CS 171: Problem Set 7

Due Date: March 18th, 2024 at 8:59pm via Gradescope. No late submissions will be accepted.

1 PRGs Imply OWFs (5 points)

We saw in lecture 12 that one-way functions (OWFs) can be used to construct pseudorandom generators (PRGs). Additionally, it turns out that PRGs can be used to construct OWFs. This means that the assumption that OWFs exist is equally strong as the assumption that PRGs exist.

Question: Let $G : \{0,1\}^{n/2} \to \{0,1\}^n$ be a PRG. Prove that G is also a OWF.

Solution

1. Overview: Assume toward contradiction that G is not a OWF. Then there exists a PPT adversary $\mathcal{A}$ that breaks OWF security by inverting G on a random input with non-negligible probability. Then we will use $\mathcal{A}$ to construct an adversary $\mathcal{B}$ that can break the PRG security of G. This is a contradiction because we are told that G satisfies PRG security. Therefore, our initial assumption was false, and in fact, G also satisfies OWF security.

2. Next, we will use $\mathcal{A}$ to construct a PPT adversary $\mathcal{B}$ that breaks the PRG security of G.

   Construction of $\mathcal{B}$:

   (a) The PRG challenger gives $\mathcal{B}$ a string $y \in \{0,1\}^n$. Either $y = G(x)$, where $x \leftarrow \{0,1\}^{n/2}$, or $y \leftarrow \{0,1\}^n$.
   (b) $\mathcal{B}$ runs $\mathcal{A}$ on inputs $(1^n, y)$. $\mathcal{A}$ outputs $x' \in \{0,1\}^{n/2}$.
   (c) $\mathcal{B}$ checks whether $G(x') = y$. If so, $\mathcal{B}$ outputs 1. If not, $\mathcal{B}$ outputs 0.

3. Pseudorandom Case: If $y = G(x)$, where $x \leftarrow \{0,1\}^{n/2}$, then $\mathcal{B}$ correctly simulates the OWF security game for G with adversary $\mathcal{A}$. In particular, $\mathcal{B}$ outputs 1 if and only if $\mathcal{A}$ wins the simulated OWF security game. Note that $\mathcal{A}$ wins with non-negligible probability because $\mathcal{A}$ breaks the OWF security of G.

4. Truly Random Case: We will show that if $y \leftarrow \{0,1\}^n$, then $\mathcal{B}$ outputs 1 with probability $\leq 2^{-n/2}$.

   The domain of G has size $2^{n/2}$, so G maps to at most $2^{n/2}$ y-values. Therefore, the image of G is a negligible fraction of $\{0,1\}^n$, the set of possible y-values. More formally:

$$\Pr_{y \leftarrow \{0,1\}^n}[y \in \mathsf{Im}(G)] = \frac{|\mathsf{Im}(G)|}{|\{0,1\}^n|} \leq \frac{2^{n/2}}{2^n} = 2^{-n/2}$$

   In the truly random case, $y \leftarrow \{0,1\}^n$, so with probability $\geq 1 - 2^{-n/2}$, $y \notin \mathsf{Im}(G)$. If this occurs, then $\mathcal{B}$ is certain to output 0 because $\mathcal{A}$ will not be able to find an x' such that $G(x') = y$.

Therefore,

$$\Pr_{y \leftarrow \{0,1\}^n}[\mathcal{B}(y) \to 1] \leq 2^{-n/2}$$

5. In conclusion,

$$\Pr_{x \leftarrow \{0,1\}^{n/2}}[\mathcal{B}(G(x)) \to 1] - \Pr_{y \leftarrow \{0,1\}^n}[\mathcal{B}(y) \to 1] \geq \mathsf{non\text{-}negl}(n) - 2^{-n/2}$$

which is non-negligible. Therefore $\mathcal{B}$ breaks the PRG security of G.

2 A Candidate Key-Exchange Protocol (5 points)

Consider the following proposal for a key exchange protocol:

1. Alice samples $k \leftarrow \{0,1\}^n$ and $r \leftarrow \{0,1\}^n$, and sends Bob the following:
$$s := k \oplus r$$
2. Bob samples $t \leftarrow \{0,1\}^n$, and sends Alice the following:
$$u := s \oplus t$$
3. Alice sends Bob the following:
$$w := u \oplus r$$
4. Alice outputs k and Bob outputs $w \oplus t$.

Questions:

(a) *Correctness:* Show that Alice and Bob will always output the same key k.

(b) *Security:* Does this scheme satisfy key-exchange security[1]? Prove your answer.

Solution

1. Correctness: Alice outputs k and Bob outputs:
$$w \oplus t = (u \oplus r) \oplus t = ((s \oplus t) \oplus r) \oplus t = (((k \oplus r) \oplus t) \oplus r) \oplus t = k.$$
2. Security: This scheme does not satisfy key-exchange security. We will construct an adversary $\mathcal{A}$ that breaks security:

   Construction of $\mathcal{A}$:

   (a) The key-exchange challenger runs the key-exchange protocol, and gives $\mathcal{A}$ a transcript of the messages sent during the protocol:
   $$\mathsf{transcript} = (s, u, w)$$
   They also sample a bit $b \leftarrow \{0,1\}$. If $b = 0$, they give $\mathcal{A}$ $\hat{k} := k$. If $b = 1$, they give $\mathcal{A}$ $\hat{k} \leftarrow \{0,1\}^n$.

   (b) $\mathcal{A}$ computes $k' = s \oplus u \oplus w$. If $k' = \hat{k}$, then $\mathcal{A}$ outputs $b' = 0$. Otherwise, $\mathcal{A}$ outputs $b' = 1$.

[1] Key-exchange security is also known as $\mathsf{KE}^{\mathsf{eav}}_{\mathcal{A},\Pi}$ security. It was defined in lecture 13, slide 26, as well as in Katz & Lindell, 3rd Edition, Definition 11.1.

3. We claim that $k' = k$:

$$\begin{aligned} k' &= s \oplus u \oplus w = s \oplus u \oplus (u \oplus r) \\ &= s \oplus r = (k \oplus r) \oplus r \\ &= k \end{aligned}$$

Once $\mathcal{A}$ obtains k, they can easily determine whether $\hat{k} = k$, so they win the key-exchange security game with overwhelming probability. This breaks the security of the candidate key-exchange protocol.