Solution Approach

1. **Recover RSA Modulus (n):**
   * Send quadratic residues (4, 9, 16, etc.) to the oracle
   * Collect multiple square roots for each residue
   * Compute GCD of differences between squared roots to recover n
2. **Factor Modulus (n):**
   * Collect distinct roots for a residue
   * Compute GCD of root differences to find prime factor p
   * Derive q = n // p
3. **Compute RSA Private Key:**
   * Calculate φ(n) = (p-1)(q-1)
   * Compute private exponent d = e⁻¹ mod φ(n) (e=65537)
4. **Sign Challenge Message:**
   * Break oracle loop to receive random challenge m
   * Compute signature s = mᵈ mod n
   * Send signature to get flag

Key Insights

* Square root responses leak information about the modulus
* Root differences reveal factors through GCD calculations
* Small quadratic residues ensure manageable computations
* Proper prompt handling is crucial for server communication

Explanation

1. **Modulus Recovery:**
   * Sends quadratic residues to oracle
   * Collects square roots and computes differences
   * GCD of differences reveals modulus n
2. **Factorization:**
   * Finds distinct roots for a residue
   * GCD of root differences reveals prime factor p
   * Computes q = n // p
3. **Private Key Calculation:**
   * Computes φ(n) from p and q
   * Derives private exponent d using modular inverse
4. **Signature Generation:**
   * Breaks oracle loop to receive challenge m
   * Computes RSA signature s = mᵈ mod n
   * Sends signature to retrieve flag