On one-way functions and the average time complexity of almost-optimal compression

Marius Zimand

June 18, 2025

CCR 2025, Bordeaux

Theorem The following are equivalent:

(1) $\exists$ a one-way function

(2) almost-optimal compression is hard on D-average, for some distribution D, which is efficiently-samplable

The result follows by combining

Ilango, Ren, Santhanam, STOC 2022
+
Bauwens, Zimand, J ACM 2023

OWF

$$x \underset{\text{HARD}}{\overset{\text{EASY}}{\rightleftarrows}} f(x)$$

- Basic primitive in cryptography.

OWF

-private -key encryption
-zero-knowledge proofs
-digital signatures
-commitment schemes

$\vdots$

public-key encryption NO

- Do OWF exist? No, if $P = NP$

FACTORING, DISCRETE LOG, SHORTEST VECTOR PROBLEM

↓

believed to be OWF

Liu, Pass (2020)

OWF exist $\Longleftrightarrow$ $K^{\text{poly-time}}$ is weakly hard on average (over uniform distribution)

K^t problem: Given x, time-bound t

Find $K^t(x)$ (time-bounded Kolmogorov complexity)

weakly hard ... : Any PPT algorithm fails on $\geq \frac{1}{\text{poly}(n)}$ fraction of $x \in \{0,1\}^n$

"PPT": probabilistic polynomial time

After [Liu, Pass 2020] many other results of the type:

OWF exist $\Longleftrightarrow$ some problem for time-bounded Kolm. complexity is hard in some sense

[Ilango, Ren, Santhanam, 2022]

OWF exist $\Longleftrightarrow$ some problem for Kolm. complexity is hard

unbounded version

Kolmogorov complexity

Fix U, universal Turing machine
with prefix-free domain

U maps input $p \in \{0,1\}^*$ into output $x \in \{0,1\}^*$
or $\perp$
" $U(p)$ undefined "

Domain of U : prefix-free set

$$K(x) = \min \{ \underbrace{|p|}_{\text{optimal description (or program) for } x} \mid U(p) = x \}$$

DISTRIBUTIONS

- Ensemble $D = (D_n)_{n \in \mathbb{N}}$,

  D_n distrib. over $\{0,1\}^n$

- D is samplable if $\exists$ algorithm SAMP

  $\forall n$, $\forall x \in \{0,1\}^n$

  $$\text{Prob}[\,\text{SAMP}(1^n) = x\,] = D_n(x).$$

- If SAMP is PPT, D is poly-time samplable

THEOREM [IRS 2022] The following are equivalent:

(1) $\exists$ OWF

(2) K is hard to approximate on average for some poly-time samplable distribution

$\exists$ OWF

$\exists f : \{0,1\}^* \longrightarrow \{0,1\}^*$, poly-time computable, s.t.

$\exists q \in \mathbb{N}$, $\forall$ PPT alg. INVERTER, a.e n

$$\Pr_{\substack{x \leftarrow U_n \\ \text{rand. of INVERTER}}}\left[\text{INVERTER}(1^n, f(x)) \notin f^{-1}(f(x)) \right] \geq \frac{1}{n^q}$$

Prob [fail] $\geq 1/n^q$

can be replaced by $1 - \text{neg}(n)$

K is hard to approx. on avg. for some effic. sampl. distribution

$\exists$ poly-time samplable distribution $D = (D_n)_{n \in \mathbb{N}}$

$\forall c$, $\forall$ PPT alg. ESTIMATOR $\forall_{a.e}$ n

$$\Pr_{\substack{x \leftarrow D_n \\ \text{rand. of ESTIMATOR}}}\left[\text{ESTIMATOR}(x) \notin (K(x) - c \cdot \log n, K(x) + c \cdot \log n) \right] > \frac{1}{100}$$

Prob [fail] $\geq 1/100$

THEOREM 1 [IRS 2022] The following are equivalent

(1) $\exists$ OWF

(2) K is hard to approximate on average for some p-time samplable distribution

THEOREM 2 [Bauwens, Zimand]

Finding $m \simeq K(x)$ $\xLeftrightarrow[\text{p-time reductions}]{}$ Finding in PPT an almost optimal program for x

$\exists$ PPT alg. COMPRESSOR

- $\forall (x, m, \varepsilon)$ returns a string z, $|z| \le m + O(\log m \cdot \log \frac{n}{\varepsilon})$
- If $m \geq K(x)$, $\text{Prob}[z \text{ program for } x] \geq 1 - \varepsilon$.

