Homework 2

Due: September 29, 2023

CS 1510: Intro. to Cryptography and Computer Security

1 Two Indistinguishabilities

Fix two probabilistic sampling algorithms $D_1(1^k)$ and $D_2(1^k)$ which, on input the security parameter 1^k, output binary strings; both run in polynomial time.

The following probabilities are well-defined for any algorithm $\mathcal{A}$ that takes as input 1^k and a sample x from the union of the sample spaces of $D_1(1^k)$ and $D_2(1^k)$.

Experiment a: Let $i \leftarrow \{1,2\}$ be chosen uniformly at random from $\{1,2\}$, and let $x \leftarrow D_i(1^k)$ be sampled according to the sampling algorithm $D_i(1^k)$.

The probability $c_{\mathcal{A}}(k)$ is the probability that the algorithm $\mathcal{A}$ *chooses* the correct sampling algorithm given a sample x from Experiment a; that is:

$$c_{\mathcal{A}}(k) = \Pr[i \leftarrow \{1,2\}; x \leftarrow D_i(1^k); i' \leftarrow \mathcal{A}(1^k, x) : i' = i]$$

Experiment b_1: Let $x \leftarrow D_1(1^k)$.

The probability $z_{\mathcal{A},1}(k)$ is the probability that the algorithm $\mathcal{A}$ outputs *zero* given a sample from Experiment b_1; that is:

$$z_{\mathcal{A},1}(k) = \Pr[x \leftarrow D_1(1^k); i \leftarrow \mathcal{A}(1^k, x) : i = 0]$$

Experiment b_2: Let $x \leftarrow D_2(1^k)$.

The probability $z_{\mathcal{A},2}(k)$ is the probability that the algorithm $\mathcal{A}$ outputs *zero* given a sample from Experiment b_2; that is:

$$z_{\mathcal{A},2}(k) = \Pr[x \leftarrow D_2(1^k); i \leftarrow \mathcal{A}(1^k, x) : i = 0]$$

Consider the following two definitions of computational indistinguishability:

Definition 1 (CIA indistinguishability) *Two sampling algorithms $D_1(1^k)$ and $D_2(1^k)$ are* CIA-indistinguishable *(computationally indistinguishable, variant A) if there exists a negligible function ν such that for all p.p.t. algorithms $\mathcal{A}$,*

$$c_{\mathcal{A}}(k) = \Pr[i \leftarrow \{1,2\}; x \leftarrow D_i(1^k); i' \leftarrow \mathcal{A}(1^k, x) : i' = i] \leq \frac{1}{2} + \nu(k)$$

We denote this by $D_1(1^k) \approx_a D_2(1^k)$.

CIA indistinguishability says that two distributions are indistinguishable if no computationally bounded adversary can determine from which distribution a random sample was chosen during Experiment a.

Definition 2 (CIB indistinguishability) *Two sampling algorithms $D_1(1^k)$ and $D_2(1^k)$ are* CIB-indistinguishable *(computationally indistinguishable, variant B) if there exists a negligible function ν such that for all p.p.t. algorithms $\mathcal{A}$,*

$$|z_{\mathcal{A},1}(k) - z_{\mathcal{A},2}(k)| \leq \nu(k)$$

where for each $i \in \{1,2\}$,

$$z_{\mathcal{A},i}(k) = \Pr[x \leftarrow D_i(1^k); b \leftarrow \mathcal{A}(1^k, x) : b = 0]$$

We denote this by $D_1(1^k) \approx_b D_2(1^k)$.

CIB indistinguishability says that two distributions are indistinguishable if no computationally bounded adversary can behave significantly differently on a sample chosen during Experiment b_1 versus a sample chosen during Experiment b_2.

In this problem, you will prove that these two definitions of computational indistinguishability are equivalent. That is, $D_1(1^k) \approx_a D_2(1^k)$ if and only if $D_1(1^k) \approx_b D_2(1^k)$.

a. First, prove that $D_1(1^k) \approx_b D_2(1^k)$ implies $D_1(1^k) \approx_a D_2(1^k)$. We'll prove this through a contradiction by assuming we have an adversary that can distinguish the two distributions by the CIA definition and prove that we can construct an adversary out of this that can distinguish by the CIB definition.

