Homework 10: Public key crypto review

Total of 170 points

1. (50 points) Here is one possible security definition for a witness encryption scheme: it is composed of two efficient algorithms (E, D) with the following property. E is a probabilistic algorithm that takes as input a circuit $C : \{0,1\}^n \rightarrow \{0,1\}$ and a message $b \in \{0,1\}$ and outputs $c = E_C(b)$. D takes as input a string w and a ciphertext c, and the condition we require is that if $C(w) = 1$ then $D_w(E_C(b)) = b$. The notion of security is that if there exists no w such that $C(w) = 1$ then the distributions $E_C(0)$ and $E_C(1)$ are computationally indistinguishable (the distributions are over the coins of the encryption algorithm).

a. (25 points) Prove that under the PRG assumption, witness encryption implies a public key encryption scheme. See footnote for hint[1]

b. (25 points) Give a construction of a witness encryption scheme using an indistinguishability obfuscator $\mathcal{O}$. See footnote for hint[2]

2. (60 points) A *puncturable PRF* is a pseudorandom function collection $\{f_s\}$ such that for every input x^*, there is a way to map an index s into an index $s^* = PUNCTURE(s, x^*)$ that allows to compute the function f_s on *every input except* x^*. That is, there is some efficient algorithm $EVAL$ such that $EVAL(s^*, x) = f_s(x)$ for every $x \neq x^*$ but such that even given s^*, the value $f_s(x^*)$ is comptuationally indstinguishable from a uniform value in $\{0,1\}^n$.

a. (30 points) Show that under the PRG assumption, there exists a puncturable PRF. See footnote for hint[3]

b. (30 points) Suppose that $\mathcal{O}$ is an IO obfuscator, $G : \{0,1\}^n \rightarrow \{0,1\}^{3n}$ is a PRG and that $\{f_s\}$ (where $f_s : \{0,1\}^{|s|} \rightarrow \{0,1\}^{|s|}$ is a puncturable PRF. Prove that the following is a *selectively secure* digital signature scheme,

[1] The public key can be a string $y = G(w)$ where $G : \{0,1\}^n \rightarrow \{0,1\}^{2n}$ is a PRG, and the private key can be w.

[2] One can phrase the goal of the encryption algorithm in a witness encryption scheme as transforming the circuit C and message b to some C' that maps w to b if $C(w) = 1$ and maps w to `error` (that can be encoded in some for, e.g., as 0) if $C(w) = 0$. Of course one needs to ensure that it won't be possible to extract b from C' if there is no w satisfying $C(w) = 1$.

[3] hint3

where by this we mean a scheme that satisfies the relaxed definition where the attacker must declare the message m^* on which she will forge a signature at the beginning of the chosen-message-attack game, before seeing the public key.

- **Key generation:** The signing key is s and the public key is $V = \mathcal{O}(V_s)$ where $V_s(m, \sigma)$ outputs 1 if $G(\sigma) = G(f_s(m))$ and outputs 0 otherwise.
- **Signature:** To sign m with key s, we output $f_s(m)$
- **Verification:** To verify (m, σ) with key V, run $V(m, \sigma)$

As a first step, worth 15 points, for every m^*, consider the following circuit $V^*_{m^*,s^*,z}$: for $m \neq m^*$ $V^*_{m^*,s^*,z}(m, \sigma)$ outputs 1 iff $G(EVAL(s^*, m)) = G(\sigma)$ and for $m = m^*$, $V^*_{m^*,s^*,z}(m, \sigma)$ outputs 1 iff $G(\sigma) = z$. Prove that if $s^* = PUNCTURE(m^*)$ and $z = G(f_s(m^*))$ then $V^*_{m^*,s^*,z}$ computes the same function as V_s. By padding you can assume they have the same size as well.

See footnote for a hint how to complete the proof[4]

3. (60 points) Suppose that Bob wants Alice to compute for him a function $f(x)$ that is polynomial time computable but still takes too much time for him to compute online (though he can invest this time in a preprocessing step, before he learns the input x he needs to compute it for). Consider the following protocols for doing so using an FHE $(G, E, D, EVAL)$. We will also assume $EVAL$ is a deterministic function.

Protocol 1:

- **Preprocessing step:** Bob computes generates keys (e, d) for the FHE, and computes $c_* = E_e(0^n)$ and $c'_* = EVAL(f, c^*)$. He sends e to Alice.
- **Bob's input:** $x \in \{0, 1\}^n$.
- **Bob->Alice:** Bob chooses $b \leftarrow_R \{0, 1\}$. Bob lets $c_b = c_*$ and $c_{1-b} = E_e(x)$ and sends c_0, c_1 to Alice.
- **Bob<-Alice:** Alice computes $c'_0 = EVAL(f, c_0)$, $c'_1 = EVAL(f, c_1)$ and sends c'_0, c'_1 to Bob.
- **Bob's output:** If $c'_b \neq c'_*$ Bob rejects. Otherwise, he outputs $D_d(c'_{1-b})$.

a. (20 points) Prove that the protocol satisfies the following notion of security: for every efficient strategy A for Alice, either Bob rejects with probability at least 1/3 or Bob outputs the correct output with probability at least 1/3.

b. (20 points) Suppose that we run Protocol 1 *twice* for two inputs x_1, x_2 with the same preprocessing step. The notion of security is now that for

[4] Think of the following series of hybrids. First we can modify the key from the obfuscation of V_s to the obfuscation of $V_{m^*,s^*,G(f_s(m^*))}$ and claim that the attackers success probability will stay the same due to the security of the IO scheme. Then we can transform the last output to $G(U_n)$ and claim that there the success would still be the same due to the punctured PRF security. Finally we can modify the value $G(U_n)$ to U_{3n} and claim that the suceess should still be the same due to the security of the PRG. But at this point, eith very high probability the verification algorithm $V_{m^*,s^*,z}$ outputs 0 on *every* input of the form (m^*, σ).

every efficient strategy A for Alice, either Bob rejects with probability at least 1/3 or Bob outputs the correct outputs for both x_1 and x_2 (i.e., $f(x_1)$ and $f(x_2)$) with probability at least 1/3. Prove that this protocol satisfies this notion of security or give a counterexample (a strategy for Alice that would violate this property).

c. (20 points) Consider the following protocol:

Protocol 2:

- **Preprocessing step:** Bob computes generates two independent pairs of keys (e,d) (e',d') for the FHE, and computes $c_* = E_e(0^n)$ and $c'_* = EVAL(f,c^*)$. He sends e,e' to Alice.
- **Bob's input:** $x \in \{0,1\}^n$.
- **Bob->Alice:** Bob chooses $b \leftarrow_R \{0,1\}$. Bob lets $c_b = c_*$ and $c_{1-b} = E_e(x)$ and sends $c'_0 = E_{e'}(c_0), c'_1 = E_{e'}(c_1)$ to Alice.
- **Bob<-Alice:** Alice defines the function $g(c) = EVAL(f,c)$ computes $c''_0 = EVAL(g,c'_0)$, $c''_1 = EVAL(g,c'_1)$ and sends $c''_0,' c'_1$ to Bob.
- **Bob's output:** If $D_{d'}(c''_b) \neq c'_*$ Bob rejects. Otherwise, he outputs $D_{d'}(D_d(c''_{1-b}))$.

Prove that for every polynomial k and $x_1,\ldots,x_k$, Protocol 2 satisfies the property that if we run the processing step once and then run the protocol k times with inputs $x_1,\ldots,x_k$ then for every efficient strategy of Alice, either Bob rejects with probability at least 1/3, or he outputs all the correct k outputs with probability at least 1/3.