Cryptography 101: Pseudorandomness

ComComX • 26 Feb 2026

The key interested question motivated by the Shannon's theorem: *can we do encryption using much shorter key than the length of plaintext?*

Pseudorandom generator

They idea is to generate a short (truly) random seed and then use a deterministic algorithm to stretch this seed to a longer sequence that "looks like" truly random. This is impossible under the requirements of *correctness* and *perfect indistinguishability*. However, if we relax the second requirement by assuming a reasonable upper bound on the computational power of the adversary (e.g., probabilistic-polynomial time computation), then we have what's called *Pseudorandom Generator* (PRG).

Definition 1 (Negligible function)**.** *A function $\nu : \mathbb{N} \to \mathbb{R}$ is negligible if for every positive polynomial function p, there exists a λ_0 such that for every $\lambda \geq \lambda_0, \nu(\lambda) < \frac{1}{p(\lambda)}$.*

Definition 2 (Distinguishing advantage)**.** *For two random variables D_0 and D_1 and an algorithm A, define A's advantage in distinguishing between D_0 and D_1 as*

$$\mathrm{adv}_A^{D_0,D_1} := \left|\Pr_{x \leftarrow D_0}[A(x) = 1] - \Pr_{x \leftarrow D_1}[A(x) = 1]\right|.$$

Definition 3 (Computational indistinguishability)**.** *Two ensembles of random variables X_λ and Y_λ is said to be computational indistinguishable if for any probabilistic-polynomial time (non-uniform) algorithm A,*

$$\mathrm{adv}_A^{X_\lambda,Y_\lambda} := \mathrm{adv}_A^{X,Y}(\lambda) := \left|\Pr[A(X_\lambda, 1^\lambda) = 1] - \Pr[A(Y_\lambda, 1^\lambda) = 1]\right| = \mathrm{negl}(\lambda).$$

Recall: an encryption scheme $\Pi = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ is perfect indistinguishable if for any plaintexts m_0 and m_1 and any algorithm A,

$$\mathrm{adv}_A^{C(m_0),C(m_1)} = 0$$

where $C(m) = \mathsf{Enc}(\mathsf{KeyGen}(), m)$ is a random variable over the space of ciphertexts.

We now give a relaxed version of this definition:

Definition 4 (Computational indistinguishability of encryption scheme)**.** *An encryption scheme* $\Pi = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ *is said to be computational indistinguishable if for any plaintexts* m_0 *and* m_1 *and any PPT-algorithm* A*,*

$$\mathrm{adv}_A^{C(m_0,\lambda),C(m_1,\lambda)} = \mathrm{negl}(\lambda)$$

where $C(m, \lambda) = \mathsf{Enc}(1^\lambda, \mathsf{KeyGen}(1^\lambda), m)$ *is a random variable over the space of ciphertexts.*

Definition 5 (Truly random variable)**.** *A random variable* X *over* $\{0,1\}^n$ *is called truly random if for any* $x \in \{0,1\}^n$*,* $\Pr[X = x] = 1/2^n$*.*

Definition 6 (Pseudorandom variable)**.** *A random variable over* $\{0,1\}^n$ *is pseudorandom variable if it is computationally indistinguishable from a truly random variable over the same sample space.*

An algorithm that generates a pseudorandom variable is called a Pseudorandom Generator (PRG).

Definition 7 (Pseudorandom Generator, standard version)**.** *A PRG is a polynomial time algorithm* G *such that for every* $\lambda \in \mathbb{N}$*,* G *maps inputs from* $\{0,1\}^\lambda$ *to outputs in* $\{0,1\}^{m(\lambda)}$ *and satisfies that*

- ***Expansion****: for all sufficiently large* λ*,*

$$m(\lambda) > \lambda$$

- ***Pseudorandomness****: for any PPT-distinguisher* A*,*

$$\mathrm{adv}_A^{G(U_\lambda),U_{m(\lambda)}} = \mathrm{negl}(\lambda)$$

where $U_{m(\lambda)}$ *is a truly random variable over* $\{0,1\}^{m(\lambda)}$*.*

The existence of PRGs has not been proved. Nevertheless, it is believed that PRGs exist.