   (1) Let $\mathcal{A}$ be fixed. Assume without loss of generality that its only possible outputs are 1 and 2. (Otherwise, you can trivially improve performance as follows: If $\mathcal{A}$ outputs something that is not a 1 or a 2, turn it into a 1. This cannot make $\mathcal{A}$'s performance *worse*, and it might make it better.) Define:

   $$c_{\mathcal{A},1}(k) = \Pr[x \leftarrow D_1(1^k); i' \leftarrow \mathcal{A}(1^k, x) : i' = 1]$$

   In other words, $c_{\mathcal{A},1}(k)$ is the probability that $\mathcal{A}$ is correct given that x comes from $D_1(1^k)$. Similarly, define:

   $$c_{\mathcal{A},2}(k) = \Pr[x \leftarrow D_2(1^k); i' \leftarrow \mathcal{A}(1^k, x) : i' = 2]$$

   Express $c_{\mathcal{A}}(k)$ in terms of $c_{\mathcal{A},1}(k)$ and $c_{\mathcal{A},2}(k)$.

   (2) Define $\mathcal{A}'(1^k, x)$ as follows: Run $\mathcal{A}(1^k, x)$. Output 0 if $\mathcal{A}$ outputs 1, and output -1 otherwise. Express $z_{\mathcal{A}',i}(k)$ in terms of $c_{\mathcal{A},1}(k)$ and $c_{\mathcal{A},2}(k)$.

   (3) Express $c_{\mathcal{A},1}(k)$ and $c_{\mathcal{A},2}(k)$ in terms of $z_{\mathcal{A}',i}(k)$.

(4) Express $c_{\mathcal{A}}(k)$ in terms of $z_{\mathcal{A}',i}(k)$.

(5) Conclude that if $D_1(1^k) \approx_b D_2(1^k)$, then $D_1(1^k) \approx_a D_2(1^k)$.

b. Next, prove that $D_1(1^k) \approx_a D_2(1^k)$ implies $D_1(1^k) \approx_b D_2(1^k)$.

(1) Show that if $D_1(1^k) \approx_a D_2(1^k)$, then for a fixed $\mathcal{A}'$ for distinguishing distributions in the CIB definition and k, we can construct another adversary $\mathcal{A}$ for distinguishing distributions in the CIA definition to obtain the relation in step (4) of part (a).

(2) Argue that this implies $D_1(1^k) \approx_b D_2(1^k)$.

2 PRGs

Let $G_1 : \{0,1\}^n \to \{0,1\}^{2n}$ and $G_2 : \{0,1\}^n \to \{0,1\}^{2n}$ be length-doubling PRGs for all n.

For each of the following, either prove that it is a secure PRG or provide a counterexample to show that it is not necessarily a PRG. In constructing a counterexample for this problem, you can assume that a PRG G exists. From such G, show that some contrived G_1 and/or G_2 can be constructed such that they are themselves PRGs, but, when you plug them into the given PRGs G_a, G_b, and/or G_c (whichever construction you're showing is not a PRG), yields something that is not a PRG.

a. $G_a(s) = G_1(s) \oplus G_2(s)$.

b. $G_b(s) = s_1 \| G_1(s_2)$ where $s = s_1 \| s_2$ and $|s_1| = |s_2| = n$. (i.e. s_1 is the first half of the input, s, and s_2 is the second half). Note this means we have $G_b : \{0,1\}^{2n} \to \{0,1\}^{3n}$.

c. $G_c(s) = G_1(s) \oplus_p s$, where $\oplus_p$ denotes "padded XOR," where if we're XORing strings of unequal length, we pad the shorter string with as many 0s on the right hand-side as is needed to make it the correct length. For example, $1010 \oplus_p 110011 = 101000 \oplus 110011 = 011011$.

3 GGM and Prefix-Constrained PRFs

A PRF $F : \{0,1\}^k \times \{0,1\}^k \mapsto \{0,1\}^k$ is said to be a prefix-constrained PRF if given the PRF key it is possible to generate a *constrained* PRF key K_π which lets you evaluate the PRF only at inputs which have a specific prefix π. More precisely, a prefix-constrained PRF has the following algorithms:

Setup : $\mathsf{Setup}(1^k)$ outputs a key $K \leftarrow \{0,1\}^k$

Constrain : For any string π such that $|\pi| \le k$, $\mathsf{Constrain}(K, \pi)$ outputs a key K_π

