CS 6501: Advanced Topics in Cryptography **Spring 2019**

Problem Set 1

Due: February 1, 2019 at 5pm (submit via Gradescope) **Instructor:** David Wu

Instructions: You **must** typeset your solution in LaTeX using the provided template:

`https://www.cs.virginia.edu/dwu4/courses/sp19/static/homework.tex`

Submission Instructions: You must submit your problem set via Gradescope. Please use course code **9YD875** to sign up. Note that Gradescope requires that the solution to each problem starts on a **new page**.

Problem 1: Definitions and Reductions [15 points].

(a) Let $F\colon \{0,1\}^\lambda \times \{0,1\}^\lambda \to \{0,1\}^\lambda$ be a secure PRF. Let $F'\colon \{0,1\}^\lambda \times \{0,1\}^\lambda \to \{0,1\}^\lambda$ be the function

$$F'(k,x) := \begin{cases} F(k,x) & x \neq 0^\lambda \\ 0^\lambda & x = 0^\lambda. \end{cases}$$

Is F' a secure PRF? If so, give a proof; otherwise, describe an attack.

(b) Let $F\colon \{0,1\}^\lambda \times \{0,1\}^\lambda \to \{0,1\}^\lambda$ be a secure PRF. Let $F'\colon \{0,1\}^\lambda \times \{0,1\}^\lambda \to \{0,1\}^\lambda$ be the function

$$F'(k,x) := \begin{cases} F(k,x) & x \neq k \\ 0^\lambda & x = k. \end{cases}$$

Is F' a secure PRF? If so, give a proof; otherwise, describe an attack.

(c) Let $F\colon \{0,1\}^\lambda \times \{0,1\}^\lambda \to \{0,1\}^\lambda$ be a secure PRF. Let $\Pi_{\mathsf{MAC}} = (\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify})$ be a candidate MAC with key-space $\{0,1\}^\lambda$, message space $\{0,1\}^\lambda$ and tag space $\{0,1\}^{2\lambda}$ defined as follows:

- $\mathsf{KeyGen}(1^\lambda) \to k$: Output $k \xleftarrow{\text{R}} \{0,1\}^\lambda$.
- $\mathsf{Sign}(k,m) \to t$: Output $F(k,0^\lambda) \| F(k,m)$.
- $\mathsf{Verify}(k,m,t) \to \{0,1\}$: Output 1 if $t = \mathsf{Sign}(k,m)$ and 0 otherwise.

Is Π_{MAC} a secure MAC? If so, give a proof; otherwise, describe an attack.

Problem 2: Key Leakage in PRFs [15 points]. Let $F\colon \{0,1\}^\lambda \times \{0,1\}^\lambda \to \{0,1\}$ be a secure PRF.

(a) Use F to construct a function $F'\colon \{0,1\}^{\lambda+1} \times \{0,1\}^\lambda \to \{0,1\}$ with the following two properties:

- F' is a secure PRF.
- If the adversary learns the last bit of the key, then F' is no longer secure.

In your solution, you should prove that F' is a secure PRF and describe an attack when the adversary knows the last bit of the PRF key. This shows that leaking even a *single* bit of the secret key can completely destroy the PRF security property. [**Hint:** Try changing the value of F at a single point.]

(b) Use F to construct a function $F'\colon \{0,1\}^{2\lambda} \times \{0,1\}^\lambda \to \{0,1\}$ such that F' is a secure PRF even against adversaries that can learn *any* single bit of the key to F'. Prove that your function F' is secure.

Problem 3. PRGs from One-Way Permutations [30 points]. In this problem, we will explore how to build PRGs from one-way permutations.[1] We will use the following definition of a one-way permutation:

> **Definition (One-Way Permutation).** Fix a security parameter $\lambda \in \mathbb{N}$ and a non-decreasing polynomial $n = n(\lambda)$. Let $f = \{f_\lambda \colon \{0,1\}^n \to \{0,1\}^n\}_{\lambda \in \mathbb{N}}$ be a function ensemble. We say that f is a one-way permutation if f_λ is a permutation for all $\lambda \in \mathbb{N}$ and for all efficient adversaries $\mathcal{A}$,
>
> $$\Pr\left[x \xleftarrow{\text{R}} \{0,1\}^n : \mathcal{A}(f_\lambda(x)) = x\right] = \text{negl}(\lambda).$$

