Problem 1: (ID protocols) Recall that in Schnorr’s ID protocol in a group \( G \) of order \( q \) the prover first chooses a random \( r \overset{R}{\leftarrow} \{1, \ldots, q\} \) and sends \( g^r \) to the verifier. To improve performance, suppose that the prover chooses \( r \overset{R}{\leftarrow} \{1, \ldots, t\} \) for some large \( t \) much smaller than \( q \) (say, \( q = 2^{256} \) but \( t = 2^{128} \)). Show that the resulting protocol is not honest verifier zero knowledge (HVZK). In particular, show that when \( t < q^{1/2} \), an honest verifier can recover the secret key after about two executions of the ID protocol.

Problem 2: (Key Exchange) Recall the EEBKE protocol discussed in class: in the first flow \( P \) generates a \((pk, sk)\) pair for a public-key encryption scheme. \( P \) sends \( pk \) to \( Q \) and receives back an encryption of a random session key \( k \). \( P \) uses \( sk \) to recover the session key and sends a signature back to \( Q \). The protocol works as follows:

\[
P P \quad Q Q
\]

\[
\begin{array}{c}
pk, \; Cert_P \\
\hline \hline 
\quad c := E(pk, k), \; \sigma_1 := \text{Sig}_Q(pk, c, id_P), \; Cert_Q \\
\quad \sigma_2 := \text{Sig}_P(c, id_Q) \\
\end{array}
\]

a. Suppose \( Q \) does not sign \( c \) in \( \sigma_1 \). Describe an attack on the protocol.

b. Support \( Q \) does not sign \( pk \) in \( \sigma_1 \). Describe an attack on the protocol.

c. Suppose \( Q \) does not sign \( id_P \) in \( \sigma_1 \). Describe an identity-misbinding attack on the protocol.

d. Suppose \( P \) does not sign \( c \) in \( \sigma_2 \). Describe an attack on the protocol.

Problem 3: (PAKE) Recall the PAKE protocol discussed in class (a.k.a SPAKE). Suppose we take \( U = V \) in the public parameters.

a. Explain where the proof of security given in class fails.

b. Show that the protocol is secure if instead of using the CDH assumption we make a stronger assumption, namely that given \((g,g^x,g^y,g^z)\) it is difficult to compute \( g^{xy} \). It suffices to explain how this stronger assumption bypasses the stumbling block you identified in part (a).

The SPAKE protocol and its proof are described at:

http://www.di.ens.fr/~mabdalla/papers/AbPo05a-letter.pdf
Problem 4: (two party protocols) Let $p$ be a prime. Suppose user $A$ has an $x \in \mathbb{Z}_p$ and user $B$ has a $y \in \mathbb{Z}_p$. They wish to compute the following function: $f(x, y) = 0$ when $x = y$ and $f(x, y) = 1$ when $x \neq y$, without revealing any other information about $x$ or $y$. Your goal is to give an efficient solution to this problem in the honest-but-curious settings.

a. Estimate the amount of communication needed for this problem using Yao’s garbled circuits method. State your estimate asymptotically as a function of $\log_2 p$. You may assume that we use the Naor-Pinkas OT in (a subgroup) of $\mathbb{Z}_p^*$. 

b. Suppose there is a third party who is willing to help. Give an efficient 3-party protocol for computing $f(x, y)$ so that nothing else is revealed to any single party (1-private). Prove 1-privacy by showing a simulator for each party’s view of the protocol (the simulator is given $f(x, y)$ and that party’s input).

c. Extra credit: can you suggest 1-private 2-party protocol that is more efficient than Yao’s garbled circuit method? Feel free to consult the web.