Question 1
Consider the following encryption scheme for a one-bit message $M \in \{0, 1\}$. Let $g$ be a quadratic residue modulo a safe prime $q$. The secret key is a random $X \in \mathbb{Z}_q^*$ and the public key is $h = g^X$. To encrypt a 0 output $(g^R, h^R)$ for a random $R$ in $\mathbb{Z}_q^*$. To encrypt a 1 output $(g^{R'}, h^{R'})$ where $R$ and $R'$ are independent random elements in $\mathbb{Z}_q^*$.

(a) Show that it is not possible to decrypt ciphertexts with probability 1.

**Solution:** The ciphertext $(g, h)$ is a possible encryption of both 0 and 1. If decryption succeeds with probability 1 it must decrypt to both 0 and 1, which is impossible for a deterministic decryption algorithm.

(b) Describe and analyze a decryption algorithm that succeeds with probability $1 - \Omega(1/q)$.

**Solution:** Given ciphertext $(C, C')$, decrypt to 0 if $C^X = C'$ and to 1 otherwise. If $(C, C')$ is an encryption of zero the decryption is correct with probability one. Otherwise, a decryption error occurs only when $(g^R, g^{XR'})$ decrypts to zero, namely when $g^{XR} = g^{XR'}$. This happens with probability at most $2/(q - 1)$ even conditioned on $X$.

(c) Show that the encryption is message indistinguishable assuming the $(s, \varepsilon)$-DDH assumption in base $g$. Work out the parameters.

**Solution:** Encryptions are $(s, \varepsilon)$-message indistinguishable. Suppose $(PK, Enc(PK, 0)) = (g^X, g^R, g^{XR})$ and $(PK, Enc(PK, 1)) = (g^X, g^R, g^{XR'})$ can be distinguished with advantage $\varepsilon$ by some circuit $D$. As $R'$ is independent of $X$ and $R$, $g^{XR'}$ is a random group element that is independent of $g^X$ and $g^R$, so $(g^X, g^R, g^{XR'})$ is identically distributed to $(g^X, g^R, g^Y)$ for a random $Y \in \mathbb{Z}_q^*$. Therefore $D$ distinguishes DDH triples with advantage $\varepsilon$.

Question 2
In this question you will analyze the following LWE-based public-key identification protocol. The secret key is a random $x \sim \{-1, 1\}^m$. The public key is $(A, z = xA)$ where $A$ is a random $m \times n$ matrix over $\mathbb{Z}_q$. All arithmetic is modulo $q$.

1. Prover chooses a random $r \sim \{-b, \ldots, b\}^m$ and sends $h = rA$.
2. Verifier sends a random bit $c \sim \{0, 1\}$.
3. Prover sends $y = r + cx$.
4. Verifier accepts if $yA = h + cz$ and $y \in \{-b - 1, \ldots, b + 1\}^m$. 
(a) Show that if \( m = 1 \) then conditioned on \(|r + x| \leq b - 1\), \( r \) and \( r + x \) are identically distributed.

**Solution:** Both random variables are uniform over \( \{-b + 1, \ldots, b - 1\} \). Since \( r \) is uniform, any pair of outcomes is equally probable under the conditioning. For \( r + x \), any outcome \( y \) can arise from \( r = y + 1, x = -1 \) or from \( r = y - 1, x = 1 \) so any two outcomes are also equiprobable.

(b) Now let \( m \) be arbitrary as in the protocol. Show that \( r \) and \( r + x \) are \( O(m/b) \)-statistically close.

**Solution:** For both random variables in part (a), the probability that the condition is not satisfied equals \( 2/(2b + 1) \). Since the entries of \( r \) and \( r + x \) are independent, the probabilities that some coordinate falls outside the range \( \{-b + 1, \ldots, b + 1\} \) are equal for \( r \) and \( r + x \) and they are both upper bounded by \( 2m/(2b + 1) \leq m/b \). Conditioned on this not happening the two are identically distributed, so the advantage of any distinguisher is at most \( m/b \).

Alternatively, we can reduce the claim for general \( m \) to the claim for \( m = 1 \) using Lemma 5 from Lecture 3: Since for each coordinate \( i \), \( r_i \) and \( r_i + x_i \) are \((\infty, 1/b)\)-indistinguishable and the \( r_i \) are independent, \( r \) and \( x \) are \((\infty, m/b)\)-indistinguishable.

(c) Show that the view of an eavesdropper who sees \( q' \) protocol transcripts is \( O(q'n/b) \)-statistically close to some random variable that can be efficiently sampled by a simulator that is given only the public key.

