part 1: Proving 2 is a Primitive Root of 13 and Finding $\log_2(10)$

To prove that 2 is a primitive root of 13, we must demonstrate that the powers of 2 modulo 13 generate all integers from 1 to 12. Let's compute successive powers:

$$\begin{aligned}
2^1 &\equiv 2 \pmod{13} \\
2^2 &\equiv 4 \pmod{13} \\
2^3 &\equiv 8 \pmod{13} \\
2^4 &\equiv 3 \pmod{13} \\
2^5 &\equiv 6 \pmod{13} \\
2^6 &\equiv 12 \pmod{13} \\
2^7 &\equiv 11 \pmod{13} \\
2^8 &\equiv 9 \pmod{13} \\
2^9 &\equiv 5 \pmod{13} \\
2^{10} &\equiv 10 \pmod{13} \\
2^{11} &\equiv 7 \pmod{13} \\
2^{12} &\equiv 1 \pmod{13}
\end{aligned}$$

As these powers generate all integers from 1 to 12 exactly once, 2 is indeed a primitive root of 13.

From these calculations, we can directly observe that $\log_2(10) = 10$ since $2^{10} \equiv 10 \pmod{13}$.

Part 2: Algorithm for Computing $a \equiv \alpha^i \pmod{p}$

To efficiently compute $a \equiv \alpha^i \pmod{p}$, we employ the **Square-and-Multiply Algorithm**:

Algorithm 1 Square-and-Multiply

```
Input: α, i, p
Output: α^i mod p
result ← 1
base ← α mod p
while i > 0 do
    if i is odd then
        result ← (result × base) mod p
    end if
    base ← (base × base) mod p
    i ← ⌊i/2⌋
end while
return result
```

Complexity Analysis:

- Time complexity: $O(\log i)$ operations
- Space complexity: $O(1)$ additional space

This algorithm is significantly more efficient than naive repeated multiplication, which would require $O(i)$ operations.

Example: Computing $2^{10} \bmod 13$

$$\begin{aligned} i = 10 &= (1010)_2 \\ 2^1 &\equiv 2 \pmod{13} \\ 2^2 &\equiv 4 \pmod{13} \\ 2^4 &\equiv 3 \pmod{13} \\ 2^8 &\equiv 9 \pmod{13} \\ 2^{10} &\equiv 10 \pmod{13} \end{aligned}$$

Summary:

- 2 is a primitive root of 13
- $\log_2(10) = 10$ in $\mathbb{Z}_{13}$
- The Square-and-Multiply algorithm provides an efficient $O(\log i)$ solution for modular exponentiation