# Cryptography 101: The Goldreich-Levin theorem

ComComX • 5 Mar 2026

We give a way to construct a PRG. Consider a function (family)

$$F = \{f_\lambda : \{0,1\}^\lambda \to \{0,1\}^{m(\lambda)}\}_{\lambda \in \mathbb{N}}.$$

**Definition 1** (One way functions (OWF))**.** *$F$ is a OWF if it satisfies that*

- *It is efficiently computable: there is a deterministic polynomial time algorithm that compute $f_\lambda(x)$ given $(1^\lambda, x)$.*
- *It is hard to invert: for any PPT algorithm $A$,*

$$\Pr[f_\lambda(1^\lambda, A(1^\lambda, f_\lambda(x))) = f_\lambda(x)] = \mathrm{negl}(\lambda).$$

Note: $m(\lambda)$ needs not to be larger than $\lambda$.

**Theorem 2.** *OWFs can not exist unless* $\mathsf{P} \neq \mathsf{NP}$.

Note: this does not means that if $\mathsf{P} \neq \mathsf{NP}$ then a OWF exists. In fact, this is currently unknown to be true or not.

**Definition 3** (One way permutation (OWP))**.** *$F$ is a OWP if it satisfies that*

- *It is an OWF with $m(\lambda) = \lambda$ for every $\lambda \in \mathbb{N}$*
- *For every $\lambda \in \mathbb{N}$, $f_\lambda$ is a bijection.*

We can think of the hardcore bit as the source where the hardness of inverting a OWF lies. A hardcore predicate is an efficient function that computes the hardcore bit.

**Definition 4** (Hardcore predicates (HCP))**.** *A function family $H = \{h_\lambda : \{0,1\}^\lambda \to \{0,1\}\}_{\lambda \in \mathbb{N}}$ is a HCP for a OWF $F$ if for any PPT algorithm $A$,*

$$\Pr_{x \leftarrow U_\lambda}[A(1^\lambda, f_\lambda(x)) = h_\lambda(x)] = \frac{1}{2} + \mathrm{negl}(\lambda).$$

**Theorem 5.** *If a OWP $F$ has a HCP $H$, then there exists a PRG.*

*Proof.* For any $\lambda \in \mathbb{N}$, let define $G(x) = f_\lambda(x)||h_\lambda(x)$ where $x \in \{0,1\}^\lambda$ is a truly random string. That is, $G$ maps a random $\lambda$-bit string to a $(\lambda+1)$-bit string. We need to show that the output of $G$ is pseudorandom.

We show this using the equivalent of pseudorandomness and computationally next-bit unpredictability. Specifically, let $y = G(x)$. Then it suffices to show that for any $i \in [\lambda + 1]$ and any PPT predictor $P$,

$$\Pr[P(y[1 : i-1], 1^\lambda) = y_i] = \frac{1}{2} + \text{negl}(\lambda).$$

Note that $y[1 : \lambda] = f_\lambda(x)$. Since $\{f_\lambda\}$ is a OWP and that $x$ is truly random, $f_\lambda(x)$ is also a truly random string, and thus the computationally next-bit unpredictability holds for all $i \in [\lambda]$. Finally, for $i = \lambda + 1$, since $\{h_\lambda\}$ is a hardcore predicate for $F$ and $y_{\lambda+1} = h_\lambda(x)$, this is also true that

$$\Pr[P(y[1 : \lambda], 1^\lambda) = y_{\lambda+1}] = \frac{1}{2} + \text{negl}(\lambda).$$

This concludes the proof. □

There is no single HCP for all OWFs. However, the following Goldreich-Levin Theorem shows that given any OWF $f$, we can construct a OWF $g$ and a HCP for $g$.

