Modern Cryptography

Pseudorandomness and Pseudorandom Generator

Shashank Singh

Pseudorandomness

Pseudo-randomness?

- Pseudorandomness is a property of a probability distribution.
- In cryptography, we deal with probability distributions having sample space $\{0, 1\}^{128}$ or even bigger.

Remark

- Listing all probabilities in such a vast sample space can be challenging and often not possible.
- We define distributions using sampling algorithms, which effectively draw elements from the specified distribution.
- This approach allows us to manage complexity while still being valuable to crypto applications.

Pseudo-randomness..

Definition 1 (Pseudorandom)

Let D_n be a distribution over $\ell(n)$ bit strings i.e., on the set $\{0,1\}^{\ell(n)}$, $\{D_n\}$ is said to be a pseudorandom distribution if for every PPT algorithm $\mathscr{A}$, there is a negligible function $\varepsilon()$ such that,

$$|\Pr_{s \leftarrow D_n}[\mathscr{A}(s) = 1] - \Pr_{s \leftarrow U_{\ell(n)}}[\mathscr{A}(s) = 1]| < \varepsilon(n), \quad (1)$$

where $U_{\ell(n)}$ is a uniform distribution on $\{0,1\}^{\ell(n)}$. ♣

Recap: $\varepsilon(n) = o\left(\frac{1}{n^c}\right)$ for all $c \in \mathbb{N}$. In other words, ε is smaller than any inverse polynomial function of n.

Remark

A distribution D on $\{0,1\}^\ell$ is called pseudorandom if it passes all efficient statistical tests. For example:

- $\Pr_{s \leftarrow D}\left[\oplus_{i=1}^{\ell} s_i = 1\right] = \frac{1}{2}$, where s_i is the i-th bit of s.
- $\Pr_{s \leftarrow D}[\text{last bit of } s \text{ is } 1] = \frac{1}{2}$.

The NIST Statistical Test Suite, outlined in NIST SP 800-22 Rev.1a, is a standard collection of statistical tests used to assess the randomness of binary sequences produced by true and pseudo-random number generators for cryptographic applications.

Pseudorandom Generator

Definition 1 (Pseudorandom Generator)

A pseudo-random generator G is a deterministic polynomial-time algorithm which takes input a string $s \xleftarrow{\$} \{0,1\}^n$, and outputs a string $G(s) \in \{0,1\}^{\ell(n)}$, for some polynomials $\ell(n)$, with the following properties:

- $\ell(n) > n \quad \forall n$.
- For any PPT algorithm $\mathscr{A}$, there is a negligible function $\varepsilon()$, such that

$$|\Pr_{s \xleftarrow{\$} \{0,1\}^n}[\mathscr{A}(s) = 1] - \Pr_{s \xleftarrow{\$} \{0,1\}^{\ell(n)}}[\mathscr{A}(s) = 1]| < \varepsilon(n)$$

♣

Pseudorandom Generator..

In informal terms, a pseudorandom generator G is an efficient, deterministic algorithm that converts a short, uniform string known as the seed into a longer output string that appears uniform.

Does there exist a PRG?

Remark

- We do not know how to definitively prove the existence of pseudorandom generators; however, we have compelling reasons to believe that they do exist.
- Furthermore, there are several practical constructions of candidate pseudorandom generators, known as stream ciphers, for which no efficient distinguishers are currently known.

Pseudo One Time Pad encryption scheme

Let $G : \{0,1\}^n \mapsto \{0,1\}^{\ell(n)}$, be a PRG, and let $\mathscr{M} = \mathscr{C} = \{0,1\}^{\ell(n)}$, while $\mathscr{K} = \{0,1\}^n$. We define the pseudo OTP through the tuple $\Pi = (\text{GEN}, \text{ENC}, \text{DEC})$ as follows:

- The keygen algorithm GEN, returns a key chosen uniformly from $\{0,1\}^n$.
- $\text{ENC}(k, m) = G(k) \oplus m$ for $m \in \mathscr{M}$ and $k \in \mathscr{K}$.
- $\text{DEC}(k, c) = G(k) \oplus c$ for $c \in \mathscr{C}$ and $k \in \mathscr{K}$.

♣

Theorem

The pseudo one-time pad is a fixed-length private-key encryption scheme that has indistinguishable encryptions in the presence of an eavesdropper. ♡

Proof. On the contrary, suppose there exists a PPT adversary $\mathscr{A}$ for which $\Pr\left[\mathrm{PrivK}^{\mathrm{eav}}_{\mathscr{A},\Pi}(n) = 1\right] > \frac{1}{2} + \varepsilon(n)$. Using $\mathscr{A}$, we construct a PPT distinguisher $\mathscr{D}$ as follows. On input $s \in \{0,1\}^{\ell(n)}$,

- $\mathscr{D}$ runs $\mathscr{A}$ and gets a pair of messages $m_0, m_1 \in \mathscr{M}$.
- $\mathscr{D}$ gives $c := s \oplus m_b$, where $b \xleftarrow{\$} \{0,1\}$, to $\mathscr{A}$ and gets a bit b' back from $\mathscr{A}$.
- $\mathscr{D}$ return $b' \stackrel{?}{=} b$.

□