CS 388H: Introduction to Cryptography

Attacks and Reductions in Cryptography

Instructor: David Wu

In this short note, we give several examples of proofs involving PRGs and PRFs.

PRG security. Let's begin by reviewing the PRG security game:

The PRG security game is played between an adversary $\mathcal{A}$ and a challenger. Let $G\colon \{0,1\}^\lambda \to \{0,1\}^n$ be a candidate PRG. The game is parameterized by a bit $b \in \{0,1\}$:

1. If $b = 0$, the challenger samples a seed $s \xleftarrow{\text{R}} \{0,1\}^\lambda$ and computes $t \leftarrow G(s)$. If $b = 1$, the challenger samples a random string $t \xleftarrow{\text{R}} \{0,1\}^n$.
2. The challenger gives t to $\mathcal{A}$.
3. At the end of the game, $\mathcal{A}$ outputs a bit $b' \in \{0,1\}$.

For an adversary $\mathcal{A}$, we define its PRG distinguishing advantage $\mathsf{PRGAdv}[\mathcal{A}, G]$ to be the quantity

$$\mathsf{PRGAdv}[\mathcal{A}, G] = \left|\Pr[b' = 1 \mid b = 0] - \Pr[b' = 1 \mid b = 1]\right|.$$

Finally, we say that a a PRG G is secure if for all efficient adversaries $\mathcal{A}$,

$$\mathsf{PRGAdv}[\mathcal{A}, G] = \mathrm{negl}(\lambda).$$

We will often refer to this game (also called an "experiment") where $b = 0$ as $\mathsf{PRGExp}_0[\mathcal{A}, G]$ and the game where $b = 1$ as $\mathsf{PRGExp}_1[\mathcal{A}, G]$. In this case, we can also write

$$\mathsf{PRGAdv}[\mathcal{A}, G] = \left|\Pr\left[\mathcal{A} \text{ outputs 1 in } \mathsf{PRGExp}_0[\mathcal{A}, G]\right] - \Pr\left[\mathcal{A} \text{ outputs 1 in } \mathsf{PRGExp}_1[\mathcal{A}, G]\right]\right|.$$

Example 1 (An Insecure PRG)**.** *Suppose $G\colon \{0,1\}^\lambda \to \{0,1\}^n$ is a secure PRG and define $G'\colon \{0,1\}^\lambda \to \{0,1\}^{n+\lambda}$ to be $G'(s) = G(s) \| s$. We show that G' is not a secure PRG.*

Proof. We construct an adversary $\mathcal{A}$ for G' as follows:

1. On input $t \in \{0,1\}^{n+\lambda}$, $\mathcal{A}$ parses the input as $t = t_1 \| t_2$ where $t_1 \in \{0,1\}^n$ and $t_2 \in \{0,1\}^\lambda$.
2. Output 1 if $G(t_2) = t_1$ and 0 otherwise.

By construction, algorithm $\mathcal{A}$ is efficient (i.e., runs in polynomial time). We compute $\mathcal{A}$'s distinguishing advantage:

- Suppose $b = 0$. In this case, $t \leftarrow G'(s)$ where $s \xleftarrow{\text{R}} \{0,1\}^\lambda$. By construction of G', $t = t_1 \| t_2$ where $G(t_2) = t_1$. In this case, the adversary outputs 1 with probability 1.

- Suppose $b = 1$. In this case, $t \xleftarrow{\text{R}} \{0,1\}^{n+\lambda}$. In particular, t_1 and t_2 are independently uniform, so $\Pr[t_1 = G'(t_2)] = 1/2^n$.

The distinguishing advantage of $\mathcal{A}$ is then

$$\mathsf{PRGAdv}[\mathcal{A}, G'] = \left|\Pr[b' = 1 \mid b = 0] - \Pr[b' = 1 \mid b = 1]\right| = 1 - 2^{-n},$$

which is non-negligible. $\square$

Example 2 (A Secure PRG)**.** *Suppose $G\colon \{0,1\}^\lambda \to \{0,1\}^n$ is a secure PRG and define the function $G'\colon \{0,1\}^\lambda \to \{0,1\}^n$ to be the function $G'(s) = G(s) \oplus 1^n$. Namely, G' simply flips the output bits of G. We show that if G is secure, then G' is also secure.*

Proof. When proving statements of this form, we will prove the contrapositive:

