Pseudorandomness

Thursday, September 21, 2017 12:11 PM

The story of founding cryptography on One Way Functions

Universal owf

$$OWF \rightarrow PRG \rightarrow PRF \rightarrow \text{symmetric encryption}$$

PRF: pseudorandom function

What do we do with PRFs?

$k \xleftarrow{\$} K_\lambda$ ← key space

Alice k, m_i —— $m_i \oplus f(k, i)$ ——> Bob k

f is a PRF

6

Is this a random number?

7654338

2cf24dba5fb0a30e26e83b2ac5b9e29e1b161e5c1fa7425e73043362938b9824

— One way functions: (OWF): "hard to invert"

A family of functions $\{f_\lambda : D_\lambda \rightarrow C_\lambda\}$ is one way,

iff $\forall A$, $\Pr[x \xleftarrow{\$} D_\lambda, y = f_\lambda(x), x' \leftarrow A(1^\lambda, y) : x = x'] \leq \text{negl}(\lambda)$

— Pseudorandom Generator (PRG): (if $|D_\lambda| = \lambda$, this is length-doubling PRG)

A family $f_\lambda : D_\lambda \rightarrow \{0,1\}^{2\lambda}$ is a PRG

iff $\{x \xleftarrow{\$} D_\lambda : f_\lambda(x)\} \approx_c \{x \xleftarrow{\$} \{0,1\}^{2\lambda} : x\}$

"Indistinguishable from random sample, given a random seed."

— Pseudorandom Function (PRF):

$f_\lambda : (K_\lambda \times D_\lambda) \rightarrow \{0,1\}^\lambda$ is a PRF iff

$\forall A, \{k \xleftarrow{\$} K_\lambda : A^{f_\lambda(k,\cdot)}(1^\lambda)\} \approx_c \{f' \xleftarrow{\$} S_\lambda : A^{f'}(1^\lambda)\}$

A gets oracle access to $f_\lambda(k, \cdot)$, can query the function with k fixed

"Indistinguishable from a random function, even if adversary chooses the inputs to the function."

— One Way Functions

"hard to invert" ← on random input

Examples:

— $\otimes G$ $\quad p = |G_\lambda|$ $\quad$ ← groups of size $|G| \approx 2^\lambda$

$f_\lambda : \mathbb{Z}_p \to G_\lambda$

$f(x) = g^x$

- Cryptographic hash functions (SHA)
- Universal one-way function

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

  What does it look like?

  Weird Turing machines trick...
- RSA function

~~Let's~~ Use OWF for encryption!

$m \oplus \mathrm{ser}(g^k)$ ← $f(x) = g^k$

$|m| = |\mathrm{ser}(g^k)|$

and only use once

Why not? Not all bits are hard to predict.

Suppose f_λ is a PRG.

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

Is f'_λ a PRG? yes ✓

Does PRG imply OWF? ✓

$\boxed{\text{To fix later}}$

If $f : \{0,1\}^\lambda \to \{0,1\}^{2\lambda}$ is PRG

is it also OWF?

Given A that wins the OWF game,

construct A' that wins PRG

$A'(y)$: y comes from PRG or x comes from $\xleftarrow{\$} \{0,1\}^{2\lambda}$

$x = A(y)$

output 0 if $y \neq f(x)$

output 1 otherwise

$D_1 \approx D_2$

$$\left| \Pr[A(x \xleftarrow{\$} D_1) = 1] - \Pr[A(x \xleftarrow{\$} D_2) = 1] \right| \leq \mathrm{negl}$$

Why? Codomain of f is much larger than domain. $\{0,1\}^{2\lambda}$ vs $\{0,1\}^\lambda$. If $y \xleftarrow{\$} \{0,1\}^{2\lambda}$, $\Pr[\exists x, f(x) = y] \leq 2^{-\lambda}$.

So, if A can find such a value x, it is because y was chosen from image of f, not from $\{0,1\}^{2\lambda}$.

