Random Permutations

We can sample a random *permutation* by choosing the values in the right column uniformly and independently, *without replacement*:

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

Counting Permutations

Question

How many permutations are there mapping $\{0,1\}^n \to \{0,1\}^n$?

Counting Permutations

Question

How many permutations are there mapping $\{0,1\}^n \to \{0,1\}^n$?

$(2^n)!$

Pseudo-random Permutations (PRPs)

We'd like to use randomly chosen permutations, but this requires exponential space!

Instead, we will use pseudo-random permutations: keyed permutations that are indistinguishable from random:
$F : \{0,1\}^* \times \{0,1\}^* \rightarrow \{0,1\}^*$
This is a 2-input function, where 1st input is the key.

The sec. param. determines the key length, the input length, and the output length.
However, the output length and the input length are now the same.
Technically, $\ell_{\text{key}}(n), \ell_{\text{in}}(n)$ and $\ell_{\text{out}}(n)$.
F is called a keyed permutation if it is one-to-one

Pseudo-random Permutations (PRPs)

We'd like to use randomly chosen permutations, but this requires exponential space!

Instead, we will use pseudo-random permutations: keyed permutations that are indistinguishable from random:
$F : \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^*$
This is a 2-input function, where 1st input is the key.

The sec. param. determines the key length, the input length, and the output length.
However, the output length and the input length are now the same.
Technically, $\ell_{\text{key}}(n), \ell_{\text{in}}(n)$ and $\ell_{\text{out}}(n)$.
F is called a keyed permutation if it is one-to-one
F is called efficient if there are polynomial time algorithms for evaluating

- F, given key k and input x,
- F^{-1} given key k and output y.

Pseudo-random Permutations (PRPs)

We'd like to use randomly chosen permutations, but this requires exponential space!

Instead, we will use pseudo-random permutations: keyed permutations that are indistinguishable from random:
$F : \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^*$
This is a 2-input function, where 1st input is the key.

The sec. param. determines the key length, the input length, and the output length.
However, the output length and the input length are now the same.
Technically, $\ell_{\text{key}}(n), \ell_{\text{in}}(n)$ and $\ell_{\text{out}}(n)$.
F is called a keyed permutation if it is one-to-one
F is called efficient if there are polynomial time algorithms for evaluating

- F, given key k and input x,
- F^{-1} given key k and output y.

We call $\ell_{\text{in}}(n)$ the block-length of the PRP.

Pseudo-random Permutations (PRPs)

We'd like to use randomly chosen permutations, but this requires exponential space!

Instead, we will use pseudo-random permutations: keyed permutations that are indistinguishable from random:
$F : \{0,1\}^* \times \{0,1\}^* \rightarrow \{0,1\}^*$
This is a 2-input function, where 1st input is the key.

The sec. param. determines the key length, the input length, and the output length.
However, the output length and the input length are now the same.
Technically, $\ell_{\text{key}}(n), \ell_{\text{in}}(n)$ and $\ell_{\text{out}}(n)$.
F is called a keyed permutation if it is one-to-one
F is called efficient if there are polynomial time algorithms for evaluating

- F, given key k and input x,
- F^{-1} given key k and output y.

We call $\ell_{\text{in}}(n)$ the block-length of the PRP.

Often, we will assume that F is length preserving:
$\ell_{\text{key}}(n) = \ell_{\text{in}}(n) = n$

Often, we will want to fix a single key k and then evaluate F on many different inputs, using the same k. In that case, we might write $F_k : \{0,1\}^* \rightarrow \{0,1\}^*$.
If it is length preserving, and the key is of length n, then $F_k : \{0,1\}^n \rightarrow \{0,1\}^n$.

Security of PRPs

Definition

Let $F : \{0,1\}^* \times \{0,1\}^* \rightarrow \{0,1\}^*$ be an efficient, length-preserving, keyed permutation. F is a *pseudorandom permutation* if $\forall$ p.p.t. adversaries $\mathcal{A}$, there is a negligible function negl(n) such that $\Pr[\mathsf{PrivK}^{\mathsf{prp}}_{\mathcal{A},\mathsf{F}}(n) = 1] \leq \frac{1}{2} + \mathsf{negl(n)}$.

Security of PRPs

Definition

Let $F : \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^*$ be an efficient, length-preserving, keyed permutation. F is a *pseudorandom permutation* if $\forall$ p.p.t. adversaries $\mathcal{A}$, there is a negligible function negl(n) such that $\Pr[\mathsf{PrivK}^{\mathsf{prp}}_{\mathcal{A},\mathsf{F}}(n) = 1] \leq \frac{1}{2} + \mathsf{negl(n)}$.

Definition

Let $F : \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^*$ be an efficient, length-preserving, keyed permutation. F is a *strong pseudorandom permutation* if $\forall$ p.p.t. adversaries $\mathcal{A}$, there is a negligible function negl(n) such that $\Pr[\mathsf{PrivK}^{\mathsf{sprp}}_{\mathcal{A},\mathsf{F}}(n) = 1] \leq \frac{1}{2} + \mathsf{negl(n)}$.