**Solution:** Let’s do \( q' = 1 \) first. The view of the eavesdropper consists of the public key \((A, h = xA)\) and the transcript \((rA, c, r + cx)\). Given the public key the simulator samples \( y \) from \( \{-b, \ldots, b\}^m \), \( c \sim \{0, 1\} \), and outputs \((yA - ch, c, y)\). When \( c = 0 \) the two views are identically distributed. When \( c = 1 \) the two views are \((A, xA, rA, 1, r + x)\) and \((A, xA, (y - x)A, 1, y)\). If these two random variables were distinguishable, so would be \((x, r + x)\) and \((x, y)\). In part (b) we showed these two are \( m/b \)-statistically close. Since \( c \) is equally likely to take values zero and one, the eavesdropper’s view and the simulated view are \( m/2b \)-statistically close.

For general \( q' \) the simulator samples each transcript independently given the public key. By the same argument, if the two views can be distinguished with advantage \( \varepsilon \) so can \((x, r_1 + c_1 x, r_2 + c_2 x, \ldots, r_{q'} + c_{q'} x)\) and \((x, y_1, \ldots, y_{q'})\). By a hybrid argument \((x, r_i + c_i x)\) and \((x, y_i)\) can then be distinguished with advantage \( \varepsilon/q' \) for some \( i \). It follows that \( \varepsilon \leq q'm/2b \).

(d) Let \( h_A(x) = xA \), where the entries of \( x \) are of magnitude at most \( 2(b+1) \). Show that if \( h \) is a collision-resistant hash function then no efficient cheating prover can handle both challenges \( c = 0 \) and \( c = 1 \). Conclude that, if repeated sufficiently many times, the protocol is secure against eavesdropping. (Work out the dependences between the security parameters.)

**Solution:** Suppose a cheating prover can handle both \( c = 0 \) and \( c = 1 \) with responses \( y_0 \) and \( y_1 \), respectively. Then \( y_0, y_1 \in \{-b - 1, \ldots, b + 1\}^n \) both satisfy the equation \( y_c A = k + ch = k + cx A \).

Subtracting these two equations gives \((y_1 - y_0)A = x A\).

We can almost conclude that the cheating prover found a collision to \( h_A \), except that \( x \) and \( y_1 - y_0 \) might be equal. It turns out that this can happen at most a \( q'/2^m \)-fraction of the time. The reason is that for a typical choice of \( x \), \( h_A \) will have collisions \( x' \neq x \) such that \( h_A(x') = h_A(x) \). Conditioned on the public key, each of these colliding values is equally likely to be the secret key, so the cheating prover could not have found \( x \) unless it guessed it correctly among all of them. If \( h_A \) was regular (i.e.
each output arises from the same number of inputs) then the probability that the cheating prover “guessed” the actual secret key $x$ is at most $q^n/2^m$. If not, the probability can only be lower (it takes a bit more work to prove this.)

In conclusion, if the cheating prover succeeds with probability at least $(1 + \varepsilon)/2$, then it can handle both $c = 0$ and $c = 1$ with probability at least $\varepsilon$ and it can find a true collision to $h_A$ with probability at least $(1 - q^n/2^m)\varepsilon$. If say $\varepsilon = 1/2$ and $m = n\log q + 1$ and the prover is run $k$ times, then any prover that passes validation with probability $(3/4)^k$ can be used to find a collision in $h_A$ with probability $1/4$.

(e) (Optional) Prove that the protocol is secure against impersonation.

**Solution:** (Sketch) The simulator (for each round of impersonation) acts like the simulator in part (b), then runs the cheating verifier and outputs the transcript if his actual challenge $c^*$ equals the guessed challenge $c$. The message received by the cheating verifier in round 1 is no longer independent of $c$ because the distributions of the first messages are a bit different when $c = 0$ and when $c = 1$. Nonetheless it can be argued that the event $c^* = c$ still occurs with probability at least $1/2 - O(m/b)$. In conclusion, after $r$ simulation attempts the simulator produces a transcript that is $O(m/b)$-indistinguishable from the real one with probability at least $(1/2 - O(m/b))^r$.

**Question 3**

Assume $F_K : \{0, 1\}^{n+1} \rightarrow \{0, 1\}^n$ is an $(s, \varepsilon)$-pseudorandom function. Which of the following is a secure MAC tagging algorithm for message length $2n$? Justify your claim.

(a) $Tag(K, M_0M_1) = (F_K(M_0, 0), F_K(M_1, 1))$, $Ver(K, M_0M_1, T_0T_1)$ accepts iff $F_K(M_0, 0) = T_0$ and $F_K(M_1, 1) = T_1$.

**Solution:** Not secure. Query the tagging oracle on messages $M_0M_1$ and $M_0'M_1'$ with $M_0 \neq M_0'$ and $M_1 \neq M_1'$ to obtain tags $T_0T_1$ and $T_0'T_1$. Then $(M_0M_1, M_0T_1)$ is a forgery.