Th.1 + Th.2

THEOREM. The following are equivalent

(1) $\exists$ OWF

(2) Almost optimal compression is hard on average for some p-time samplable distribution

Given x: find a program p for x of length $K(x) + O(\log^2 n)$

KOLMOGOROV complexity VS. SAMPLABLE DISTRIBUTIONS

LEMMA 1. If D is samplable

$$\operatorname*{Prob}_{x \leftarrow D_n}\left[K(x) \leq \log \frac{1}{D(x)} + 4 \log n \right] \geq 1 - 2^{-n}$$

Proof: Coding ...

LEMMA 2: For every distrib. D,

$$\operatorname*{Prob}_{x \leftarrow D_n}\left[K(x) \geq \log \frac{1}{D(x)} - 4 \log n \right] \geq 1 - \frac{1}{n^4}$$

Proof: Kraft - Chaitin ineq. ...

So: For every samplable D

Estimating $K(x)$ with high D-prob. $\iff$ Estimating $D(x)$. with high D-prob.

THEOREM The following are equivalent:

(1) $\exists$ OWF

(2) $\exists$ poly-time sampl. D which is hard to approximate on D-average

Proof by showing contrapositives.

I $\forall$ p-time f has INVERTER $\Rightarrow$ $\forall$ p-time sampl. D has ESTIMATOR

$\forall\, poly(n)$, $\exists$ PPT alg. INVERTER, i.o n

$$\Pr_{x \leftarrow U_n}\left[\text{INVERTER}(x) \in f^{-1}(x)\right] \geq 1 - \frac{1}{poly(n)}$$

$\forall\, poly(n)$, $\exists$ PPT alg. ESTIMATOR, i.o. n

$$\Pr_{x \leftarrow D}\left[\text{ESTIMATOR}(x) \in \left[\frac{D(x)}{256}, D(x)\right]\right] \geq 1 - \frac{1}{poly(n)}$$

II $\forall$ p-time sampl. D has ESTIMATOR $\Rightarrow$ $\forall$ poly-time f has INVERTER

I $\forall$ poly-time f has INVERTER $\Rightarrow$ $\forall$ p-time samplable D has ESTIMATOR

<u>Proof.</u> Take $D = (D_n)$, p-time samplable

$\text{Prob}[\text{SAMP}(1^n) = y] = D(y)$

$\uparrow$ PPT alg., uses m random bits, $m = \text{poly}(n)$

- Using INVERTER, given y, we can find an element of $\text{SAMP}^{-1}(y)$
- But we want to estimate $\#\,\text{SAMP}^{-1}(y)$

- **SOLUTION A:**

  use pairwise-indep-hashing +
  Left over hash lemma

- **SOLUTION B:**

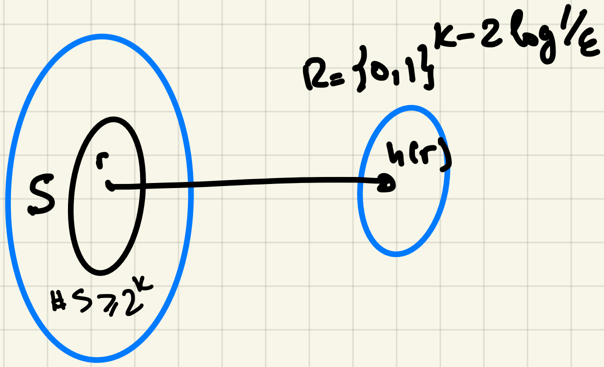
  use "Short lists for short programs in short time"

• SOLUTION A:

Left-over hash lemma

$$H_{m,k} = \{ h : \{0,1\}^m \to \{0,1\}^{k-2\log 1/\varepsilon} \}$$

family of 2-wise indep. hash functions



$$(H, H(U_S)) \approx_\varepsilon (H, U_R)$$

For every $S \subseteq \{0,1\}^m$ with $\# S \geq 2^k$

$$\Big(\underset{\substack{\uparrow U \\ H_{m,k}}}{h},\ h(\underset{\substack{\uparrow U \\ S}}{r})\Big) \approx_\varepsilon \Big(\underset{\substack{\uparrow U \\ H_{m,k}}}{h},\ \underset{\substack{\uparrow U \\ \{0,1\}^k}}{v}\Big)$$

Recall STATISTICAL DISTANCE between distributions:

$$D_1 \approx_\varepsilon D_2 \ : \ \forall \text{ event } A,\ |D_1(A) - D_2(A)| \leq \varepsilon$$

Function f :

$$\boxed{\begin{array}{l} \text{input: } (n, k, h, r) \qquad h \in H_{n,k} \\ y = SAMP(1^n, r) \qquad r \in \{0,1\}^m \\ \text{output } (n, y, k, h, h(r)) \end{array}}$$

