6.875 Spring 2017	**Problem Set 4**	Problem 5
Skanda Koppula		
Collaborators: Jonathan Frankle, Sam DeLaughter		April 22, 2017

In the given protocol, the verifier (V) may choose one of three actions after recieving the commitments:

1. **Case 1**: Open the commitments for $s_{\pi(i)}$, $r_{\pi(i)}$, $r'_{\pi(i)}$
2. **Case 2**: Open the commitments for $r_{\pi(i)}$, $b_{\pi(i)}$, σ, σ'
3. **Case 3**: Open the commitments for $r'_{\pi(i)}$, $b_{\pi(i)}$, σ, σ'

Problem 5.1 In all three cases, V needs to check the integrity of all recieved commitments (verifying that the revealed values follow from the original secure commitment). In addition, V performs case-specific checks:

1. **Case 1**
   (a) Check that the set of $s_{\pi(i)}$ are the original members of S
   (b) Check that $r_{\pi(i)} + r'_{\pi(i)} = s_{\pi(i)} \forall 1 \leq i \leq n$
2. **Case 2**
   (a) Check that $\sigma + \sigma' = T \mod 2T - 1$
   (b) Check that $\sum_{i=1}^{n} r_{\pi(i)} b_{\pi(i)} = \sigma$
3. **Case 3**:
   (a) Check that $\sigma + \sigma' = T \mod 2T - 1$
   (b) Check that $\sum_{i=1}^{n} r'_{\pi(i)} b_{\pi(i)} = \sigma'$

Problem 5.2 We show that the protocol satisfies completeness, soundness, and ZK-ness:

1. **Completeness** If a set S is indeed in the language, and if the prover holds a valid witness S_1, S_2, then each of the checks above must be true, as per the definitions given in the protocol description (e.g. $\sigma + \sigma'$ represent two parts of the sum that constitutes $\sum S_1$, and by definition, $\sum S_1 = T$). Thus, V always accepts when S is in the language with a protocol-adhering prover.
2. **Soundness** There are three possible actions, and in each case, we examine the steps of malicious prover may take to convince us that S is in the language:
   (a) If the prover guessed correctly that V would choose case 1, it could forge verifiable $r_{\pi(i)}, r'_{\pi(i)}, s_{\pi(i)}$ and not worry about verifiable values for other commitments. A malicious prover would have had the latitude to create nonsense σ, σ' and partition values b_i. However, if the malicious prover doesn't guess this correctly, then it must output non-checkable $r_{\pi(i)}, r'_{\pi(i)}, s_{\pi(i)}$ (since $S \notin \mathsf{EQ\text{-}PART}$), and V's checks would fail.

(b) If the prover guessed correctly that V would choose case 2, it could forge verifiable $r_{\pi(i)}, b_{\pi(i)}, \sigma, \sigma'$ and not worry about verifiable values for other commitments. The prover would have the latitude to create nonsense $r'_{\pi(i)}, s_{\pi(i)}$. However, if the malicious prover doesn't guess this correctly, then it must output non-checkable $r_{\pi(i)}, b_{\pi(i)}, \sigma, \sigma'$, and V's checks would fail.

(c) The argument is the same as the previous case, replacing $r \leftrightarrow r'$

Attempting to match V, the prover guesses at random V's selection (and forges a response that passes verification). This means soundness, the probability V guesses a case that the prover chose not to forge is $\frac{2}{3}$.

3. **Zero-Knowledge** We create a simulator to produce a valid protocol transcript. The simulator randomly picks V's chosen case, and proceeds accordingly:

   (a) **Case 1**:

      i. Sample a random permutation $\pi(i)$. Sample $r_1, \ldots, r_n \xleftarrow{\$} \mathbb{Z}_{2T+1}$.
      ii. Compute $r'_{\pi(i)} = s_{\pi(i)} - r_{\pi(i)} \forall 1 \leq i \leq n$.
      iii. Create commitments to $s_{\pi(i)}, r_{\pi(i)}, r'_{\pi(i)}$, and commitments to randomly chosen $b_{\pi(i)}, \sigma, \sigma'$.
      iv. Send this to V, record the message on the transcript.
      v. Replay V until it chooses Case 1. Record this on the transcript.
      vi. Reveal the commitments to V. Record this on the transcript. V will accept, by construction.

   (b) **Case 2**:

      i. Sample $\sigma \xleftarrow{\$} \mathbb{Z}_{2T+1}$; $b_1 \ldots b_n \xleftarrow{\$} \{0,1\}^n$; and $r_1, \ldots, r_{n-1} \xleftarrow{\$} \mathbb{Z}_{2T+1}$.
      ii. Compute $\sigma' = T - \sigma \mod 2T+1$ and r_n such that $\sum_{i=1}^{n} r_i b_i = \sigma$
      iii. Create commitments to $r_i, b_i, \sigma, \sigma'$, and commitments to randomly chosen s_i, r'_i.
      iv. Send this to V, record the message on the transcript.
      v. Replay V until it chooses Case 2. Record this on the transcript.
      vi. Reveal the commitments to V. Record this on the transcript. V will accept, by construction.

   (c) **Case 3**: The transcript construction is exactly analagous to Case 2, replacing $r' \leftrightarrow r, \sigma \leftrightarrow \sigma'$, and Case 2 $\leftrightarrow$ Case 3.

   Simulated transcripts are indistinguishable from real transcripts, and so the protocol is ZK.

Problem 5.3 We construct an extractor that extracts the witness (S_1, S_2) from a prover P:

1. Run P until the point t at which it has selected all $\pi(i), r_i, r'_i$, and b_i, and sent commitments.

2. Request from P reveals for Case 1 commitments. This means we have plaintext values for all $s_{\pi(i)}, r_{\pi(i)}$, and $r'_{\pi(i)}$

3. Rewind P back to t until and request from P commitment reveals for Case 2 commitments. This means we have plaintext values for all $b_{\pi(i)}, \sigma$, and σ'.

4. From this, we can reconstruct a witness. With knowledge of $s_{\pi(i)}$ and $b_{\pi(i)}$, we can correspond $s_{\pi(i)}$ to members of S (with equal-value elements being interchangeable), and read from $b_{\pi(i)}$ values the partition of $s_{\pi(i)}$'s into the two sets S_1, S_2.