In this problem, you will use a PRG to implement what we’ll call a secure “locking” scheme. A locking scheme is a protocol between two players, a locker \( L \) and a verifier \( V \). It allows \( L \) to lock itself into one of two choices (0 or 1) without \( V \) knowing which choice was made, then later reveal its choice. The protocol works in two phases: in the first “locking” phase, \( L \) and \( V \) exchange some messages, which result in \( L \) being bound to its (secret) choice bit. In the second “unlocking” phase, \( L \) reveals its choice bit and some additional information, which allows \( V \) to check consistency with the earlier messages.

We define the following model for a locking scheme, in which the locking phase consists of an initial message from the verifier, followed by a response from the locker.

- The verifier \( V() \) is a PPT algorithm that takes no input (except for the implicit security parameter \( 1^n \) and its random coins) and outputs some message \( v \in \{0, 1\}^* \).
- The locker \( L(\sigma, v; r_L) \) is a PPT algorithm that takes a choice bit \( \sigma \in \{0, 1\} \), the verifier’s initial message \( v \), and random coins \( r_L \), and outputs some message \( \ell \in \{0, 1\}^* \).

In the unlocking phase, the locker simply reveals \( \sigma \) and \( r_L \), and the verifier checks that \( \ell = L(\sigma, v; r_L) \).

(a) (3 points) A secure locking scheme should be “hiding,” i.e., a malicious (but computationally bounded) verifier \( V^* \) should not be able to learn anything about the honest locker \( L \)’s choice bit \( \sigma \), no matter what initial message \( v^* \) the malicious verifier sent.

Using the notion of indistinguishability, give a formal definition of this hiding property.

(b) (3 points) A secure locking scheme should also be “binding” against even a computationally unbounded malicious locker \( L^* \). That is, there should not exist any \( \ell^* \) that can successfully be unlocked as both choice bits \( \sigma \in \{0, 1\} \), except with negligible probability over the choice of the honest verifier \( V \)’s initial message \( v \).

Give a formal definition of this binding property.

(c) (3 points) Let \( G \) be any length-tripling function, i.e., one for which \( |G(x)| = 3|x| \) for every \( x \in \{0, 1\}^* \). Give an upper bound on the probability, over the choice of a random \( 3n \)-bit string \( R \), that there exist two inputs \( x_1, x_2 \in \{0, 1\}^n \) such that \( G(x_1) \oplus G(x_2) = R \).

(d) (6 points) Let \( G \) be a length-tripling PRG (which we have seen can be obtained from any PRG). Use \( G \) to construct a secure locking scheme, and prove that it is both hiding and binding according to your definitions. [I need a hint! (ID 19922)]

---

\(^1\)From Peikert
(e) (0 points) Think how using the locking scheme two remote parties can toss a fair coin over the Internet, even if one of them is dishonest. For more discussion and cool applications, see Dodis’s lecture 14.