POIS ASSIGNMENT 1

TASK 2

BUILD A PROVABLY SECURE PRF FROM PRG

THEORY

PRF : Pseudo Random Function
PRG : Pseudo-random Number Generator

A pseudorandom function is a deterministic function of a key and an input that is indistinguishable from a truly random function of the input. More precisely, let s be a security parameter, let K be a key of length s bits, and let f (K,x) be a function on keys K and inputs x. Then f is a pseudo-random function if:

- f can be computed in polynomial time in s
- if K is random, then f cannot be distinguished from a random function in polynomial time.

In this context, "distinguishability" refers to the ability of an algorithm to tell whether a function is not truly random.

Let g be a truly random function of x with the same output length as f. Suppose a polynomial-time algorithm A is given access to a "oracle" which, on input x, either consistently returns f (K, x), or consistently returns g(x). After some (polynomial) number of accesses to the oracle, the algorithm outputs a guess, b, as to whether the oracle is f or g.

Suppose Alice wishes to authenticate herself to Bob, by proving she knows a secret that they share. With PRNG's they could proceed as follows. They both seed a PRNG with the shared

But this solution requires state, and they both have to compute i random numbers. Instead, we would like "random access" to the sequence. This is the intuition behind pseudo-random functions: Bob gives alice some random i, and Alice returns $F_K(i)$, where $F_K(i)$ is indistinguishable from a random function, that is, given any $x_1, \ldots, x_m, F_K(x_1), \ldots, F_K(x_m)$, no adversary can predict $F_K(x_{m+1})$ for any x_{m+1}

secret. Bob picks and sends Alice some random number i, and Alice proves she knows the shared secret by responding with the ith random number generated by the PRNG.

Definition: a function $f : \{0,1\}^n \times \{0,1\}^s \to \{0,1\}^m$ is a (t, ϵ, q)-PRF if

- Given a key $K \in \{0,1\}^s$ and an input $X \in \{0,1\}^n$ there is an "efficient" algorithm to compute $F_K(X) = F(X, K)$.
- For any t-time oracle algorithm A, we have

$$|Pr_{K \leftarrow \{0,1\}^s}[A^{f_K}] - Pr_{f \in \mathcal{F}}[A^f]| < \epsilon$$

where $\mathcal{F} = \{f : \{0,1\}^n \to \{0,1\}^m\}$ and A makes at most q queries to the oracle.

Pseudorandom functions are vital tools in the construction of cryptographic primitives, especially secure encryption schemes.

In more detail, let G : {0,1}λ → {0,1}2λ be a length-doubling PRG. For convenience, let GL</sub>(k) and GR(k) denote the first λ bits and last λ bits of $G(k)$, respectively. The main idea is to imagine a complete binary tree of height *in* (*in* will be the input length of the PRF), where each node in the tree represents an application of G. If a node gets input k, then it sends GL(k) to its left child and sends GR(k) to its right child. There will be 2in leaves, whose values will be the outputs of the PRF. To access the leaf with index $x \in$ {0,1}in. we can traverse the tree from root to leaf, taking left and right turns at each node according to the bits of x.