The story of sending cryptography

\textbf{One Way Functions:} (OWF): \textit{"hard to invert"}

A family of functions \( f_x : D_x \rightarrow C_x \) is one way
iff \( \forall A, \Pr [x \in D_x, y = f_x(x), X \leftarrow A(1^n), x' = X] \leq n n(1) \)

\textbf{Pseudorandom Generator (PRG):} \( (\text{length}, \text{randomness}) \)

A family \( f_x : D_x \rightarrow \mathbb{Z}^{2^n} \) is a PRG
iff \( \exists x \in D_x : f_x(X) \approx x \approx x \in \mathbb{Z}^{2^n} : X \)

\textit{Indistinguishable from random sample, given a random seed.}

\textbf{Pseudorandom Function (PRF):} \( f_x : K_x \rightarrow \mathbb{Z}^{2^n} \)

A gets oracle access to \( f_x(K_x) \), can
\textit{Indistinguishable from a random function, even if adversary chooses the inputs to the function.}

\textbf{One Way Functions:} \( \text{"hard to invert" \leq \text{on random input} \)}

\textbf{Examples:}

- \( B, G \)

- \( p \leq |G| \leq \text{groups of size } 161 \text{ or } 2 ")
- Cryptographic hash functions (SHA)
- Universal one-way function

Guaranteed to be one-way, IFF any OWF exist.

What does it look like?

Let's use OWF for encryption!

$m \oplus \text{ser}(g^k)$

and only once.

Why not? Not all bits are hard to predict.

Suppose $S_2$ is a PRG.

Let $f' \colon x \rightarrow \begin{cases} f(x) & \text{if } x \neq 0 \\ 0^2 & \text{if } x = 0 \end{cases}$

Is $f'$ a PRG?

YES

Does PRG imply OWF?

If $f : \{0,1\}^n \rightarrow \{0,1\}^2$ is PRG

is it also OWF?

Given $A$ that wins the OWF game,

Construct $A'$ that wins PRG

$A'(y) = \begin{cases} 2 & \text{if } y \notin \text{ran}(f) \\ \text{accept} & \text{otherwise} \end{cases}$

Why? Codomain of $f$ is much larger than domain.

$\{0,1\}^2 \neq \{0,1\}^n$.

If $y \notin \text{ran}(f)$,

$\Pr[A(f(x) \oplus y) = 1] = \frac{1}{2}$

So, if $A$ can find such a value $y$, it is because $y$ was chosen from the image of $f$, not from $f(x)$.
2.8 DLOG relation
3.4.1 "a hardcore bit of DLOG"
Based on CRT.

Open question: does this hold for positive $(\beta, \gamma)$ in $\mathbb{Z}_q^{256}\times 1$.

**Hardcore predicate** for a OWF $f$
$h_f : D_2 \rightarrow \{0, 1\}^\beta$
"adversary can't predict $h_f(x)$ even after seeing $f(x)$.
\[ \forall A \ P_r \left[ x \notin D_2 : A(f(x)) = h_f(x) \right] \leq \frac{1}{2} + \text{negl}(\lambda) \]
\[ \beta = \text{half}(\beta^x) \text{ for Schnorr groups} \]
- $\text{LSB of RSA}$
- $\text{seq}^{256\times 1}$?

**Universal Hardware Predicate** "Goldreich-Levin"

Let $\epsilon : D_2 \rightarrow D_2$ be a OWF.

Then let $\mathcal{E} : (D_2 \times \{0, 1\}^\beta) \rightarrow (D_1 \times \{0, 1\}^\beta)$
\[ \mathcal{E}(x, c) = (f(x), c) \text{ is a OWF} \]
and
\[ h_\mathcal{E}(x, c) = \Theta(x, \mathcal{E}(x, c)) \text{ with } 1 \text{ bit of seed } \mathcal{E}(x) \]

Composing PRG from hardware predicates.

\[ s \in D_2 \text{ a one} \]

\[ \begin{align*}
  & s \\
  \xrightarrow{\epsilon = \mathcal{E}(s)} & h_\mathcal{E}(s) \\
  \xrightarrow{\mathcal{E}} & \epsilon \mathcal{E}(s) \\
  \xrightarrow{s_0} & b_0 \\
  \xrightarrow{\mathcal{E}(s_0)} & \epsilon \mathcal{E}(s_0) \\
  \end{align*} \]
Then \( f_2 : D_2 \rightarrow D_2 \) is one

Then \( f_3(s) = (b_1, \ldots, b_9) \) is a PRG

Composing PRG into PRF

Let \( f : \{0, 1\}^{2^m} \rightarrow \{0, 1\}^{2^m} \) be a length-doubling PRG

Let \( f_L(s) = \) first \( m \) bits of \( f(s) \)

Let \( f_R(s) = \) second \( m \) bits of \( f(s) \)

Naive approach:

\[
\sum'(k, x) = \underbrace{f_R(f_L(\ldots f_L(K))))}_{x \text{ times}} = f_R(f_L(x))(k)
\]

Works, but is expensive.

Idea: Tree construction:

\[
s_0 = f_L(s) \\
s_1 = f_R(f_L(s_0)) \\
\vdots \\
s_h = f_R(f_L(\ldots f_L(s_0))))
\]

Each leaf is an output of the PRF, \( x \) denotes the path through the tree, and \( K \) is the seed \( s_0 \).