Problem 1: Consider the following ElGamal-like encryption system in a group $G$ of prime order $q$: the public key is $g, h \in G$ and an encryption of message $m \in \{0, 1\}$ is $(g^r, h^rg^m)$ where $r$ chosen at random in $\mathbb{Z}_q$. Your goal is to devise an honest-verifier zero-knowledge proof for proving that an ElGamal ciphertext is an encryption of 0 or 1. That is, the proof system should recognize the language
\[
\{(g, h, g^r, h^r) \mid r \in \mathbb{Z}_q\} \cup \{(g, h, g^r, h^r g^m) \mid r \in \mathbb{Z}_q\} \subseteq \mathbb{G}^2.
\]
Remember to prove completeness, soundness, and zero-knowledge.

Hint: start from the Chaum-Pedersen protocol for proving equality of discrete-log. Generalize the protocol into an OR proof as we did in class. If you get stuck, this paper might help: www.win.tue.nl/~berry/papers/crypto94.pdf

Extra credit: Design an efficient zero-knowledge proof that a 4-tuple is not a Diffie-Hellman tuple. That is, the protocol should recognize the language $\{(g, h, g^r, h^s) : r \neq s\}$.

Problem 2: In this problem we consider a candidate construction for Identity Based Encryption based on the discrete-log problem in a group $G$ of prime order $q$ with generator $g$. The group $G$ need not have a pairing.

The setup algorithm generates a random $a, b, c \in \mathbb{Z}_q$ and outputs the public parameters $\text{pp} = (g, g_1 := g^a, g_2 := g^b, g_3 := g^c)$ and master key $\text{mk} = (a, b, c)$. Let $H : \{0, 1\}^* \rightarrow \mathbb{Z}_q$ be a hash function and define the secret key for identity $id$ as $\text{sk}_id := (\text{pp}, \alpha, \beta)$ where $\alpha, \beta \in \mathbb{Z}_q$ is a random pair satisfying $(a + H(id))\alpha + b\beta = c$ in $\mathbb{Z}_q$. To encrypt a message $m \in G$ to identity $id$ the encryption algorithm chooses a random $r \in \mathbb{Z}_q$ and outputs the ciphertext $\text{ct} := (g^{H(id)}g_1^r, g_2^r, m \cdot g_3^r)$.

a. Explain how the key generation algorithm, $\text{KeyGen}$(mk, id), and decryption algorithm, $\text{Dec}(\text{sk}_id, \text{ct})$, work.

b. Show that if an attacker obtains the secret keys of any three identities $id_1, id_2, id_3$ (where $H(id_1), H(id_2), H(id_3)$ are distinct) he can completely break the system. That is, he can decrypt all ciphertexts, even those not intended for identities $id_1, id_2, id_3$. 


**Problem 3:** Aggregate signatures. Let $\mathbb{G}$ be a pairing group of order $q$ where $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$ denotes the pairing in $\mathbb{G}$. Let $g \in \mathbb{G}$ be a generator. In class we defined the BLS signature scheme: the public key is $\mathcal{vk} = g^\alpha$ and a signature on a message $m \in \{0, 1\}^*$ is defined as $\sigma := H(\mathcal{vk}, m)^\alpha$ where $H : \mathbb{G} \times \{0, 1\}^* \rightarrow \mathbb{G}$ is a hash function.

Suppose we have $n$ public keys $\mathcal{vk}_1 = g^\alpha_1, \ldots, \mathcal{vk}_n = g^\alpha_n$ and $n$ messages $m_1, \ldots, m_n \in \{0, 1\}^*$. We are given $n$ signatures $\sigma_i := H(\mathcal{vk}_i, m_i)^{\alpha_i}$ for $i = 1, \ldots, n$. We wish to aggregate all the signatures $\sigma_1, \ldots, \sigma_n$ into a single signature $\sigma$ that will serve as a signature validating the fact that user $i$ signed $m_i$ for all $i = 1, \ldots, n$.

Let us define $\sigma := \prod_{i=1}^n \sigma_i$. This $\sigma$ is called an aggregate signature. Show how a verifier, given $(\mathcal{vk}_1, m_1), \ldots, (\mathcal{vk}_n, m_n)$ and $\sigma$, can verify that indeed user $i$ signed $m_i$ for all $i = 1, \ldots, n$.

Note: this construction can be used to compress all the signatures in a certificate chain into a single signature. The construction can be proven secure under standard assumptions in bilinear groups.