CMSC 28400 Introduction to Cryptography **Autumn 2021**

Notes #7: Pseudorandom Permutations

Instructor: David Cash

In these notes we introduce and briefly study the concept of a *pseudorandom permutation* (PRP). This formalizes a notions of security that one can reasonably believe block ciphers like AES achieve. Our plan is to use the PRP notion in the next set of notes to guide the design of randomized encryption schemes.

For a preview and some motivation, we plan to analyze the following simple randomized encryption scheme Π, which is a basic version of real commonly-used schemes. The relevant sets are $\mathcal{K} = \mathcal{M} = \mathcal{R} = \{0,1\}^{128}$,and $\mathcal{C} = \{0,1\}^{128} \times \{0,1\}^{128}$. Encryption is defined by:

$$\mathsf{Enc}(k, m, r) = (r, \mathsf{AES}(k, r) \oplus m).$$

Intuitively, this encryption scheme is picking r at random, and sending it "in the clear" in the ciphertext. The "pad" is $\mathsf{AES}(k, r)$. (Decryption recomputes this pad and XORs it off.) The idea is that the sender and receiver both know k and can easily compute the pad. But an attacker will know r (from the ciphertext) but not k, and thus cannot evaluate $\mathsf{AES}(k, r)$. Based on the design of AES and the lack of attacks, it is generally believed the pad in this scenario will "look random", and intuitively we'll get security analogously to how we did with the pseudo-OTP. Even better, since r is random each time, we can get *many-time* security if all of several such pads look random. The concepts we outline next will give us a foundation for rigorously analyzing this situation, similar to how we used PRGs for the pseudo-OTP.

7.1 Pseudorandom Permutations

We now give an abstraction describing a main security goal of block ciphers. The following definitions use the concept of a *random permutation* $\boldsymbol{\pi} : \{0,1\}^\ell \to \{0,1\}^\ell$. Formally, this makes perfect sense: the set of all possible permutations on the set $\{0,1\}^\ell$, and we mean to select one uniformly at random. You can think of such a random variable as a table listing an output for every possible input, and such that the outputs are a uniformly random list of every possible output. For a concrete example with $\ell = 3$, we might have $\boldsymbol{\pi}$ represented by the table:

x	$f(x)$
000	100
001	011
010	010
011	101
100	011
100	111
101	001
110	000
111	110

So "picking a random permutation" means filling in the right side of the table with the elements of $\{0,1\}^3$ in random order.

Example 7.1. *Let $\boldsymbol{\pi}$ be a random function from $\{0,1\}^{128}$ to $\{0,1\}^{128}$. Then the following hold:*

$$\Pr[\boldsymbol{\pi}(0^{128}) = 0^{128}] = \frac{1}{2^{128}},$$
$$\Pr[\boldsymbol{\pi}(0^{128}) \textit{ starts with } 0] = \frac{1}{2},$$
$$\Pr[\boldsymbol{\pi}(0^{128}) \oplus \mathbf{f}(1^{128}) = 1^{128}] = \frac{1}{2^{128}-1}.$$

Things are not always so simple. For instance,

$$\Pr[\boldsymbol{\pi}(\boldsymbol{\pi}(0^{128})) = 0^{128}] = \frac{1}{2^{128}} + (1 - \frac{1}{2^{128}})\frac{1}{2^{128}-1},$$

which can be seen by applying the law of total probability to split up the probability into the cases where $f(0^{128}) = 0^{128}$ and $f(0^{128}) \neq 0^{128}$.

We can also define a *random permutation* $\boldsymbol{\pi} : \{0,1\}^\ell \to \{0,1\}^\ell$ similarly: It's just a randomly chosen sample from the set of all possible permutations on $\{0,1\}^\ell$. We can sample $\boldsymbol{\pi}$ using the table view above, except now we must ensure that the values on the right column are all distinct.

7.1.1 Psuedorandom Permutation Definition

We want to formalize a statement like "$\mathsf{AES}(\mathbf{K},\cdot)$ looks like a random permutation when $\mathbf{K}$ is a random key", and then use such an assumption to analyze encryption schemes. For PRGs we were able to do this by asking that no distinguisher could effectively tell the PRG output from random. But how do we do this for a block cipher? A quick attempt might be to give a distinguisher the table for either $\mathsf{AES}(\mathbf{K},\cdot)$, or the table for a random permutation, and ask it to distinguish these cases. But this definition isn't very useful; The tables are absolutely gigantic, well beyond what is possible to write down using all of the atoms in the universe (we'd need $\log_2 |\mathrm{Perm}(\{0,1\}^\ell)|$ bits)! So while that definition may make sense mathematically, the distinguishers involved wouldn't correspond to real attacks.