> *If G' is not a secure PRG, then G is not a secure PRG.*

To prove the contrapositive, we begin by assuming that G' is not a secure PRG. This means that there exists an efficient adversary $\mathcal{A}$ that breaks the security of G' with non-negligible advantage ε (i.e., $\mathsf{PRGAdv}[\mathcal{A}, G'] = \varepsilon$). We use $\mathcal{A}$ to construct an efficient adversary $\mathcal{B}$ that breaks the security of G:[1]

1. At the beginning of the game, algorithm $\mathcal{B}$ receives a challenge $t \xleftarrow{R} \{0,1\}^n$ from the challenger. We are constructing an adversary for the PRG security game for G. This game begins with the challenger sending a challenge $t \in \{0,1\}^n$ to the adversary where either $t \leftarrow G(s)$ or $t \xleftarrow{R} \{0,1\}^n$.
2. Algorithm $\mathcal{B}$ starts running algorithm $\mathcal{A}$. Essentially, we are constructing a reduction here. Our goal is to reduce the problem of distinguishing G to the problem of distinguishing G'. To do this, we will rely on our adversary $\mathcal{A}$ for distinguishing G'.
3. Algorithm $\mathcal{B}$ sends $t \oplus 1^n$ to $\mathcal{A}$ and outputs whatever $\mathcal{A}$ outputs. Algorithm $\mathcal{A}$ is an adversary for G', so it expects a single input $t \in \{0,1\}^n$ where either $t \leftarrow G'(s)$ or $t \xleftarrow{R} \{0,1\}^n$. Note that this is the only setting for which we have guarantees on the behavior of $\mathcal{A}$. The behavior of algorithm $\mathcal{A}$ on a string drawn from some other distribution is *undefined*. As part of our analysis, we need to argue that $\mathcal{B}$ correctly *simulates* the view of $\mathcal{A}$ in the PRG distinguishing game against G'.

First, if $\mathcal{A}$ is efficient, then $\mathcal{B}$ is also efficient (by construction). It suffices to compute the distinguishing advantage of algorithm $\mathcal{B}$. We consider two cases:

- Suppose $b = 0$. Then, $\mathcal{B}$ receives a string $t \leftarrow G(s)$ where $s \xleftarrow{R} \{0,1\}^\lambda$. In this case, $t \oplus 1^n$ is precisely the value of $G'(s)$. Namely, $\mathcal{B}$ has simulated $\mathsf{PRGExp}_0[\mathcal{A}, G']$ for $\mathcal{A}$. Since $\mathcal{A}$ is a distinguisher for G', this means that

$$\Pr\left[\mathcal{B} \text{ outputs } 1 \mid b = 0\right] = \Pr\left[\mathcal{A} \text{ outputs } 1 \text{ in } \mathsf{PRGExp}_0[\mathcal{A}, G']\right].$$

- Suppose $b = 1$. Then, $\mathcal{B}$ receives a random string $t \xleftarrow{R} \{0,1\}^n$. Since t is uniformly random over $\{0,1\}^n$, the string $t \oplus 1^n$ is also uniformly random over $\{0,1\}^n$. This means that $\mathcal{B}$ has simulated $\mathsf{PRGExp}_1[\mathcal{A}, G']$ for $\mathcal{A}$. This means that

$$\Pr\left[\mathcal{B} \text{ outputs } 1 \mid b = 1\right] = \Pr\left[\mathcal{A} \text{ outputs } 1 \text{ in } \mathsf{PRGExp}_1[\mathcal{A}, G']\right].$$

[1] In the following description, we provide some clarifying remarks in green. These remarks are unnecessary in a formal proof.

We conclude now that the distinguishing advantage of $\mathcal{B}$ is exactly

$$\begin{aligned}\mathsf{PRGAdv}[\mathcal{B},G] &= \left|\Pr\left[\mathcal{B}\text{ outputs }1 \mid b=0\right] - \Pr\left[\mathcal{B}\text{ outputs }1 \mid b=1\right]\right| \\ &= \left|\Pr\left[\mathcal{A}\text{ outputs 1 in }\mathsf{PRGExp}_0[\mathcal{A},G']\right] - \Pr\left[\mathcal{A}\text{ outputs 1 in }\mathsf{PRGExp}_1[\mathcal{A},G']\right]\right| \\ &= \mathsf{PRGAdv}[\mathcal{A},G'] = \varepsilon,\end{aligned}$$

which is non-negligible by assumption. □

PRF security game. Next, we review the definition of a secure PRF. Let $F\colon \mathcal{K}\times\mathcal{X}\to\mathcal{Y}$ be a function with key-space $\mathcal{K}$, domain $\mathcal{X}$, and range $\mathcal{Y}$. The PRF security game is defined as follows:

