1. Design and analyze an identification scheme that satisfies the following properties:
   - it should be secure against (unrestricted) cheating verifier attacks, assuming the computation of cube roots modulo a composite \( n \) is hard (here, we presume that \( n \) is chosen from a distribution such that \( 3 \nmid \phi(n) \));
   - it should consist of 3 flows, each of which is a message whose bit-length \( O(\text{len}(n)) \).

2. In class, we have only defined security for identification schemes with respect to a single prover (and corresponding public key/private key pair). Develop a reasonable formal definition of security against (unrestricted) cheating verifier attacks for an \( n \)-prover system, and prove that security for a single prover implies security for \( n \) provers.

3. In class, we discussed an interactive proof system for proving that two discrete logarithms are equal. This problem examines a proof system for proving that two discrete logarithms are unequal. The common input to the prover \( P \) and verifier \( V \) consists of \( p, q, \gamma, \gamma', \alpha, \alpha' \), where \( p \) and \( q \) are primes with \( q \mid (p-1) \), \( \gamma, \gamma' \) are generators for the subgroup \( G \) of order \( q \) in \( \mathbb{Z}_p^* \), and \( \alpha, \alpha' \in G \) with \( \log_\gamma \alpha \neq \log_\gamma \alpha' \). The prover has an additional input, namely \( x \in \mathbb{Z}_q \) with \( \alpha = \gamma^x \). The protocol runs as follows:
   
   (a) \( P \) chooses \( r \in \mathbb{Z}_q^*, \hat{r} \in \mathbb{Z}_q, \hat{s} \in \mathbb{Z}_q \) at random, and computes
   
   \[
   s \leftarrow xr, \quad \kappa \leftarrow (\gamma')^s/(\alpha')^r, \quad \mu' \leftarrow (\gamma')^{\hat{s}}/(\alpha')^{\hat{r}}, \quad \mu \leftarrow \gamma^x/\alpha^\hat{r},
   \]

   and then sends \((\kappa, \mu', \mu)\) to \( V \).
   
   (b) Upon receiving \((\kappa, \mu', \mu)\) \( \in G(3) \), \( V \) chooses \( c \in \mathbb{Z}_q \) at random, and sends \( c \) to \( P \).
   
   (c) Upon receiving \( c \in \mathbb{Z}_q \), \( P \) computes
   
   \[
   \hat{s} \leftarrow \hat{s} + cs \in \mathbb{Z}_q, \quad \hat{r} \leftarrow \hat{r} + cr \in \mathbb{Z}_q,
   \]

   and sends \((\hat{s}, \hat{r})\) to \( V \).
   
   (d) Upon receiving \((\hat{s}, \hat{r}) \in \mathbb{Z}_q^2 \), \( V \) checks that
   
   \[
   \kappa \neq 1, \quad (\gamma')^{\hat{s}}/(\alpha')^{\hat{r}} = \mu'\kappa^c \quad \text{and} \quad \gamma^x/\alpha^{\hat{r}} = \mu.
   \]

   If these checks pass, then \( V \) accepts, otherwise \( V \) rejects.

Show that the above protocol is an honest-verifier zero-knowledge proof of inequality of discrete logarithms. That is, show that (1) if \( \log_\gamma \alpha \neq \log_\gamma \alpha' \), then an honest prover always makes an honest verifier accept, (2) if \( \log_\gamma \alpha = \log_\gamma \alpha' \), then any prover makes an honest verifier accept with only negligible probability, and (3) a conversation between a honest prover and an honest verifier can be simulated by an algorithm that is given only the common input of the prover and verifier (and assuming that \( \log_\gamma \alpha \neq \log_\gamma \alpha' \)).

4. Suppose we have a semantically secure public-key encryption scheme. The encryption algorithm is \( E \). For simplicity, assume the message space is \( \{0,1\} \). Also, assume the space of public keys is a finite abelian group \( G \) (written multiplicatively), where the key generation algorithm generates public keys that are uniformly distributed over \( G \) (ElGamal encryption is an example of such a scheme).

In addition, suppose we have a collision resistant hash function \( H : G \rightarrow \{0,1\}^\ell \)

Now consider the following game, played between a challenger and an adversary \( A \):
(a) The challenger generates system parameters for the encryption scheme and the hash function, and sends these to \( \mathcal{A} \).

(b) \( \mathcal{A} \) sends \( h \in \{0,1\}^\ell \) to the challenger.

(c) The challenger chooses \( k_2 \in G \) at random, and sends \( k_2 \) to \( \mathcal{A} \).

(d) \( \mathcal{A} \) sends \( k_1 \) to \( \mathcal{A} \).

(e) The challenger checks that \( H(k_1) = h \). If not, the challenger halts the game. Otherwise, the challenger chooses \( b \in \{0,1\} \) at random, computes

\[
k \leftarrow k_1 \cdot k_2 \in G, \quad c \leftarrow E(k, b),
\]

and sends the ciphertext \( c \) to \( \mathcal{A} \).

(f) \( \mathcal{A} \) outputs \( \hat{b} \in \{0,1\} \).

For completeness, if the protocol halts for any reason before step (f), let us define \( \hat{b} := 0 \).

We define \( \mathcal{A} \)'s advantage to be \( |\Pr[\hat{b} = b] - 1/2| \).

The above game is much like the usual game defining semantic security, except that now, instead of having the challenger generate a public key \( k \) at random, the public key is generated using an interactive protocol.

Prove the following: under the assumptions that the encryption scheme is semantically secure, and the hash function is collision resistant, every efficient adversary has only a negligible advantage in the above game.