Homework 8: Fully homomorphic encryption

Total of 145 points

1. (45 points) The *Pallier cryptosystem* is defined as follows:

   - **Key generation:** Choose $m = pq$ where p, q are random n bit primes (e.g., chosen randomly in the interval $[2^{n-1}, 2^n]$). Choose g to be a random element in $\mathbb{Z}^*_{m^2}$. The public key is m, g and the secret key is the factorization p, q.
   - **Verifying keys:** Define λ be the least common multiple of $(p-1)$ and $(q-1)$ and for every $a \in \mathbb{Z}^*_{m^2}$ let $\Lambda(a) = \lfloor \frac{(a^\lambda \mod m^2)-1}{m} \rfloor (\mod m)$. If $gcd(\Lambda(g), m) \neq 1$ then regenerate the keys.
   - **Encryption:** To encrypt a message $x \in \mathbb{Z}*_m$, choose r at random in $\mathbb{Z}^*_m$ and output $g^x r^m (\mod m^2)$.
   - **Decryption:** To decrypt the ciphertext c, output $\Lambda(c)\Lambda(g)^{-1} (\mod m)$.

   It can be shown to be CPA secure under reasonable number theoretical assumptions, similar to those underlying RSA.

   a. (15 points) Prove that if the key passes the check then the order of g in $\mathbb{Z}^*_{m^2}$ is a multiple of m.

   b. (15 points) Prove that decryption works: for every message $x \in \mathbb{Z}^*_m$, the decryption of the encryption of x is x (i.e., $D_d(E_e(x)) = x$).

   c. (15 points) Prove that it is *additively homomorphic*: for every $x, x' \in \mathbb{Z}^*_m$, $D_d(E_e(x)E_e(x') (\mod m^2)) = x + x' (\mod m)$.

   After you finish this exercise you might want to look at this document from 2009 showing how Pallier encryption can be used for electronic voting. Note the list of "joke candidates" for the world parliament they used on page 8.

2. (60 points) Consider the following zero knowledge protocol based on a homomorphic encryption scheme $(G, E, D, EVAL)$ for a family $\mathcal{F}$:

   The public input is $f \in \mathcal{F}$ where $|f| = poly(n)$ and maps $\{0,1\}^n$ to $\{0,1\}$. Alice has an input $x \in \{0,1\}^m$ such that $f(x) = 1$ and wants to prove to

Bob that there exists such an x. We consider the following protocol for this task:

Protocol 1:

- Alice generates (e, d) keys for the encryption scheme, and sends e, together with $(c_1, \ldots, c_n)$ where $c_i = E_e(x_i)$ to Bob
- Bob chooses $b \leftarrow_R \{0,1\}$. If $b = 1$ then Bob sends $c' = EVAL(c_1, \ldots, c_n, f)$ to Alice and if $b = 0$ then Bob sends $c' = E_e(0)$.
- Alice responds with $b' = D_d(c')$ and Bob accepts if $b = b'$.

Lets say that the scheme has the *identical ciphertexts* property if for every string e there exists a unique private key d such that (e, d) has nonzero probability to be output by the generator G and for every ciphertexts $c_1, \ldots, c_n$ and function $f \in \mathcal{F}$, the distribution of $EVAL(c_1, \ldots, c_n, f)$ and $E_e(f(D_d(x_1), \ldots, D_d(x_n)))$ is *identical*. (The randomness in this distribution is only over the potential internal coins of $EVAL$ and E.[1])

a. (20 points) Prove that if the scheme has the identical ciphertexts property then Protocol 1 is honest verifier zero knowledge, in the sense that it satisfies completeness (with probability 1), soundness (success in cheating is at most 0.9) and the zero knowledge property is guaranteed if the verifier follows the protocol.

b. (20 points) Prove that Protocol 1 is *not* zero knowledge with respect to a malicious verifier by giving a verifier strategy that always succeeds in learning the 17th bit of x.

Consider now the following protocol:

Protocol 2:

- Alice generates (e, d) keys for the encryption scheme, and sends e, together with $(c_1, \ldots, c_n)$ where $c_i = E_e(x_i)$ to Bob
- Bob chooses $b \leftarrow_R \{0,1\}$ and $z \leftarrow_R \{0,1\}^{3n}$. If $b = 1$ then Bob sends z and $c' = EVAL(c_1, \ldots, c_n, f)$ to Alice and if $b = 0$ then Bob sends z and $c' = E_e(0)$.
- Alice computes $b' = D_d(b)$ and chooses $w \leftarrow_R \{0,1\}^n$ and sends to Bob $y = PRG(w) + b'z(\mod 2)$.
- Bob sends to Alice all the internal randomness used to compute c'.
- Alice checks that c' was indeed computed as desired and if so sends to Bob w and b'. Bob accepts if it indeed holds that $y = PRG(w) + b'z(\mod 2)$ and $b' = b$.

[1] The $EVAL$ procedures we saw in the lectures were deterministic but we can get probabilistic variants of them that satisfy a close enough property to the weak identical ciphertexts property.

c. (20 points) Prove that if the scheme has the identical ciphertexts property then Protocol 2 is a zero knowledge protocol (without the need for zero knowledge)

3. (40 points) Hash functions and public key encryption seem like very different creatures, but here we will show a relation between the two. Recall that a collection of functions H mapping $10n$ bits to n bits is a *collision resistant hash function* collection if for every efficient adversary A, if A is given h chosen at random from H, then the probability that A outputs $x \neq x' \in \{0,1\}^{10n}$ such that $H(x) = H(x')$ is negligible.

Let $(G, E, D, EVAL)$ be an XOR homomorphic encryption scheme with respect to XOR's of $10n$ ciphertexts, where n is the length of the ciphertext. Consider the following collection of functions H: to choose a random $h \in H$, we generate keys (e, d) for the homomorphic crypto system, and let $c_1, \ldots, c_{10n}$ be n independent encryptions of zero. The function h will map $x \in \{0,1\}^{10n}$ to $EVAL(f, c_1, \ldots, c_{10n}$ where $f(b_1, \ldots, b_{10n}) = \oplus_{i:x_i=1} b_i$.

Prove that if the scheme is CPA secure then this hash function collection is collision resistant. See footnote for hint[2]

[2]If there is such an adversary A and we are given a ciphertext c that we don't know if it is an encryption of 0 or an encryption of 1, we can make $10n - 1$ encryptions of 0, and plug c in some random index i among them to obtain $c_1, \ldots, c_{10n-1}$ such that $c_i = c$, and use this tuple to define h. Then we could argue that the probability that the adversary outputs x and x' that satisfy $h(x) = h(x')$ but disagree on the i^{th} bit is different depending on whether c was an encryption of 0 opposed to when it was an encryption of 1.