Evaluate : $\mathsf{Eval}(K_\pi, x)$ outputs $F(K, x)$ iff. $x = \pi \| t$ for some $t \in \{0,1\}^{k-|\pi|}$, else output fail

The security notion for a constrained PRF key K_π is that it should reveal no information about the PRF evaluation at points that do not have the prefix π. More precisely, for any string π such that $|\pi| \le k$, let X_π be the set of all $x \in \{0,1\}^k$ that <u>do not</u> have π as their prefix. We say $F : \{0,1\}^k \times \{0,1\}^k \mapsto \{0,1\}^k$ is a *spring-break*-secure prefix-constrained PRF if for all PPT $\mathcal{A}$, there exists a negligible $\nu(\cdot)$ such that

$$\left|\Pr[\mathcal{A}(1^k) \text{ is in Exp 1 } : b' = 0] - \Pr[\mathcal{A}(1^k) \text{ is in Exp 2 } : b' = 0]\right| \le \nu(k)$$

where

Exp 1	**Exp 2**
Choose key $K \leftarrow \mathsf{Setup}(1^k)$	Choose key $K \leftarrow \mathsf{Setup}(1^k)$ Choose random function $R : \{0,1\}^k \mapsto \{0,1\}^k$
$\mathcal{A}(1^k)$ chooses a prefix π with $\lvert\pi\rvert \le k$ and obtains $K_\pi = \mathsf{Constrain}(K, \pi)$	$\mathcal{A}(1^k)$ chooses a prefix π with $\lvert\pi\rvert \le k$ and obtains $K_\pi = \mathsf{Constrain}(K, \pi)$
$\mathcal{A}(1^k)$ adaptively queries $F(K, \cdot)$ on any inputs $x_1, \ldots, x_q \in X_\pi$ and obtains values $F(K, x_i)$ for $1 \le i \le q$	$\mathcal{A}(1^k)$ adaptively queries $R(\cdot)$ on any inputs $x_1, \ldots, x_q \in X_\pi$ and obtains values $R(x_i)$ for $1 \le i \le q$
$\mathcal{A}$ outputs a guess b'	$\mathcal{A}$ outputs a guess b'

In this problem, we will prove that the Goldreich-Goldwasser-Micali (GGM) PRF is also a prefix-constrained PRF. The GGM PRF is obtained as follows: Start with a length-doubling PRG $G : \{0,1\}^k \to \{0,1\}^k \times \{0,1\}^k$. So $G(s)$ for any $s \in \{0,1\}^k$ outputs a string of length $2k$; We will call the first half as $G_0(s)$ and second half as $G_1(s)$. Let input be $x = x_1 x_2 \ldots x_k$ where each $x_i \in \{0,1\}$, then the PRF, with key K is defined as follows:

$$F(K, x_1 x_2 \ldots x_k) = G_{x_k}(\ldots G_{x_2}(G_{x_1}(K))\ldots)$$

a. For the GGM PRF, what could be the constrained key K_0 that lets you evaluate $F(K, x)$ for all x starting with a 0? How will you evaluate the PRF with this constrained key?

b. Design the $\mathsf{Constrain}(K, \pi)$ algorithm for any prefix π with $|\pi| \le k$ for the GGM PRF.

c. Describe the corresponding $\mathsf{Eval}(K_\pi, x)$ algorithm.

d. Prove that your prefix-constrained PRF is *spring-break*-secure.

4 Leaky PRF

Construct a PRF $F : \{0,1\}^{k+1} \times \{0,1\}^n \mapsto \{0,1\}^n$ with the property that, if an adversary learns the first bit of the secret key of the PRF, then F *is* distinguishable from random. You may assume that PRFs exists, and use another PRF in your construction.

a. Prove that your construction of F is a PRF.

b. Show how the adversary can distinguish F from random if it knows the first bit of the secret key.

5 Summary Question

Summarize the most important insights from this week's material, including from the lectures, notes, textbooks, homework problems, and other resources you find helpful, into a one-page resource. You will be permitted to use this one-page resource (along with the other weeks' resources) on the midterm and final.

Changes to this document prior to the exams are permitted, but for each change, you will be asked to state what you changed and why. For example, if you dropped something and replaced it with something else, justify why the thing you dropped wasn't as important as the thing you inserted, why you think it might be more useful for the exam, etc.

Please note that the purpose of this question is to help you organize and synthesize the material for your own future use. It will be graded based on completion—we will not be checking it for correctness.