The PRF security game is played between an adversary $\mathcal{A}$ and a challenger. Let $F\colon \mathcal{K}\times\mathcal{X}\to\mathcal{Y}$ be a candidate PRF. The game is parameterized by a bit $b\in\{0,1\}$:

1. If $b=0$, then the challenger samples a key $k \xleftarrow{\text{R}} \mathcal{K}$ and sets $f \leftarrow F(k,\cdot)$. If $b=1$, the challenger samples a uniformly random function $f \xleftarrow{\text{R}} \mathrm{Funs}[\mathcal{X},\mathcal{Y}]$.
2. The adversary chooses $x\in\mathcal{X}$ and sends x to the challenger.
3. The challenger replies with $f(x)$.
4. The adversary can continue to make queries to the adversary (repeating steps 2 and 3). At the end of the game, adversary outputs a bit $b'\in\{0,1\}$.

For an adversary $\mathcal{A}$, we define the PRF distinguishing advantage $\mathsf{PRFAdv}[\mathcal{A},F]$ to be the quantity

$$\mathsf{PRFAdv}[\mathcal{A},F] = \left|\Pr[b'=1 \mid b=0] - \Pr[b'=1 \mid b=1]\right|.$$

We say that a PRF F is secure if for all efficient adversaries $\mathcal{A}$,

$$\mathsf{PRFAdv}[\mathcal{A},F] = \mathrm{negl}(\lambda),$$

where λ is a security parameter (typically, the keys of the PRF are $\mathrm{poly}(\lambda)$ bits long: $\log|\mathcal{K}| = \mathrm{poly}(\lambda)$). Similar to the case with PRGs, we will often refer to the game (or "experiment") where $b=0$ as $\mathsf{PRFExp}_0[\mathcal{A},F]$ and the game where $b=1$ as $\mathsf{PRFExp}_1[\mathcal{A},F]$. In this case, we can write

$$\mathsf{PRFAdv}[\mathcal{A},F] = \left|\Pr\left[\mathcal{A}\text{ outputs 1 in }\mathsf{PRFExp}_0[\mathcal{A},F]\right] - \Pr\left[\mathcal{A}\text{ outputs 1 in }\mathsf{PRFExp}_1[\mathcal{A},F]\right]\right|.$$

Example 3 (An Insecure PRF)**.** *Suppose $F\colon\{0,1\}^n\times\{0,1\}^n\to\{0,1\}^n$ is a secure PRF and define $F'\colon\{0,1\}^n\times\{0,1\}^n\to\{0,1\}^n$ to be $F'(k,x)=F(k,x)\oplus F(k,x\oplus 1^n)$. We claim that F' is not a secure PRF.*

Proof. We construct an adversary $\mathcal{A}$ for F' as follows:

1. Submit the query $x_1 = 0^n$ to the challenger. The challenger replies with a value y_1.
2. Submit the query $x_2 = 1^n$ to the challenger. The challenger replies with a value y_2.
3. Output 1 if $y_1 = y_2$ and 0 otherwise.

By construction, $\mathcal{A}$ is efficient (i.e., runs in polynomial time). We compute $\mathcal{A}$'s distinguishing advantage:

- Suppose $b = 0$. In this case, the challenger samples $k \xleftarrow{\text{R}} \{0,1\}^n$ and replies with

$$y_1 = F'(k, x_1) = F(k, x_1) \oplus F(k, x_1 \oplus 1^n) = F(k, 0^n) \oplus F(k, 1^n)$$
$$y_2 = F'(k, x_2) = F(k, x_2) \oplus F(k, x_2 \oplus 1^n) = F(k, 1^n) \oplus F(k, 0^n).$$

  In this case $y_1 = y_2$, and $\mathcal{A}$ outputs 1 with probability 1.

- Suppose $b = 1$. In this case, the challenger samples $f \xleftarrow{\text{R}} \mathsf{Funs}[\{0,1\}^n, \{0,1\}^n]$ and replies with $y_1 = f(x_1)$ and $y_2 = f(x_2)$. Since $x_1 \neq x_2$, y_1 and y_2 are independent and uniformly random. Thus, $\Pr[y_1 = y_2] = 1/2^n$.