: DLOG in Schnorr group

$\text{half}(g^x)$ $\begin{cases} 1 & \text{if } g^x \geq \frac{|G|}{2} \\ 0 & \text{otherwise} \end{cases}$

2.8 DLOG collection

3.4.1 "a hardcore bit from DLOG"
based on sqrt.

open question: does this hold for positive $(g^x).y$ in Secp256k1.

$f: D_\lambda \to C_\lambda$

Hardcore predicate for a OWF f, $h_f : D_\lambda \to \{0,1\}$

"adversary can't predict $h_f(x)$ even after seeing $f(x)$."

$$\forall A \; \Pr[x \xleftarrow{\$} D_\lambda : A(f(x)) = h(x)] \leq \frac{1}{2} + \text{negl}(\lambda)$$

- $\text{half}(g^x)$ for Schnorr groups
- LSB of RSA
- secp256k1?

Universal Hardcore Predicate "Goldreich-Levin" HCP 3.4.2

Let $f_\lambda : D_\lambda \to D_\lambda$ be a OWF $|\text{ser}(D_\lambda)| = \{0,1\}^\lambda$

Then let $f' : (D_\lambda \times \{0,1\}^\lambda) \to (D_\lambda \times \{0,1\}^\lambda)$

$f'(x, r) = (f(x), r)$ is a owf

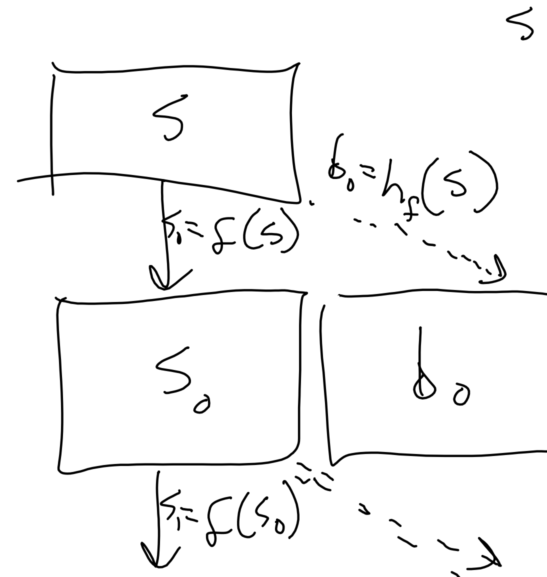
and

$$h((x,r)) = \bigoplus_i (x_i \wedge r_i)$$

$x_i = \text{ser}(x)[i]$
ith bit of serialized x

Composing PRG from hardcore predicates.

$s \xleftarrow{\$} D_\lambda$ assume $f: D_\lambda \to D_\lambda$ is a OWF



λ | $b_1, \ldots$

Thm. $f_\lambda : D_\lambda \to D_\lambda$ is OWF $\quad f' : D_\lambda \to \{0,1\}^{2\lambda}$

then $f'_\lambda(s) = \{b_0, b_1, \ldots, b_{2\lambda}\}$ is a PRG

Composing PRG into PRF

Let $f : \{0,1\}^\lambda \to \{0,1\}^{2\lambda}$ be a length-doubling PRG

Let $f_L(s)$ = first λ bits of $f(s)$

$f_R(s)$ = second λ bits of $f(s)$

Naïve approach:

$$f'(k, x) = \underbrace{f_R(f_L(f_L(\ldots f_L(K))))}_{x \text{ times.}} = f_R\left(f_L^{(x-1)}(K)\right)$$

Works, but is expensive.

Idea: Tree construction:

s

$s_0 = f_L(s)$ $\qquad$ $s_1 = f_R(s)$ $\qquad$ height $= \lambda$

$\ldots$

s_{00} $\qquad$ $s_{01} = f_R(f_L(s_0))$

$\ldots$

$s_{010100110} = f_L(f_R(f_L(f_R(f_L(f_L(f_R(f_R(f_L(s)))))))))$

Each leaf is an output of the PRF, x denotes the path through the tree, and k is the seed (s)