**Anish Banerjee, Shankh Gupta** **Problem Set - 1**

COL759: Cryptography August 2023

## Problem 1: Perfect 2 time security

*Solution:*

## Problem 2 : Secure/Insecure PRGs and PRFs

*Solution:*

(a) PRGs

i. $\mathcal{G}' = \left\{ G'_n : \{0,1\}^{2n} \to \{0,1\}^{3n} \right\}_{n \in \mathbb{N}}$, where

$$G'_n(s_1 \mid\mid s_2) = G_n(s_1) \wedge G_n(s_2).$$

ii. $\mathcal{G}' = \left\{ G'_n : \{0,1\}^{2n} \to \{0,1\}^{3n} \right\}_{n \in \mathbb{N}}$, where

$$G'_n(s_1 \mid\mid s_2) = G_n(s_1) \oplus G_n(s_2).$$

(b) PRFs

i. $\mathcal{F}' = \left\{ F'_n : \{0,1\}^{n} \times \{0,1\}^{2n} \to \{0,1\}^{n} \right\}_{n \in \mathbb{N}}$ where

$$F'_n(k, (x_1, x_2)) = F_n(k, x_1) \oplus F_n(k, x_2).$$

The given family $\mathcal{F}'$ is **insecure**. Consider a PPT attacker $\mathcal{A}$ who sends $\mathsf{poly}(\lambda)$ distinct $(x_i, x_i)$ queries to the challenger. If the challenger chooses $b = 0$ then it will end up sending

$$F_n(k, x_i) \oplus F_n(k, x_i) = 0^n$$

for each of the queries. The attacker outputs 0 if all the responses are 0 and 1 otherwise. Advantage of the attacker is close to 1.

ii. $\mathcal{F}' = \{F'_n : \{0,1\}^{n} \times \{0,1\}^{n} \to \{0,1\}^{n}\}_{n \in \mathbb{N}}$ where

$$F'_n(k, x) = F_n(k, x) \oplus x.$$

The given family is secure. Given an adversary $\mathcal{A}$ which breaks PRF security of $\mathcal{F}'$, we can construct an adversary $\mathcal{B}$ which breaks the security of $\mathcal{F}$.

**Problem 2(b)(ii)**

- Challenger picks a uniformly random bit $b \leftarrow \{0,1\}$ and a seed $s \leftarrow \{0,1\}^n$.
- The adversary $\mathcal{A}$ makes polynomially many queries to $\mathcal{B}$, who passes them to the challenger. Challenger replies as in the PRF Game.
- Upon receiving the response $y_i$ of each query, $\mathcal{B}$ sends $y_i \oplus x_i$ to $\mathcal{A}$
- After polynomially many queries, $\mathcal{B}$ forwards the response send by $\mathcal{A}$ ($b'$) and wins if $b = b'$.

Figure 1: Reduction for Problem 2(b)(ii)

## Problem 3 : PRG Security does not imply Related-Key-PRG Security

*Solution:*

## Problem 4 : Constructing PRFs from PRGs

*Solution:*