**Theorem 6** (The Goldreich-Levin Theorem)**.** *Let $F$ be a OWF. Let define a function families $G$ and $H$ as*

$$G = \{g_{2\lambda} : \{0,1\}^{2\lambda} \to \{0,1\}^{2\lambda}\}_{\lambda \in \mathbb{N}}$$

$$H = \{h_{2\lambda} : \{0,1\}^{2\lambda} \to \{0,1\}\}_{\lambda \in \mathbb{N}}$$

*where*

$$g_{2\lambda}(x) = x_1 \dots x_\lambda || f_\lambda(x_{\lambda+1} \dots x_{2\lambda})$$

$$h_{2\lambda}(x) = \langle x_1 \dots x_\lambda, x_{\lambda+1} \dots x_{2\lambda} \rangle \mod 2.$$

*Then, for any PPT algorithm $A$,*

$$\Pr_{x \leftarrow U_{2\lambda}}[A(1^\lambda, g_{2\lambda}(x)) = h_{2\lambda}(x)] = \frac{1}{2} + \text{negl}(\lambda).$$

*That is, $G$ is a OWF with HCP $H$.*

*Proof.* Assume for contradiction that there is a predictor $A$ such that

$$\Pr_{x \leftarrow U_{2\lambda}}[A(1^\lambda, g_{2\lambda}(x)) = h_{2\lambda}(x)] = \frac{1}{2} + \epsilon(\lambda)$$

where $\epsilon(\lambda)$ is a noticeable function. We show that $F$ is not a OWF.

For the ease of exposition, let write $x \in \{0,1\}^{2\lambda}$ as $(y, r)$ and $g_{2\lambda}(x) = (f_\lambda(y), r)$ where $|y| = |r| = \lambda$.

**Attempt 1**: suppose $A$ is a "perfect" predictor, i.e., $\forall y, \epsilon(\lambda) = 1/2$. This implies that

$$\forall (y, r) \in \{0,1\}^{2\lambda}, A(1^\lambda, f_\lambda(y), r) = \langle y, r \rangle \mod 2.$$

Observe that for every $i$, $\langle y, e_i \rangle \mod 2 = y_i$ where $e_i$ is the $i$-th unit vector. This means that given any $f_\lambda(y)$, we can recover the $y$ in $\lambda$ queries from $A$ by using $r \in \{e_1, \ldots, e_\lambda\}$, thus contradicts the fact that $F$ is a OWF.

**Attempt 2**: now suppose $\forall y, \epsilon(\lambda) = 0.3$:

$$\forall y \in \{0,1\}^\lambda, \Pr_{r \leftarrow U_\lambda}[A(1^\lambda, f_\lambda(y), r) = \langle y, r \rangle \mod 2] = 0.8.$$

Now, we can not claim that

$$\forall y, i, \Pr[A(1^\lambda, f_\lambda(y), e_i) = \langle y, e_i \rangle \mod 2] = 0.8$$

as we did before. The solution for this is randomization. The first observation is that if we sample $r$ randomly, then $r \oplus e_i$ is also a truly random string. This means that $\forall y$ and $i$, we have both

$$\Pr[A(1^\lambda, f_\lambda(y), r) = \langle y, r \rangle \mod 2] = 0.8$$

and

$$\Pr[A(1^\lambda, f_\lambda(y), r \oplus e_i) = \langle y, r \oplus e_i \rangle \mod 2] = 0.8.$$

The second observation is that by the linearity of the inner product, we have

$$\langle y, e_i \rangle = \langle y, r \rangle \oplus \langle y, r \oplus e_i \rangle.$$

It follows that if $A$ is right in both queries with $r$ and $r \oplus e_i$, then we can recover $y_i$ by

$$y_i = \langle y, e_i \rangle = A(1^\lambda, f_\lambda(y), r) \oplus A(1^\lambda, f_\lambda(y), r \oplus e_i).$$

The probability of this event is:

$$\begin{aligned} \Pr[A \text{ is right in both queries}] &= 1 - \Pr[A \text{ is wrong in at least one query}] \\ &= 1 - (0.2 + 0.2) \\ &= 0.6. \end{aligned}$$

That is, given $f_\lambda(y)$, we can recover $y$ with probability of 0.6 on each dimension $i$. This probability can be amplified by repeating the process $k$ times and outputting the more frequent output. Specifically, for each dimension $i$:

- Sample $T$ random vectors $r^1, \ldots, r^T k$.
- For each vector $r^t$, compute $A(1^\lambda, f_\lambda(y), r^t) \oplus A(1^\lambda, f_\lambda(y), r^t \oplus e_i)$
- Output the more frequent result.

