CS 355: Topics in Cryptography **Spring 2024**

Problem Set

Due: Friday, 12 April 2024 (submit via Gradescope)

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

`https://crypto.stanford.edu/cs355/24sp/homework.tex`

Submission Instructions: You must submit your problem set via Gradescope. Please use course code **RKN4PX** to sign up. Note that Gradescope requires that the solution to each problem starts on a **new page**.

Bugs: We make mistakes! If it looks like there might be a mistake in the statement of a problem, please ask a clarifying question on Ed.

Problem 1: Composition [10 points]. Determine whether each of the following statements is TRUE or FALSE. For the first two statements, prove that your answer is correct. That is, give a security proof or describe an adversary and prove it has high advantage.

Let $F(k,x) \to y$ be a PRF, and $f(x) \to y$ be a OWF, and let $G(s) \to r$ be a PRG. Assume that the PRF outputs fixed-length bit-strings.

1. The function $F_1(k,(x_1,x_2)) = F(k,x_1) \oplus F(k,x_2)$ is a PRF
2. The function $F_2((k_1,k_2),x) = F(k_1,x) \oplus F(k_2,x)$ is a PRF
3. The function $G_3(s) = (G(s),G(s))$ is a PRG
4. The function $G_4((s_1,s_2)) = (s_1,G(s_2))$ is a PRG (where $s_1,s_2 \in \mathcal{S}$: the seed space for G)
5. The function $f_5((x_1,x_2)) = (f(x_1),f(x_2))$ is a OWF

Problem 2: A weak form of the Goldreich-Levin Theorem [10 points]. We say that $b\colon \{0,1\}^* \to \{0,1\}$ is a hard*ish*-core bit of a permutation f if $b(x)$ is efficiently computable given x, and for every efficient algorithm $\mathcal{A}$ it holds that

$$\Pr[\mathcal{A}(f(x)) = b(x) : \ x \xleftarrow{R} \{0,1\}^n] \le 3/4 + \epsilon$$

for every positive constant ϵ.[1] We will prove that if $f\colon \{0,1\}^n \to \{0,1\}^n$ is a one-way permutation, then $b(x,r) = \langle x,r\rangle$ is a hardish-core bit of the permutation $g\colon \{0,1\}^{2n} \to \{0,1\}^{2n}$ defined as $g(x,r) = (f(x),r)$ for $x,r \in \{0,1\}^n$. Here $\langle x,r\rangle$ denotes the inner product of x and r mod 2.

For the sake of contradiction, suppose that there exists a constant $\epsilon > 0$, and an algorithm $\mathcal{A}$ that takes as input $f(x), r \in \{0,1\}^n$ and outputs a bit, such that

$$\Pr_{x,r}[\mathcal{A}(f(x),r) = \langle x,r\rangle] \ge 3/4 + \epsilon,$$

where the probability is taken over both x and r, each chosen uniformly from $\{0,1\}^n$. We will construct an algorithm $\mathcal{B}$ that breaks the one-wayness of f.

[1] An actual hard-core bit cannot be guessed with probability noticeably better than 1/2.

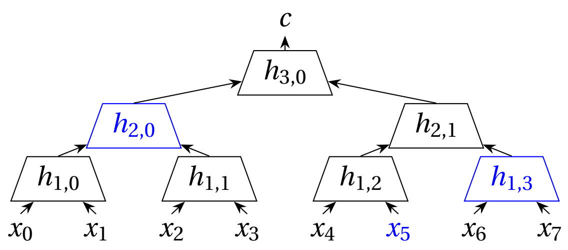

Figure 1: A Merkle tree

(a) As a warmup, suppose first that $\Pr[\mathcal{A}(f(x), r) = \langle x, r\rangle] = 1$. Show how to construct an efficient algorithm $\mathcal{B}$ that perfectly inverts f (i.e., given $f(x)$ outputs x).

(b) Next, we say that $x \in \{0,1\}^n$ is *good* for $\mathcal{A}$ if $\Pr_r[\mathcal{A}(f(x), r) = \langle x, r\rangle] \geq 3/4 + \epsilon$ for some positive constant ϵ, where the probability is taken *only* over $r \xleftarrow{R} \{0,1\}^n$. Construct an efficient algorithm $\mathcal{B}$ that takes as input $f(x)$ for a good $x \in \{0,1\}^n$ and outputs x with probability at least $1/2$, by calling $\mathcal{A}$ at most $O(n \cdot \log n)$ times.