The definition we give next resolves this problem by only giving the adversary *oracle access* to either $\mathsf{AES}(\mathbf{K},\cdot)$ or a random permutation. That is, the adversary gets input/output access to an oracle computing one those functions, and attempts to determine which one it is talking to. It can make as many queries as it wants, with whatever inputs it can come up with. The only bound comes from the runtime of the adversary.

Definition 7.1. *Let $E : \{0,1\}^n \times \{0,1\}^\ell \to \{0,1\}^\ell$ be a block cipher and $\mathcal{A}$ be an adversary. Let $\mathbf{K}$ be uniform on $\{0,1\}^n$, and let $\boldsymbol{\pi}$ be a random permutation on $\{0,1\}^\ell$. We define the* pseudorandom permutation (PRP) distinguishing advantage of $\mathcal{A}$ against E *to be*

$$\mathbf{Adv}^{\mathrm{prp}}_E(\mathcal{A}) = \left|\Pr[\mathcal{A}^{E(\mathbf{K},\cdot)} = 1] - \Pr[\mathcal{A}^{\boldsymbol{\pi}(\cdot)} = 1]\right|.$$

One way to interpret this definition is to look at E as a "family of permutations," indexed by the key. Then $E(\mathbf{K},\cdot)$ is a random member of this family, while $\boldsymbol{\pi}(\cdot)$ is a random permutation

from amongst all possibilities. Practical block ciphers like AES are designed to look like random permutations in this manner, resisting all efficient attempts to find patterns in their outputs.

An adversary $\mathcal{A}$ can be judged according to its runtime, the number of queries it issues, and its advantage. We won't distinguish much between runtime and queries, but in practice an adversary will usually have far more computation time than it does queries, because the queries need to be run by the parties under attack. They won't be enabling anything like 2^{80} queries, but the adversary might have that much computation. If the advantage of every reasonable $\mathcal{A}$ is small (say 2^{-64} or 2^{-128}, depending on the application), we informally say that E is a good PRP.

In that case we think of E with a random key as "looking like a random permutation," and all of the properties of random permutations are thus inherited by E. So $E(\mathbf{K}, x)$ and $E(\mathbf{K}, x')$ ($x \neq x'$) should look like two random distinct strings, even when x and x' differ only by one bit. If not, then there would be an adversary with good PRP advantage.

Example 7.2. *Suppose* $E : \{0,1\}^{128} \times \{0,1\}^{128} \to \{0,1\}^{128}$ *satisfies* $E(k, 0^{128}) = 0^{128}$ *for every* $k \in \{0,1\}^{128}$. *We show that* E *is not a good PRP. Consider* $\mathcal{A}^{\mathcal{O}}$ *that queries* 0^{128} *to its oracle* $\mathcal{O}$ *and calls the response* y. *If* $y = 0^{128}$ *then* $\mathcal{A}$ *outputs* 1, *else* 0. *Then*

$$\Pr[\mathcal{A}^{E(\mathbf{K},\cdot)} = 1] = 1$$

and

$$\Pr[\mathcal{A}^{\boldsymbol{\pi}(\cdot)} = 1] = \frac{1}{2^{128}},$$

the latter because a random permutation satisfies $\mathbf{f}(0^{128}) = 0^{128}$ *with probability* $\frac{1}{2^{128}}$. *Thus*

$$\mathbf{Adv}^{\mathrm{prp}}_{E}(\mathcal{A}) = 1 - \frac{1}{2^{128}},$$

which is high. Moreover, $\mathcal{A}$ *only issues one query and performs some trivial computation.*

Note that we had to define $\mathcal{A}$ with respect to a generic oracle $\mathcal{O}$; We don't get to say how $\mathcal{A}$ works with $E(\mathbf{K}, \cdot)$ and $\boldsymbol{\pi}(\cdot)$ separately, since $\mathcal{A}$ only gets to see their input/output behavior. More complicated patterns can be found with more clever attacks.

Exercise 7.1. *Suppose* $E : \{0,1\}^{128} \times \{0,1\}^{128} \to \{0,1\}^{128}$ *satisfies* $E(k, 0^{128}) = \overline{E(k, 1^{128})}$ *(the bar indicates bitwise complement) for every* $k \in \{0,1\}^{128}$. *Similar to the previous example, show that* E *cannot be a good PRP. What is the advantage of your adversary?*