Suppose the success probability on each dimension is $\frac{1}{2} + \delta$ (which is 0.6 in above setting). Let $X^t$ be the indicator for the event that the output of the $t$-th test is correct and $X = \sum_{t=1}^{T} X^t$. We want to bound the probability that majority of the $T$ outputs is incorrect, i.e., $\Pr[X \leq T/2]$. We have

$$\begin{aligned}
\Pr[X \le T/2] &= \Pr[X - \mathbb{E}[X] \le T/2 - \mathbb{E}[X]] \\
&= \Pr\left[X - \mathbb{E}[X] \le T/2 - \sum_{t=1}^{T} \mathbb{E}[X^t]\right] \\
&= \Pr\left[X - \mathbb{E}[X] \le T/2 - \sum_{t=1}^{T} \left(\frac{1}{2} + \delta\right)\right] \\
&= \Pr\left[X - \mathbb{E}[X] \le -\delta T\right] \\
&\le \frac{\mathrm{Var}(X)}{\delta^2 T^2},
\end{aligned}$$

where the last inequality is due to Chebyshev's inequality. Note that variables $X^t$ are mutually independent. Therefore,

$$\mathrm{Var}(X) = \sum_{t=1}^{T} \mathrm{Var}(X^t) = T \cdot 0.6 \cdot (1 - 0.6) = \frac{6T}{25}.$$

By setting $T = O(\lambda/\delta^2)$, the failure probability on each dimension is reduced to at most $1/10\lambda$. Then, by union bound, the probability of successfully recovering $y$ is at least $1 - \lambda \cdot 1/10\lambda = 0.9$, which is noticeable and contradicts the fact that $F$ is a OWF.

**Last attempt**: the proof in the last attempt is valid only when $\delta > 0$, which requires $\epsilon(\lambda)$ to be strictly larger than $\frac{1}{4}$. But our goal is for any $\epsilon(\lambda)$ that is noticeable? How can we remove the assumption about $\epsilon(\lambda)$ in the last attempt? **This is where Goldreich-Levin gets famous!**

If we look back to the proof, the only issue is this use of union bound:

$$\Pr[A \text{ is right in both queries}] = 1 - \Pr[A \text{ is wrong in at least one query}].$$

If $\epsilon(\lambda) < 0.25$ then this probability drops below 0.5 that fails the majority vote.

The brilliant idea: as we have argued in the last attempt that $\forall y$ and $i$, we have

$$\Pr[A(1^\lambda, f_\lambda(y), r) = \langle y, r\rangle \mod 2] = \frac{1}{2} + \epsilon(\lambda)$$

and

$$\Pr[A(1^\lambda, f_\lambda(y), r \oplus e_i) = \langle y, r \oplus e_i\rangle \mod 2] = \frac{1}{2} + \epsilon(\lambda).$$

Now, what if we only need to do one of the two queries for each $r$, say, we already known the true value of $\langle y, r\rangle$ and only need to query from $A$ the value of $\langle y, r \oplus e_i\rangle$? In this case, we are done with the proof in previous attempt. But how do we know the value of $\langle y, r\rangle$?

The final observation is that in order to use the Chebyshev's inequality does not need $r^1, \ldots, r^T$ to be mutually independent (for mutually independent variables, Chernoff's inequality gives a stronger bound!). Instead, the variables only need to be pairwise independent so that the covariances are zero. The magic is as follows:

- Generate $K = O(\log T)$ random strings $s^1, \ldots, s^K$.
- Generate the set of $T$ strings

$$R = \left\{ r^I = \bigoplus_{k \in I} s^k \middle| I \subseteq [K], I \neq \varnothing \right\}.$$

Then, it can be seen that all these $T$ strings are pairwise independent. Moreover, for each string $r^I \in R$, we have that

$$\langle y, r^I \rangle = \bigoplus_{k \in I} \langle y, s^k \rangle.$$

That is, we only need $K$ values of $\langle y, s^k \rangle$ to compute all $T$ values of $\langle y, r^I \rangle$ for all $I$. Since $K = O(\log T)$, we can try all $2^K$ assignments and still keep the running time in $O(T) = \text{poly}(\lambda)$. This concludes the proof. $\square$