CS276 Scribe Notes

Lenny Grokop <lgrokop@eecs.berkeley.edu>

February 2, 2004 (part 1)

1 Pseudo-Random Functions

Imagine an input/output black box in one of two states:

(0) outputs are generated by a truly random function f

(1) outputs are generated by a pseudo-random function F_K, with seed K generated randomly

An algorithm A is run on the box to determine which state it is in. A has oracle access to F_K, enabling it to obtain input-output examples of its choice. It queries the box by inputting a sequence of strings and noting the output. The input may be random, and may also be chosen adaptively based on the previous output. Ultimately A outputs an estimate of the state, either a 0 or a 1.

Let Keys denote the set of all keys (typically the set $\{0,1\}^k$).

A *pseudo-random function* is a mapping $F : \mathsf{Keys} \times \{0,1\}^n \to \{0,1\}^m$, i.e. it takes as input a key $K \in \mathsf{Keys}$ and a message string $x \in \{0,1\}^n$ and produces an output string $F_K(x) \in \{0,1\}^m$ representing an encrypted version of the message.

Let $\mathsf{Function}$ denote the set of all $f : \{0,1\}^n \to \{0,1\}^m$.

Definition 1.1. *Algorithm A's advantage at distinguishing a random function $f \in \mathsf{Function}$ from a pseudo-random function F_K generated by a random key $K \in \mathsf{Keys}$, is defined as*

$$\mathsf{Adv}(A) \triangleq |\Pr_{K \leftarrow \mathsf{Keys}}[A^{F_K} = 1] - \Pr_{f \leftarrow \mathsf{Function}}[A^f = 1]|. \tag{1}$$

Definition 1.2. *F is said to be (t, q, ϵ) secure as a PRF if $\mathsf{Adv}(A) \leq \epsilon, \forall$ algorithms A running in time $\leq t$ with $\leq q$ queries.*

An example of a PRF is AES. For AES, $n = m = 128$. Each input-ouput mapping has a 2^{128} bit domain and 2^{128} bit range. Thus the total number of functions possible is $|\mathsf{Function}| = (2^{128})^{2^{128}}$.

2 Applications of PRF's

The following claim provides the link between PRF's and key recovery.

Claim 2.1. *If F is a (t, q, ϵ)-PRF with $F : \mathsf{Keys} \times \{0,1\}^{\mathsf{n}} \to \{0,1\}^{\mathsf{m}}$ then*

$$\Pr_{K \in \mathsf{Keys}}[A^{F_K} = K] \leq \epsilon + 2^{-m} \tag{2}$$

for all A with running time $\leq t$ and number of queries $\leq q - 1$.

Proof. Proof by contraposition. We use a reduction argument. If there exists an A that can choose the key correctly with probability $> \epsilon + 2^{-m}$ then we can use it in an algorithm B (with running in time $\leq t$ and queries $\leq q$) to gain an advantage greater than ϵ against F. Here are the details.

Let A satisfy $\Pr_{K \in \mathsf{Keys}}[A^{F_K} = K] > \epsilon + 2^{-m}$, with running time $\leq t - O(q)$ and number of queries $\leq q - 1$.

Define B like so:

1. Simulate A^g. Denote the output K'. Also let S = set of queries A makes.
2. Let x be in $\{0,1\}^n - S$.
3. If $g(x) = F_{K'}(x)$, output 1, else 0.

In words, the algorithm B runs algorithm A on the black box. A makes some queries and then takes a guess at the key. B takes note of A's queries, then chooses a different query and runs that through the box. B computes for itself the expected output based on A's guess at the key. If the box output agrees with B's prediction, B outputs a 1. If the box outputs something else, B's outputs a 0.

Now the analysis. We compute B's advantage.

We assume without loss of generality $\Pr_{K \leftarrow \mathsf{Keys}}[B^{F_K} = 1] \geq \Pr_{f \leftarrow \mathsf{Function}}[B^f = 1]$.

Then

$$\begin{aligned} \mathsf{Adv}(B) &= |\Pr_{K \leftarrow \mathsf{Keys}}[B^{F_K} = 1] - \Pr_{f \leftarrow \mathsf{Function}}[B^f = 1]| \\ &= \Pr_{K \leftarrow \mathsf{Keys}}[B^{F_K} = 1] - \Pr_{f \leftarrow \mathsf{Function}}[B^f = 1] \\ &\geq \Pr_{K \leftarrow \mathsf{Keys}}[A^{F_K} = 1] - \Pr_{f \leftarrow \mathsf{Function}}[B^f = 1], \end{aligned}$$

as B succeeds *at least* whenever A guesses correctly. The converse is "almost true" in that the probability B outputs a 1 when the box output is generated by a truly random function is 2^{-m}. Thus

$$\begin{aligned} \mathsf{Adv}(B) &\geq \Pr_{K \leftarrow \mathsf{Keys}}[A^{F_K} = 1] - 2^{-m} \\ &> \epsilon + 2^{-m} - 2^{-m} \\ &= \epsilon. \end{aligned}$$

The running time of B is equal to the running time of A plus the time to make 1 additional query, $O(q)$ (and a small computation which we ignore). So we have described an algorithm running in time $t - O(q) + O(q) = t$ making $q - 1 + 1 = q$ queries with advantage ϵ against F. This proves our claim. □

Moral of the story: An ϵ secure PRF is also approximately ϵ secure against key recovery.