(a) Fix an $x \in \{0,1\}^n$. Suppose there exists an efficient algorithm $\mathcal{A}_x$ such that

$$\Pr\left[r \xleftarrow{\text{R}} \{0,1\}^n : \mathcal{A}_x(r) = \langle r, x\rangle \bmod 2\right] \ge 3/4 + \varepsilon$$

for some constant $\varepsilon > 0$. Here, $\langle r, x\rangle = \sum_{i \in [n]} x_i r_i$ denotes the inner product of x and r. Construct an efficient algorithm $\mathcal{B}$ that outputs x with probability at least $1/2$ by calling $\mathcal{A}_x$ at most $O(n \log n)$ times. [**Hint:** Use the union bound and the Chernoff bound.]

(b) Let $f = \{f_\lambda\}_{\lambda \in \mathbb{N}}$ be a function ensemble. Define $g = \{g_\lambda\}_{\lambda \in \mathbb{N}}$ where $g_\lambda \colon \{0,1\}^{2n} \to \{0,1\}^{2n}$ that implements the mapping $(r, x) \mapsto (r, f_\lambda(x))$ where $r, x \in \{0,1\}^n$. Show that if f is a one-way permutation, then g is also a one-way permutation.

(c) Let $g = \{g_\lambda\}_{\lambda \in \mathbb{N}}$ be the function ensemble from Part (b). Suppose that there exists an efficient algorithm $\mathcal{A}$ such that

$$\Pr\left[r, x \xleftarrow{\text{R}} \{0,1\}^n : \mathcal{A}(g_\lambda(r, x)) = \langle r, x\rangle \bmod 2\right] \ge 3/4 + \varepsilon,$$

for some constant $\varepsilon > 0$. Use your result from Part (a) to show that there exists an efficient algorithm that breaks the one-wayness of g. We refer to the bit $\langle r, x\rangle \bmod 2$ as a *hard-core bit* for the one-way permutation g: namely, predicting the hard-core bit is as hard as inverting g. [**Hint:** There are many ways to argue this, but one method is to use Markov's inequality.]

(d) In fact, it is possible to strengthen the result from Part (c) and show that if there exists an efficient adversary $\mathcal{A}$ such that

$$\Pr\left[r, x \xleftarrow{\text{R}} \{0,1\}^n : \mathcal{A}(g_\lambda(r, x)) = \langle r, x\rangle \bmod 2\right] \ge 1/2 + \varepsilon,$$

for any non-negligible ε, then there exists an efficient algorithm that breaks the one-wayness of g. Use this fact to show the following: if g is one-way, then the following two distributions are computationally indistinguishable:

$$\left\{r, x \xleftarrow{\text{R}} \{0,1\}^n : (g_\lambda(r, x), \langle r, x\rangle \bmod 2)\right\} \text{ and } \left\{r, x \xleftarrow{\text{R}} \{0,1\}^n, b \xleftarrow{\text{R}} \{0,1\} : (g_\lambda(r, x), b)\right\}.$$

Your argument to this problem shows that an *unpredictable* bit is in fact *pseudorandom*. [**Hint:** Show that if there exists an efficient algorithm that distinguishes $\langle r, x\rangle \bmod 2$ from random (given $g_\lambda(r, x)$) with probability ε, then there exists an efficient algorithm that can predict $\langle r, x\rangle \bmod 2$ (given $g_\lambda(r, x)$) with probability $1/2 + \varepsilon$.]

[1] A classic result by Håstad, Impagliazzo, Levin, and Luby show how to build a PRG from any one-way *function*.

(e) Let $f = \{f_\lambda\}_{\lambda \in \mathbb{N}}$ be a one-way permutation. Use the above results to show how to use f to construct a PRG with 1-bit stretch. Observe now that using the Blum-Micali construction described in class, this suffices to construct a PRG with arbitrary polynomial stretch (from one-way permutations).

Problem 4: Time Spent [5 points for answering]. How long did you spend on this problem set? This is for calibration purposes, and the response you provide will not affect your score.

Optional Feedback [0 points]. Please answer the following *optional* questions to help us design future problem sets. You do not need to answer these questions. However, we do encourage you to provide us feedback on how to improve the course experience.

(a) What was your favorite problem on this problem set? Why?

(b) What was your least favorite problem on this problem set? Why?

(c) Do you have any other feedback for this problem set?

(d) Do you have any other feedback on the course so far?