INVERTER on input $(n, y, k, h, h(r))$

has to find $r' \in SAMP^{-1}(y)$

$h(r') = h(r)$

Function f:

$$\boxed{\begin{aligned} &\text{input: } (n, k, h, r) \quad h \in H_{m,k},\ r \in \{0,1\}^m \\ &y = \mathrm{SAMP}(1^n, r) \\ &\text{output } (n, k, y, h, h(r)) \end{aligned}}$$

f is poly.-time computable

Assumption $\Rightarrow \exists$ PPT INVERTER, i.o. n

$$\Pr_{\substack{k \leftarrow_U [m] \\ h \leftarrow_U H_{m,k} \\ r \leftarrow_U \{0,1\}^m \\ z = f(n,k,h,r) = (n,k,y,h,h(r))}} \left[\mathrm{INVERTER}(z) \in f^{-1}(z) \right] \geq 1 - \frac{1}{m \cdot n}$$

Given y, we want to estimate $\#\mathrm{SAMP}^{-1}(y)$

(because $D(y) = \#\mathrm{SAMP}^{-1}(y) / 2^m$)

Given y, we want to estimate $\# SAMP^{-1}(y)$

For each $k \in [m]$ (2^k is our guess for $\# SAMP^{-1}(y)$)

we do $E(k, y)$:

- choose random $h \in H_{m,k}$, $v \in \{0,1\}^k$
- if INVERTER finds a pre-image of (n, k, y, h, v), return 1, else 0.

so, finds $r \in SAMP^{-1}(y)$ & $h(r) = v$

ISSUE: INVERTER SUCCEEDS when $v = h(r)$

(1) If k large (meaning: $2^k > 4 \cdot \# SAMP^{-1}(y)$)

$\text{Prob}[E(k,y) = 1]$ is small

Why: Most v in $\{0,1\}^k$ are not hashes of elemts. in $SAMP^{-1}(y)$

(2) If k small (meaning: $2^k \le \frac{1}{64} \# SAMP^{-1}(y)$)

$\text{Prob}[E(k,y) = 1]$ is large

Why: $(h, v) \sim_{\varepsilon} (h, h(r))$

(1) + (2): we can approximate $\# SAMP^{-1}(y)$ in PPT.

k large

<u>Case 1</u> If $2^k > 4 \cdot \# \text{SAMP}^{-1}(y)$

then $\text{Prob}[E(k,y) = 1] \leq \frac{1}{4}$

For every h, $\# \text{SAMP}^{-1}(y) \geq h(\# \text{SAMP}^{-1}(y))$

So: $\# h(\text{SAMP}^{-1}(y)) \leq \frac{1}{4} \cdot 2^k$

So: $\frac{3}{4}$ fraction of $v \in \{0,1\}^k$
cannot be in $h(\text{SAMP}^{-1}(y))$

So: Prob of success $\leq 1/4$.

k small

<u>Case 2</u> If $2^k \leq \frac{1}{64} \cdot \# \, SAMP^{-1}(y)$

then $\text{Prob}[E(k,y)=1] \geq \frac{3}{4} - O(1/n^2)$

Consider the following 2 distributions:

D_1: sample $h \leftarrow H_{m,k}$, $v \leftarrow \{0,1\}^k$

output (n, k, y, h, v)

D_2: sample $h \leftarrow H_{m,k}$, $r \leftarrow SAMP^{-1}(y)$

output $(n, k, y, h, h(r))$

Leftover hash lemma: statistical dist $(D_1, D_2) \leq \sqrt{\frac{2^k}{\# \, SAMP^{-1}(y)}}$

$\leq 1/8$ (Case 2)

$H_{m,k}$ 2-wise indep. family

$$\underset{D_1}{\mathrm{Prob}}[E(k,y)=0] \le \frac{1}{8} + \underset{D_2}{\mathrm{Prob}}[\mathrm{INV.}\ (n, y, k, h, h(r))\ \text{fails}]$$