Conjecture 8 (Existence of PRGs)**.** *For every integer* λ*, there exists a PRG that maps* λ *bits to* $\lambda + 1$ *bits.*

Theorem 9. *PRG can not exist unless* $\mathsf{P} \neq \mathsf{NP}$*.*

Proof. Assume for contradiction that G is a PRG and $\mathsf{P} = \mathsf{NP}$. Let construct a distinguisher D with non-negligible advantage in distinguishing $G(U_\lambda)$ and $U_{m(\lambda)}$. The idea is that given $y \in \{0,1\}^{m(\lambda)}$, since $\mathsf{P} = \mathsf{NP}$, we can find, in polynomial time (by simply guessing), the preimage $x \in \{0,1\}^\lambda$ such that $G(x) = y$ if there is such an x. The distinguisher D simply returns 1 if x exists and 0 otherwise. Then, it follows that

$$\begin{aligned}\mathrm{adv}_D^{G(U_\lambda),U_{m(\lambda)}} &= \left|\Pr[D(G(U_\lambda)) = 1] - \Pr[D(U_{m(\lambda)}) = 1]\right| \\ &= \left|1 - \frac{2^\lambda}{2^{m(\lambda)}}\right|,\end{aligned}$$

which is noticeable and contradict the assumption that G is a PRG. □

Example 10 (Subset sum PRG)**.** *The subset sum problem is as follows: given n numbers $A = \{a_1, \ldots, a_n\}$ and a number T, each with ℓ bits long. Find a subset $S \subseteq A$ such that $\sum_{a \in S} a = T \mod 2^\ell$. This problem has been proven to be NP-hard. We can view the problem as inverting the following function:*

$$f(A, S) = \left(A, \sum_{a \in S} a \mod 2^\ell \right).$$

The input length of f is $n\ell + n$ and the output length is $n\ell + \ell$. This means that if $\ell > n$, we have a PRG since the best algorithm for inverting the above function or, to solve the subset sum problem, still requires exponential time.

Theorem 11. *If there is a PRG G that expands λ bits to $\lambda + 1$ bits, then for any polynomial $p(\lambda)$, there is also a PRG G' that expands λ bits to $m(\lambda) = \lambda + p(\lambda)$ bits.*

Proof. Let construct G' as follows: given a uniformly random input $x_0 \in \{0,1\}^\lambda$, iterating G for $p(\lambda)$ iterations, where the last λ bits of the output of each iteration is used as the input of the next iteration:

1. $x_0 \leftarrow U_\lambda$
2. for $i = 1, \ldots, p(\lambda)$: $(b_i, x_i) \leftarrow G(x_{i-1}, 1^\lambda)$ where $|x_j| = \lambda$ for all j
3. Output $x = (b_1, \ldots, b_{p(\lambda)}, x_{p(\lambda)})$.

Since G runs in polynomial time, G' also runs in polynomial time. What remains is to show that the final output x of the above procedure is pseudorandom.

We show the pseudorandom of x using the **hybrid argument** as follows. Let construct $p(\lambda)$ hybrids strings $H_0, H_1, \ldots, H_{p(\lambda)}$ where $H_j \in \{0,1\}^{m(\lambda)}$ is defined as follows:

- The first j bits are uniformly random
- The last $m(\lambda) - j$ bits are generated by simulating the chain of G for $m(\lambda) - j - \lambda$ iterations, *starting with a uniformly random seed*.

Thus, H_0 follows the same distribution as of the output x and $H_{p(\lambda)}$ is a truly random string. Assume for contradiction that x is not pseudorandom. That is, there is a distinguisher D and a polynomial $q(\lambda)$ such that

$$\mathrm{adv}_D^{H_{p(\lambda)}, H_0} = \left| \Pr[D(H_{p(\lambda)}, 1^\lambda) = 1] - \Pr[D(H_0, 1^\lambda) = 1] \right| \geq \frac{1}{q(\lambda)}.$$

