Please answer each of the below questions. Remember, you may work together, but everyone MUST write up and understand their own solutions. Solutions that appear copied will be considered a violation of the academic honesty code. Also, you must list all people you work with.

1. **Modular Arithmetic:** Please answer the following questions by hand, show your work
   
   (a) Compute \([-8 \mod 16]\)
   
   (b) Compute \([1234567890 + 9876543210 \mod 5]\)
   
   (c) Is the following statement true: if \(2x = 2y \mod 6\), then \(x = y \mod 6\)? Explain why, or give a counterexample.
   
   (d) Is the following statement true: if \(x + 7 = y + 7 \mod 63\), then \(x = y \mod 63\)? Explain why, or give a counterexample.
   
   (e) Compute \([5^{-1} \mod 8]\).
   
   (f) Compute \([999^{-1} \mod 1000]\).

2. **Shamir Sharing:**
   
   Consider an \((n = 5, t = 3)\)-Shamir secret-sharing scheme over \(\mathbb{Z}_{17}\). Suppose \(P_1\) receives share \((x = 1, y = 2)\), \(P_2\) receives share \((2, 1)\), and \(P_3\) receives share \((3, 14)\). Calculate the corresponding Lagrange interpolating polynomial and calculate the secret. (Show your work.)

3. **Additive Sharing:**
   
   Consider a \((2, 2)\)-additive secret-sharing scheme over \(\mathbb{Z}_5\). Suppose that that the last bit of each party’s share leaks to the adversary (i.e., the adversary learns the least significant bit of \(s_1\) and \(s_2\), the shares held by \(P_1\) and \(P_2\)). Explain what the adversary knows about the secret \(s\).

4. **WRK18:**
   
   In the maliciously secure version of GRW18 [1] as I described in class, it is critical that the parties perform the cross-check on every wire. That is, they need to check that for every circuit wire \(w\), \(m^{(1)}_w + \lambda^{(2)}_w = m^{(2)}_w + \lambda^{(1)}_w\) where \(\lambda^{(i)}_w\) is a share of the wire mask from execution \(i\) of the semi-honest protocol, and \(m^{(i)}_w\) is the wire mask from the \(i\)th execution (recall that we run the semi-honest protocol twice with different parties).

   Now, suppose that to save communication, the parties try to batch their cross check. Specifically, they compute \(\Lambda^{(1)} = \sum_{w \in C} \lambda^{(1)}_w\) and \(M^{(1)} = \sum_{w \in C} m^{(1)}_w\) (\(\Lambda^{(2)}\) and \(M^{(2)}\) are defined similarly for the second execution). Then, the parties do a single batched cross-check to check that \(\Lambda^{(1)} + M^{(2)} = \Lambda^{(2)} + M^{(1)}\). Describe an attack that a malicious adversary corrupting one of the parties can do on this modified protocol.
References