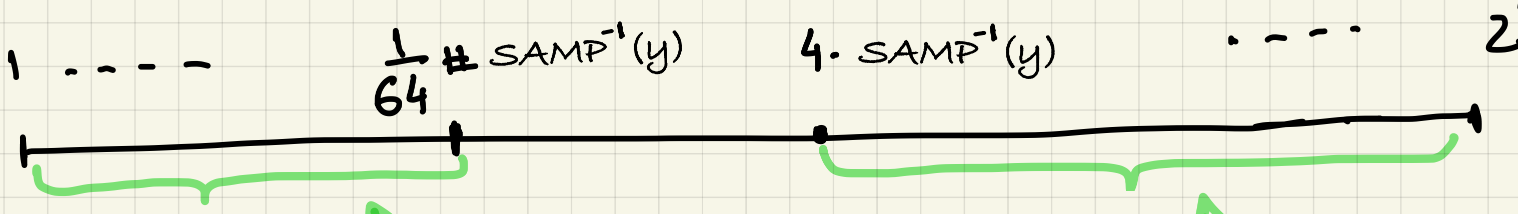
y is good if this $\le \frac{1}{8}$ for all k

$$\frac{1}{m\cdot n} \ge \underset{D_2}{\Pr}[\mathrm{INV.}\ \text{fails}] \ge \sum_{k=1}^{m} \Pr[y \text{ bad for } k]\cdot\frac{1}{8}\cdot\frac{1}{m}$$

$$\Rightarrow \Pr[y \text{ bad}] \le 8/n \quad \Rightarrow \Pr[y \text{ is good}] \ge 1 - \frac{8}{n}$$

Conditioned on "y good", for every k

$$\mathrm{Prob}[E(k,y)=0] \le 1/8 + 1/8 = 1/4$$



If 2^k here
$E(k,y)=1$
w. prob $\geq \frac{3}{4} - \frac{8}{n}$

If 2^k here
$E(k,y)=1$ w. prob $\leq 1/4$

For $k = 1, 2, \ldots, m$

Repeat $E(k,y)$ m^2 times

Output largest k for which $\#$"successes" $\geq \frac{m^2}{2}$

"success" when $E(k,y)=1$

With high prob:

$$\frac{1}{64} \cdot \# SAMP^{-1}(y) \leq 2^k \leq 4 \cdot \# SAMP^{-1}(y)$$

SOLUTION B:

use "Short lists for short programs in short time"

THEOREM (Bauwens-Makhlin-Vereshchagin-Z. 2012, Teutsch 2012, Z 2013)

There exists poly-time alg. A and const. c_{LIST}

- on input n-bit x, A prints LIST(x) with n^7 strings
- $\forall y$, LIST(x) contains a program p of x given y with

$$|p| \leq C(x|y) + c_{LIST}$$

- for every $\ell \in [|x| + c_{UNIV.KOLM}]$,
  LIST(x) has the same number of elements of length ℓ

Function f

$$\boxed{\begin{array}{l} \text{input: } (n, r, \ell) \\ y = \mathrm{SAMP}(1^n, r) \\ \text{output } (n, y, \ell, \mathrm{LIST}(r)_\ell) \end{array}}$$

f is poly-time computable

Assumption $\Rightarrow$ $\exists$ PPT INVERTER, s.t. for i.o. n:

$$\Pr_{\substack{r \leftarrow_U \{0,1\}^m \\ \ell \leftarrow_U [m] \\ y = \mathrm{SAMP}(1^n, r) \\ z = (n, y, \ell, \mathrm{LIST}(r)_\ell)}} \left[\mathrm{INVERTER}(z) \in f^{-1}(z) \right] \geq 1 - \frac{1}{m \cdot n}$$

Given y, we want to estimate $\# \mathrm{SAMP}^{-1}(y)$

(because $D(y) = \# \mathrm{SAMP}^{-1}(y) / 2^m$)

PROCEDURE: On input y, estimate $\# SAMP^{-1}(y)$

For each $k \in [m]$ (2^k is our guess for $\# SAMP^{-1}(y)$)

we do $E(k, y)$:

- choose random $\ell \in [m^7]$, $v \in \{0,1\}^{\le k}$
- if INVERTER finds a pre-image of (n, y, l, v), return 1, else 0.

↓ so, finds $r \in SAMP^{-1}(y)$ & $LIST(r)_\ell = v$

ISSUE: INVERTER succeeds when $v = LIST(r)_\ell$

(1) If k large (meaning: $2^{k+1} > C_1 \cdot m^7 \cdot \# SAMP^{-1}(y)$)

Prob $[E(k,y) = 1]$ is small $\left(< \frac{1}{C_1} \right)$

Why: Most v in $\{0,1\}^{\le k}$ are not in $\bigcup_{r \in SAMP^{-1}(y)} LIST(r)$

(2) If k small (meaning: $2^{k+1} = 2^{c_{u.k} + c_L} \cdot \# SAMP^{-1}(y)$)

Prob $[E[k,y] = 1]$ is large $\left(> \frac{1}{2^{c_{u.k} + c_L + 1}} \right)$

Why: NEXT SLIDE

(1) + (2): we can approximate $\# SAMP^{-1}(y)$ in PPT.