(c) Show that x chosen uniformly from $\{0,1\}^n$ is *good* with some constant probability. Conclude that algorithm $\mathcal{B}$ breaks the one-wayness of f.

Problem 3: Vector Commitments [5 points]. In this problem we consider *vector commitments*: commitments to vectors which can be opened to one index at a time.

The classic construction vector commitment scheme is the *Merkle tree*, which is parameterized by a hash function $H : \mathcal{X} \times \mathcal{X} \to \mathcal{X}$. The core of this construction is a hash tree, as shown in Figure 1. Each internal node in the tree represents an evaluation of the hash function over the child nodes. The Commit procedure evaluates the hash tree, returning the root at the commitment. The IdxOpen procedure produces the siblings along a path (e.g., the blue nodes, for x_4), and the IdxVerify checks the hash evaluations along the path.

More precisely, the Merkle tree vector commitment scheme is defined by three algorithms:

- $\mathsf{Commit}(d \in \mathbb{N}, x \in \mathcal{X}^{2^d}) \to \mathcal{X}$:

```
    for $i \in \{0, 1, \ldots, 2^d - 1\}$: $h_{0,i} \leftarrow x_i$
    for $\ell \in \{1, 2, \ldots, d\}$, for $i \in \{0, 1, \ldots, 2^{d-\ell} - 1\}$: $h_{\ell,i} \leftarrow H(h_{\ell-1,2i}, h_{\ell-1,2i+1})$
    return $h_{d,0}$
```

- $\mathsf{IdxOpen}(d \in \mathbb{N}, i \in \mathbb{N}, x \in \mathcal{X}^{2^d}) \to \mathcal{X}^d$:

```
    compute $h_{\ell,i}$ as in Commit
    for $\ell \in \{0, \ldots, d-1\}$ : $p_i \leftarrow h_{\ell,((i \gg \ell) \oplus 1)}$[2]
    return $(p_0, \ldots, p_{d-1})$
```

[2]Here, $\gg$ is logical right shift and $\oplus$ is bitwise XOR, as in C. That is, $x \gg y$ denotes $\lfloor x/2^y \rfloor$ and $x \oplus y$ denotes the integer with binary representation equal to the bitwise XOR of the binary representations of x and y.

- $\mathsf{IdxVerify}(d \in \mathbb{N}, i \in \mathbb{N}, x, \in \mathcal{X}, c \in \mathcal{X}, p \in \mathcal{X}^d) \to \{0,1\}$:

```
$h_0 \leftarrow x$
for $\ell \in \{1, \ldots, d\}$:
    $j \leftarrow i \gg (\ell - 1)$
    if $j$ is odd: $h_\ell \leftarrow H(p_{\ell-1}, h_{\ell-1})$
    else: $h_\ell \leftarrow H(h_{\ell-1}, p_{\ell-1})$
return $h_d \overset{?}{=} c$
```

A vector commitment scheme is *index binding* if for all $d \in \mathbb{N}$ and for all efficient adversaries $\mathcal{A}$,

$$\Pr\left\{\mathsf{IdxVerify}(d,i,x,c,p) = 1 \wedge \mathsf{IdxVerify}(d,i,x',c,p') = 1 \wedge x \neq x' : (c,i,x,p,x',p') \leftarrow \mathcal{A}(d,\lambda)\right\} = \text{negl}(\lambda)$$

where λ is the security parameter of H. A hash function H is *collision resistant* if for all efficient adversaries $\mathcal{A}$,

$$\Pr\left\{H(x,y) = H(x',y') \wedge (x,y) \neq (x',y') : (x,y,x',y') \leftarrow \mathcal{A}(\lambda)\right\} = \text{negl}(\lambda)$$

Please **prove** that Merkle tree vector commitments are index binding if H is collision-resistant.

Problem 4: Key Leakage in PRFs [5 points]. Let F be a secure PRF defined over $(\mathcal{K}, \mathcal{X}, \mathcal{Y})$, where $\mathcal{K} = \mathcal{X} = \mathcal{Y} = \{0,1\}^n$. Let $\mathcal{K}_1 = \{0,1\}^{n+1}$. Construct a new PRF F_1, defined over $(\mathcal{K}_1, \mathcal{X}, \mathcal{Y})$, with the following property: the PRF F_1 is secure; however, if the adversary learns the last bit of the key then the PRF is no longer secure. 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.]

Optional Feedback [0 points]. Please answer the following questions to help us design future problem sets. You do not need to answer these questions, and if you would prefer to answer anonymously, please use this form. 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?

Problem 5: Time Spent [1 point 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.