The distinguishing advantage of $\mathcal{A}$ is then

$$\mathsf{PRFAdv}[\mathcal{A}, F'] = \left|\Pr[b' = 1 \mid b = 0] - \Pr[b' = 1 \mid b = 1]\right| = 1 - 2^{-n},$$

which is non-negligible. □

Example 4 (A Secure PRF)**.** *Suppose $F \colon \mathcal{K} \times \mathcal{X} \to \{0,1\}^n$ is a secure PRF. Then, the function $F' \colon \mathcal{K}^2 \times \mathcal{X} \to \{0,1\}^n$ where $F'((k_1, k_2), x) = F(k_1, x) \oplus F(k_2, x)$ is also a secure PRF.*

Proof. Similar to the case with PRGs, we will prove the contrapositive:

If F' is not a secure PRF, then F is not a secure PRF.

To prove the contrapositive, we begin by assuming that F' is not a secure PRF. This means that there exists an efficient adversary $\mathcal{A}$ that breaks the security of F' with non-negligible advantage ε (i.e., $\mathsf{PRFAdv}[\mathcal{A}, F'] = \varepsilon$). We use $\mathcal{A}$ to construct an adversary $\mathcal{B}$ that breaks the security of F:

1. Choose a key $k_2 \xleftarrow{\text{R}} \mathcal{K}$.
2. Start running the adversary $\mathcal{A}$ for F'.
   (a) Whenever $\mathcal{A}$ makes a query $x_i \in \mathcal{X}$, forward the query to the challenger to obtain a value $y_i \in \{0,1\}^n$. Give $y_i \oplus F(k_2, x_i)$ to $\mathcal{A}$.
3. Output whatever $\mathcal{A}$ outputs.

Observe that the number of queries $\mathcal{B}$ makes is the same as the number of queries that $\mathcal{A}$ makes. Thus, if $\mathcal{A}$ is efficient, then $\mathcal{B}$ is also efficient. It suffices to compute the distinguishing advantage of algorithm $\mathcal{B}$. We consider two cases:

- Suppose $b = 0$. In this case, the challenger in $\mathsf{PRFExp}_0[\mathcal{B}, F]$ samples a key $k \xleftarrow{\text{R}} \mathcal{K}$ and replies with $y_i \leftarrow F(k, x_i)$ on each query. Algorithm $\mathcal{B}$ in turns replies to $\mathcal{A}$ with the value

$$y_i \oplus F(k_2, x_i) = F(k, x_i) \oplus F(k_2, x_i) = F'((k, k_2), x_i).$$

  Since k and k_2 are both sampled uniformly and independently from $\mathcal{K}$, algorithm $\mathcal{B}$ answers all of $\mathcal{A}$'s queries according to the specification of $\mathsf{PRFExp}_0[\mathcal{A}, F']$. Thus,

$$\Pr\left[\mathcal{B} \text{ outputs } 1 \mid b = 0\right] = \Pr\left[\mathcal{A} \text{ outputs } 1 \text{ in } \mathsf{PRFExp}_0[\mathcal{A}, F']\right].$$

- Suppose $b = 1$. In this case, the challenger in $\mathsf{PRFExp}_1[\mathcal{B}, F]$ samples $f \xleftarrow{\text{R}} \mathsf{Funs}[\mathcal{X}, \{0,1\}^n]$ and replies with $y_i \leftarrow f(x_i)$ on each query. Algorithm $\mathcal{B}$ in turn replies to $\mathcal{A}$ with the value $y_i \oplus F(k_2, x_i) = f(x_i) \oplus F(k_2, x_i)$. Since k_2 is independent of f, and f is a random function, the value of $f(x_i) \oplus F(k_2, x_i)$ is uniform and independently random over $\{0,1\}^n$. Thus, algorithm $\mathcal{B}$ answers all of $\mathcal{A}$'s queries according to the specification of $\mathsf{PRFExp}_1[\mathcal{A}, F']$, and so

$$\Pr\left[\mathcal{B} \text{ outputs } 1 \mid b = 1\right] = \Pr\left[\mathcal{A} \text{ outputs } 1 \text{ in } \mathsf{PRFExp}_1[\mathcal{A}, F']\right].$$

By definition, the distinguishing advantage of $\mathcal{B}$ is then

$$\mathsf{PRFAdv}[\mathcal{B}, F] = \left|\Pr\left[\mathcal{B} \text{ outputs } 1 \mid b = 0\right] - \Pr\left[\mathcal{B} \text{ outputs } 1 \mid b = 1\right]\right| = \mathsf{PRFAdv}[\mathcal{A}, F'] = \varepsilon,$$

which is non-negligible by assumption. $\square$