k is small: $k = \log(\#\,\mathrm{SAMP}^{-1}(y)) + c_{\mathrm{UNIV.KOLM}} + c_{\mathrm{LIST}} + 1$

For every $r \in \mathrm{SAMP}^{-1}(y)$:

- $C(r \mid y) \leq \log(\#\,\mathrm{SAMP}^{-1}(y)) + c_{\mathrm{UNIV.KOLM}}$
- $\exists\, r^*$ in $\mathrm{LIST}(r)$, program for r with
  $|r^*| \leq \log(\#\,\mathrm{SAMP}^{-1}(y)) + c_{\mathrm{UNIV.KOLM}} + c_{\mathrm{LIST}} = k$

So: $\{v \mid v \in \bigcup_{r \in \mathrm{SAMP}^{-1}(y)} \mathrm{LIST}(r),\ |v| \leq k\} \supseteq \{r^* \mid r \in \mathrm{SAMP}^{-1}(y)\}$

size $= \#\,\mathrm{SAMP}^{-1}(y)$

$= \frac{1}{2^{c_{u.k} + c_L}} \cdot \#\{0,1\}^{\leq k}$

INVERTER fails on this set with prob. $\leq \frac{1}{n}$

NEXT SLIDE

So, succeeds w. prob $\geq 1 - \frac{1}{n}$

So: INVERTER succeeds on $\{0,1\}^{\leq k}$ w. prob. $\geq \frac{1}{2^{c_{u.k} + c_L + 1}}$

$\mathrm{Prob}[E(k,y) = 1]$

EVALUATION of FAIL probability

ASSUMPTION:

$$\frac{1}{m \cdot n} > \text{Prob}\left[\text{INVERTER}\left(n, y, \ell, \text{LIST}(r)_\ell\right) \text{ fails}\right]$$

$$\geq \text{Prob}\left[\text{INVERTER}(\ldots) \text{ fails} \;\middle|\; |\text{LIST}(r)_\ell| \leq k\right] \cdot \underbrace{\text{Prob}\left[|\text{LIST}(r)_\ell| \leq k\right]}_{= \frac{k}{m + C_{\text{UNIV. KOLM}}}}$$

So: $\text{Prob}\left[\text{INVERTER}(\ldots) \text{ fails} \;\middle|\; |\text{LIST}(r)_\ell| \leq k\right]$

$$\leq \frac{m + C_{U.K}}{k} \cdot \frac{1}{m \cdot n} \leq \frac{1}{n}$$

COMPARING SOLUTION A and SOLUTION B

SOLUTION A - Left-over hash Lemma

$$(n, k, r, h) \xrightarrow{f} (n, k, \mathrm{SAMP}(r), h, h(r))$$

length $\simeq 2m$

SOLUTION B - Short lists w. short programs

$$(n, r, \ell) \xrightarrow{f} (n, \mathrm{SAMP}(r), \ell, \mathrm{LIST}(r)_\ell)$$

length $\simeq m$

DONE (with 2 proofs)

I $\forall$ poly-time f has INVERTER $\Rightarrow$ $\forall$ p-time sampl. D has ESTIMATOR

II $\forall$ poly-sampl. D has ESTIMATOR $\Rightarrow$ $\forall$ poly-time f has INVERTER

NEXT

Just a sketch

Π $\forall$ poly-sampl. D has ESTIMATOR =>
$\forall$ poly-time f has INVERTER

Proof (sketch)

Suppose $\exists f$, poly-time, has no INVERTER

So f is OWF

[HILL'99]
$\Rightarrow$ $\exists$ p.r.g $G: \{0,1\}^{n^{1/3}} \longrightarrow \{0,1\}^n$

$\Downarrow$

no PPT alg. A can distinguish $(G(U_{n^{1/3}}), U_n)$

$|\Pr[A(G(U_{n^{1/3}}))=1] - \Pr[A(U_n)=1]| < \frac{1}{n}$

Distribution D: $\frac{1}{2} G(U_{n^{1/3}}) + \frac{1}{2} U_n$

$G(U_{n^{1/3}})$ has Kolm. complexity $\leq n^{1/3}$ w.prob$_{U_{n^{1/3}}}$ $=1$

so $D(G(U_{n^{1/3}}))$ is large

U_n has Kolm. complexity $\geqslant n - \log n$ w.prob$_{U_n}$ $1-\frac{1}{n}$

so $D(U_n)$ is small

Distribution D is poly-time samplable

$\Downarrow$

has poly-time ESTIMATOR

$\Downarrow$

Distinguisher breaking p.r.g. $\times$