It follows by the triangle inequality that

$$\begin{aligned}\sum_{j=0}^{p(\lambda)-1} &\left|\Pr[D(H_j, 1^\lambda) = 1] - \Pr[D(H_{j+1}, 1^\lambda) = 1]\right| \\ &\geq \left|\Pr[D(H_{p(\lambda)}, 1^\lambda) = 1] - \Pr[D(H_0, 1^\lambda) = 1]\right| \\ &\geq \frac{1}{q(\lambda)},\end{aligned}$$

and therefore, by the pigeonhole principle, there exists $j^* \in [p(\lambda)]$ such that

$$\left|\Pr[D(H_{j^*}, 1^\lambda) = 1] - \Pr[D(H_{j^*+1}, 1^\lambda) = 1]\right| \geq \frac{1}{q(\lambda) \cdot p(\lambda)}.$$

That is, the distinguisher D distinguishes H_{j^*} and H_{j^*+1} with non-negligible advantage $\frac{1}{q(\lambda)\cdot p(\lambda)}$.

Given the above D, we construct a distinguisher D' that distinguishes $G(U_\lambda)$ and $U_{\lambda+1}$ as follows: for an input $y \in \{0,1\}^{\lambda+1}$,

1. Construct a string z:
    - The first j^* bits of z are uniformly random
    - The (j^*+1)-th bit is the first bit of y
    - The last remaining $m(\lambda) - j^* - 1$ bits are generated by simulating the chain of G starting with $y_{2:\lambda+1}$.
2. Output $D(z, 1^\lambda)$.

Observe that if y is truly random, then z has the same distribution as of H_{j^*+1}. On the other hand, if $y = G(U_\lambda)$, then z has the same distribution as H_{j^*}. Since D can distinguish H_{j^*} and H_{j^*+1} with noticeable advantage $\frac{1}{q(\lambda)\cdot p(\lambda)}$, D' inherits this advantage in distinguishing $G(U_\lambda)$ and $U_{\lambda+1}$, and thus contradicts the fact that G is a PRG. □

Encryption using PRG

We can now have an encryption scheme with keys shorter than input messages. Let $G : \{0,1\}^\lambda \to \{0,1\}^{m(\lambda)}$ be a PRG, $\mathcal{M} = \{0,1\}^{m(\lambda)}$, $\mathcal{K} = \{0,1\}^\lambda$, and $\mathcal{C} = \{0,1\}^{m(\lambda)}$. Define

- $\mathsf{KeyGen}(1^\lambda)$:
    1. $k_s \leftarrow \mathsf{KeyGen}(1^\lambda)$
- $\mathsf{Enc}(1^\lambda, k_s, m)$:
    1. $k_r \leftarrow G(1^\lambda, k_s)$
    2. output $c = m \oplus k_r$

- $\mathsf{Dec}(1^\lambda, k_s, c)$:
  1. $k_r \leftarrow G(1^\lambda, k_s)$
  2. output $\hat{m} = c \oplus k_r$

Theorem 12. *The above encryption scheme satisfies correctness and computational indistinguishability.*

Proof. The proof is similar to the previous proof of one-time pad scheme:

Correctness: we have

$$\begin{aligned}\mathsf{Dec}(1^\lambda, k_s, \mathsf{Enc}(1^\lambda, k_s, m)) &= \mathsf{Enc}(1^\lambda, k_s, m)) \oplus k_r \\ &= m \oplus k_r \oplus k_r \\ &= m.\end{aligned}$$

Computational indistinguishability: we prove the second property of our encryption scheme by a security reduction. Specifically, assume for contradiction that G is a PRG and there exist two messages $m_0, m_1 \in \mathcal{M}$ and a distinguisher D with non-negligible advantage in distinguishing $C(m_0, \lambda)$ and $C(m_1, \lambda)$, where $C(m_0, \lambda)$ and $C(m_1, \lambda)$ are the corresponding random variables under our encryption scheme. That is, there is a polynomial $p(\lambda)$ such that

$$\begin{aligned}\mathrm{adv}_D^{C(m_0,\lambda),C(m_1,\lambda)} &= \left|\Pr[D(C(m_0,\lambda), 1^\lambda) = 1] - \Pr[D(C(m_1,\lambda), 1^\lambda) = 1]\right| \\ &\geq \frac{1}{p(\lambda)}.\end{aligned}$$

