Homework 9: Multiparty secure computation

Total of 115 points

1. (25 points) Let F be the two party functionality such that $F(H\|C, H')$ outputs $(1,1)$ if the graph H equals the graph H' and C is a Hamiltonian cycle and otherwise outputs $(0,0)$. Prove that a protocol for computing F is a zero knowledge proof (w.r.t. an *efficient* prover) system for the language of Hamiltonicity.

2. (25 points) Let F be the k-party functionality that on inputs $x_1, \ldots, x_k \in \{0,1\}$ outputs to all parties the majority value of the x_i's. Prove that in any protocol that securely computes F, for any adversary that controls less than half of the parties, if at least $k/2+1$ of the other parties' inputs equal 0, then the adversary will not be able to cause an honest party to output 1.

3. (25 points) For two distributions X, Y over some set Ω, we define their *total variation distance*, denoted as $\Delta(X,Y)$ as $\sum_{\omega\in\Omega} |\Pr[X=\omega] - \Pr[Y=\omega]|$. If X is a distribution over Ω then we denote by X^m the distribution over Ω^m where every entry of X^m is sampled independently from X (i.e., $\Pr[X^m = (\omega_1, \ldots, \omega_m)] = \Pr[X=\omega_1]\cdots\Pr[X=\omega_m]$). Prove that if two distributions X and Y satisfy $\Delta(X,T) < \delta$ then $\Delta(X^m, Y^m) \leq m\delta$.

4. (40 points) For a prime $p > 5$, suppose that we select a random degree 2 polynomial $S(x) = s_0 + s_1 x + s_2 x^2$ moudlo p by selecting s_0, s_1, s_2 independently and uniformly from $\mathbb{Z}_p$, and consider the random variable $(S(1), S(2), S(3), S(4), S(5)) \in \mathbb{Z}_p^5$.

   a. (10 points) Prove that for every distinct $i, j, k \in \{1, \ldots, 5\}$, there is an algorithm to recover $S(0)$ from $S(i), S(j), S(k)$.

   b. (10 points) Prove that for every $i, j \in \{1, \ldots, 5\}$, the distribution of $S(0), S(i), S(j)$ is the uniform distribution over $\mathbb{Z}_p^3$. Conclude that there is no algorithm to recover $S(0)$ from $S(i)$ and $S(j)$.

   c. (20 points) The "pretty good privacy (PGP)" software used to have (essentially) the following mechanism for key recovery. To hide a key $K \in \{0,1\}^n$, we pick a prime $p > 2^n$ (and so can think of K as a member of $\mathbb{Z}_p$). The user would record 5 question answer pairs $(q_1, a_1), \ldots, (q_5, a_5)$ (each encoded as a string). We let H be a hash function that maps $\{0,1\}^*$ to $\mathbb{Z}_p$ and model it as a random oracle. Then we pick a random salt $salt \in \{0,1\}^n$, random degree 2 polynomial S as above subject to $S(0) = K$ and store the data block $D = (q_1, \ldots, q_5, salt, z_1, \ldots, z_5)$ where $z_i = H(a_i \| salt) + S(i) (\bmod q)$ on the user's machine.

i. (10 points) Prove that given this information, a user who remembers at least three of the answers to the questions can recover the key K.

ii. (10 points) Prove that in the random oracle model, one can transform a time T adversary A that succeeds in recovering a random key K from D with probability at least 1/2 into an adversary A' that outputs three of the answers in $\{a_1, \ldots, a_5\}$ with probability at least 1/4.