(b) $Tag(K, M_0M_1) = F_K(M_0, 0) + F_K(M_1, 1)$, $Ver(K, M_0M_1, T)$ accepts iff $F_K(M_0, 0) + F_K(M_1, 1) = T$.

**Solution:** Not secure. Let $M_0M_1$ and $M_0'M_1'$ be as in part (a). Query the tagging oracle on $M_0M_1$, $M_0'M_1'$ and $M_0'M_1$ to obtain tags $T_1, T_2, T_3$. Then $(M_0M_1, T_1 + T_2 + T_3)$ is a forgery.

**Question 4**

In this question you will show that using an obfuscator, an adversary can plant a collision in a hash function that makes it insecure against him, but secure against everyone else. Let $h : \{0, 1\}^m \rightarrow \{0, 1\}^n$ be a collision-resistant hash, $Obf$ an obfuscator, and $A$ the following algorithm:

1. Sample a random key $K$ and a random input $\hat{x} \sim \{0, 1\}^m \setminus \{0\}$.

2. Construct a circuit $h'$ that implements the function

   $$h'(x) = \begin{cases} h_K(0), & \text{if } x = \hat{x}, \\ h_K(x), & \text{if not}. \end{cases}$$
3. Output \( H = Obf(h') \).

Then \( A \) knows a collision for \( H \), namely the pair \((0, \hat{x})\). We can view \( H \) both as a random key and the function described by it, so \((s, \varepsilon)\)-collision-resistance means that the probability that \( C(H) \) outputs a collision for \( H \) is at most \( \varepsilon \) for every \( C \) of size at most \( s \).

(a) Show that the views \( D^{h_K} \) and \( D^{h'} \) are \( q/(2^m - 1) \)-statistically close for any distinguisher \( D \) that makes at most \( q \) queries to its oracle.

**Solution:** Even for fixed \( K \), \( h_K \) and \( h' \) differ on a single input random input \( \hat{x} \). Consider a hybrid \( h_i \) that answers its first \( i - 1 \) queries like \( h_K \) and the rest like \( h' \). Then \( D^{h_i} \) and \( D^{h_{i+1}} \) are \( 1/(2^m - 1) \)-statistically close because they can only be distinguished if \( D \) queries \( \hat{x} \) in round \( i \), which happens with probability \( 1/(2^m - 1) \). By the triangle inequality the views \( D^{h_K} \) and \( D^{h'} \) are \( q/(2^m - 1) \)-statistically close.

(b) Show that if \( h \) is \((s, \varepsilon)\)-collision resistant and \( Obf \) is \((s + 2t + O(n), \varepsilon')\)-VBB secure, \( H \) is \((s - tt', \varepsilon + \varepsilon' + q/(2^m - 1))\)-collision resistant, where \( t \) and \( t' \) are the sizes \( h \) and the VBB simulator, respectively.

**Solution:** Suppose some collision-finder \( C \) on input \( H \), finds a collision \( x, x' \) for \( H \) with probability \( \varepsilon^* \). By the VBB security of \( H \) there is a simulator \( D \) of size \( s \) such that the outputs of \( S^{h'} \) and \( C(H) \) are \((2t + O(m), \varepsilon')\)-indistinguishable. This means the probabilities that \( C(H) \) and \( D^{h'} \) output collisions can differ by at most \( \varepsilon' \), because a distinguisher that tests if \( x, x' \) is a collision has size \( 2t + O(m) \). In conclusion, the probability that \( D^{h'} \) outputs a collision for \( h' \) is at least \( \varepsilon^* - \varepsilon' \).

By part (a) the probability that \( D^{h_K} \) outputs a collision for \( h' \) is then at least \( \varepsilon^* - \varepsilon' - q/(2^m - 1) \). Since \( h_K \) is independent of \( \hat{x} \), this collision involves \( \hat{x} \) with probability at most \( 1/(2^m - 1) \). Otherwise, \( D^{h_K} \) must have output a collision for \( h_K \). This happens with probability \( \varepsilon^* - \varepsilon' - (q + 1)/(2^m - 1) \) (a little worse than advertised).

It remains to do a bit of technical work: An actual collision-finder takes the key \( K \) as an input. We want to use it to implement \( D^{h_K} \), namely provide oracle access to \( h_K \). As there are at most \( t' \) oracle calls, each implementable in size \( t \), this results in a circuit that is larger by \( tt' \), accounting for the deterioration in the size parameter.

(c) Show that the MAC from Theorem 5 in Lecture 6 is insecure against a forger that knows \( \hat{x} \).

**Solution:** A forger that knows \( \hat{x} \) can query the oracle on \( \hat{x} \) and obtain the tag \( T = Tag(H(\hat{x})) \). Then the message-tag pair \((0, T)\) is a forgery because \( H(\hat{x}) = H(0) \).