The idea is that given this distinguisher D, we construct a distinguisher D' that has a noticeable advantage in distinguishing $G(U_\lambda)$ and $U_{m(\lambda)}$, and thus contradicts the fact that G is a PRG. To construct D', we first observe that if we use the truly random keys instead of keys generated by G, then the scheme is identical to the one-time pad scheme and therefore, any distinguisher will have zero advantage:

$$\left|\Pr[D(U_{m(\lambda)} \oplus m_0, 1^\lambda) = 1] - \Pr[D(U_{m(\lambda)} \oplus m_1, 1^\lambda) = 1]\right| = 0.$$

It follows that

$$
\begin{aligned}
\frac{1}{p(\lambda)} &\leq \left|\Pr[D(C(m_0,\lambda),1^\lambda)=1] - \Pr[D(C(m_1,\lambda),1^\lambda)=1]\right| \\
&= \Big|\Pr[D(C(m_0,\lambda),1^\lambda)=1] - \Pr[D(U_{m(\lambda)}\oplus m_0,1^\lambda)=1] \\
&\qquad + \Pr[D(U_{m(\lambda)}\oplus m_1,1^\lambda)=1] - \Pr[D(C(m_1,\lambda),1^\lambda)=1]\Big| \\
&\leq \Big|\Pr[D(C(m_0,\lambda),1^\lambda)=1] - \Pr[D(U_{m(\lambda)}\oplus m_0,1^\lambda)=1]\Big| \\
&\qquad + \Big|\Pr[D(U_{m(\lambda)}\oplus m_1,1^\lambda)=1] - \Pr[D(C(m_1,\lambda),1^\lambda)=1]\Big|
\end{aligned}
$$

which implies that either

$$
\left|\Pr[D(C(m_0,\lambda),1^\lambda)=1] - \Pr[D(U_{m(\lambda)}\oplus m_0,1^\lambda)=1]\right| \geq \frac{1}{2\cdot p(\lambda)}
$$

or

$$
\left|\Pr[D(U_{m(\lambda)}\oplus m_1,1^\lambda)=1] - \Pr[D(C(m_1,\lambda),1^\lambda)=1]\right| \geq \frac{1}{2\cdot p(\lambda)}.
$$

Assume w.l.g. that the former is the case. We construct $D'(x,1^\lambda)$ as follows:

- $y \leftarrow m_0 \oplus x$ (here m_0 is an advice string that is independent of x)
- Output $D(y,1^\lambda)$.

Then, we have

$$
\begin{aligned}
\mathrm{adv}_{D'}^{G(U_\lambda),U_{m(\lambda)}} &= \left|\Pr[D'(G(U_\lambda),1^\lambda)=1] - \Pr[D'(U_{m(\lambda)},1^\lambda)=1]\right| \\
&= \left|\Pr[D(G(U_\lambda)\oplus m_0,1^\lambda)=1] - \Pr[D(U_{m(\lambda)}\oplus m_0,1^\lambda)=1]\right| \\
&\geq \frac{1}{2\cdot p(\lambda)},
\end{aligned}
$$

which contradicts the assumption that G is a PRG, and therefore, the computational distinguishability holds. □

Pseudorandomness and Next-bit unpredictability

We give an alternative definition of the PRG:

Definition 13 (Pseudorandom Generator, Next-bit unpredictability)**.** *A PRG is a polynomial time algorithm G such that for every $\lambda \in \mathbb{N}$, G maps inputs from $\{0,1\}^\lambda$ to outputs in $\{0,1\}^{m(\lambda)}$ and satisfies that*

- ***Expansion**: for all sufficiently large λ,*

$$m(\lambda) > \lambda$$

- ***Computationally next-bit unpredictability**: for any PPT-predictor A and all $i = 1, 2, \ldots, m(\lambda)$,*

$$\Pr_{x \leftarrow U_\lambda, y \leftarrow G(x)}[A(y_{1:i-1}, 1^\lambda) = y_i] = \frac{1}{2} + \mathrm{negl}(\lambda).$$

The following theorem shows that the two definitions are indeed equivalent:

Theorem 14. *A variable $X \in \{0,1\}^n$ is pseudorandom iff it is computationally next-bit unpredictable.*

Proof. We need to show the following two statements are equivalent:

- **Pseudorandomness**: for any PPT-distinguisher D,

$$\begin{aligned}\mathrm{adv}_D^{X,U_n} &= \left|\Pr[D(X, 1^\lambda) = 1] - \Pr[D(U_n, 1^\lambda) = 1]\right| \\ &= \mathrm{negl}(\lambda).\end{aligned}$$

  Computationally next-bit unpredictability: for any PPT-predictor P and all $i = 1, 2, \ldots, n$,

$$\Pr[P(X_{1:i-1}, 1^\lambda) = X_i] = \frac{1}{2} + \mathrm{negl}(\lambda).$$

Pseudorandomness $\Longrightarrow$ Computationally next-bit unpredictable:
Assume for contradiction that X is pseudorandomness but computationally next-bit predictable. I.e., there is a predictor P, an $i \in [n]$, and a polynomial $p(\lambda)$ such that

$$\Pr[P(X_{1:i-1}, 1^\lambda) = X_i] \geq \frac{1}{2} + \frac{1}{p(\lambda)}.$$

We use P to construct a distinguisher D that has non-negligible advantage in distinguishing X and U_n, and thus give a constradiction. Let $D(x)$ be as follows:

- $\bar{x}_i \leftarrow P(x_{1:i-1}, 1^\lambda)$
- If $\bar{x}_i = x_i$ then output 1
- Otherwise, output 0.

Clearly, we have

$$\Pr[D(X, 1^\lambda) = 1] = \Pr[P(X_{1:i-1}, 1^\lambda) = X_i] \geq \frac{1}{2} + \frac{1}{p(\lambda)}.$$

On the other hand,

$$\Pr[D(U_n, 1^\lambda) = 1] = \Pr_{y \leftarrow U_n}[P(y_{1:i-1}, 1^\lambda) = y_i] = \frac{1}{2}.$$

Therefore,

$$\mathrm{adv}_D^{X,U_n} = \left|\Pr[D(X, 1^\lambda) = 1] - \Pr[D(U_n, 1^\lambda) = 1]\right| \geq \frac{1}{p(\lambda)}$$

which contradicts the assumption that X is pseudorandom.

(Note: since we only need to show the existence of the distinguisher D, it doesn't matter what is the value of i in the above proof (and also in the previous proofs). Just the existence of an i is enough.)

Pseudorandomness $\Longleftarrow$ **Computationally next-bit unpredictable**:
We prove the backward direction using the **hybrid argument**.

Assume for contradiction that X is computationally next-bit unpredictable but is not pseudorandomness. That is, for any $i \in [n]$ and any PPT-predictor P,

$$\Pr[P(X_{1:i-1}, 1^\lambda) = X_i] = \frac{1}{2} + \mathrm{negl}(\lambda)$$

and there exist a PPT-distinguisher D and polynomial $p(\lambda)$ such that

$$\mathrm{adv}_D^{X,U_n} = \left|\Pr[D(X, 1^\lambda) = 1] - \Pr[D(U_n, 1^\lambda) = 1]\right| \geq \frac{1}{p(\lambda)}.$$

Let construct hybrids $H_0, \ldots, H_n$ where H_j is defined as:

- The first j bits are the first j bits of X
- The last $n - j$ bits are uniformly random bits.

We then have H_n is identical to the variable X and H_0 is uniformly random strings. By the triangle inequality, we have that

$$\begin{aligned}
\frac{1}{p(\lambda)} &\leq \left|\Pr[D(X, 1^\lambda) = 1] - \Pr[D(U_n, 1^\lambda) = 1]\right| \\
&= \left|\Pr[D(H_n, 1^\lambda) = 1] - \Pr[D(H_0, 1^\lambda) = 1]\right| \\
&\leq \sum_{i=0}^{n-1} \left|\Pr[D(H_i, 1^\lambda) = 1] - \Pr[D(H_{i+1}, 1^\lambda) = 1]\right|.
\end{aligned}$$

By the pigeonhole principle, this implies that the must be an $i^* \in [n]$ such that

$$\left|\Pr[D(H_{i^*}, 1^\lambda) = 1] - \Pr[D(H_{i^*+1}, 1^\lambda) = 1]\right| \geq \frac{1}{n \cdot p(\lambda)}. \tag{1}$$

Let define a string H'_{i^*+1} be identical as H_{i^*+1} excepting that the (i^*+1)-th bit is flipped. The following result will be useful:

Claim 15.

$$\Pr[D(H_{i^*}, 1^\lambda) = 1] = \frac{1}{2}\left(\Pr[D(H_{i^*+1}, 1^\lambda) = 1] + \Pr[D(H'_{i^*+1}, 1^\lambda) = 1]\right).$$

Proof. Note that for any string $H \in \{0,1\}^n$, we have by the law of total probability that

$$\Pr[D(H, 1^\lambda) = 1] = \sum_{x \in \{0,1\}^n} \Pr[D(x, 1^\lambda) = 1] \cdot \Pr[x = H].$$

Moreover, since H_{i^*+1} and H'_{i^*+1} are identical excepting their i^*-th bits are flipped, we also have

$$\Pr[x = H_{i^*}] = \frac{\Pr[x = H'_{i^*+1}] + \Pr[x = H_{i^*+1}]}{2}.$$

Combining the two equalities concludes the proof. □

Next, we use this D to construct a predictor Pred that predicts the (i^*+1)-th bit of X as follows: given the first i^* bits of X,

1. Construct a string y with length n:
   - The first i^* bits of y is the the first i^* bits of X
   - The last $n - i^*$ bits is uniformly random
2. $b = D(y, 1^\lambda)$
3. If $b = 1$ then output y_{i^*+1} else output $\bar{y}_{i^*+1}$.

We have that

$$\begin{aligned}
&\Pr[\mathsf{Pred}(X_{1:i^*}, 1^\lambda) = X_{i^*+1}] \\
&\quad= \Pr[(D(y, 1^\lambda) = 1) \cap (y_{i^*+1} = X_{i^*+1})] \\
&\qquad + \Pr[(D(y, 1^\lambda) = 0) \cap (y_{i^*+1} = \bar{X}_{i^*+1})] \\
&\quad= \Pr[D(y, 1^\lambda) = 1 | y_{i^*+1} = X_{i^*+1}] \cdot \Pr[y_{i^*+1} = X_{i^*+1}] \\
&\qquad + \Pr[D(y, 1^\lambda) = 0 | y_{i^*+1} = \bar{X}_{i^*+1}] \cdot \Pr[y_{i^*+1} = \bar{X}_{i^*+1}] \\
&\quad= \frac{1}{2}\left(\Pr[D(y, 1^\lambda) = 1 | y_{i^*+1} = X_{i^*+1}] + \Pr[D(y, 1^\lambda) = 0 | y_{i^*+1} = \bar{X}_{i^*+1}]\right) \\
&\quad= \frac{1}{2}\left(\Pr[D(H_{i^*+1}, 1^\lambda) = 1] + 1 - \Pr[D(H'_{i^*+1}, 1^\lambda) = 1]\right) \\
&\quad= \frac{1}{2} + \Pr[D(H_{i^*+1}, 1^\lambda) = 1] - \frac{1}{2}\left(\Pr[D(H_{i^*+1}, 1^\lambda) = 1] + \Pr[D(H'_{i^*+1}, 1^\lambda) = 1]\right) \\
&\quad= \frac{1}{2} + \left(\Pr[D(H_{i^*+1}, 1^\lambda) = 1] - \Pr[D(H_{i^*}, 1^\lambda) = 1]\right) \\
&\quad\geq \frac{1}{2} + \frac{1}{n \cdot p(\lambda)}
\end{aligned}$$

where the third identity is due to the fact that y_{i^*+1} is uniformly random, the fourth identity is due to the definition of H'_{i^*+1}, the sixth identity is due to claim

15, and the inequality is from (1). This inequality contradicts the assumption that X is computationally next-bit unpredictable, and thus X must be pseudorandom. $\square$