Non-Adaptive Universal One-Way Hash Functions from Arbitrary One-Way Functions

Xinyu Mao* Noam Mazor** Jiapeng Zhang*

April 26, 2023

* University of Southern California

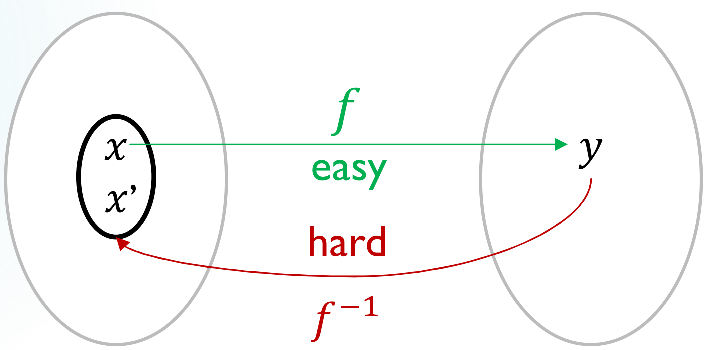
** Tel-Aviv University

One-Way Functions

- A function $f\colon \{0,1\}^n \to \{0,1\}^n$ is **one-way function** if:
  - Easy to compute: f is computable in poly(n) time.
  - Hard to invert: $\forall$ PPT A

$$\Pr_{x \leftarrow \{0,1\}^n}\left[A(f(x)) \in f^{-1}\big(f(x)\big)\right] = \mathrm{negl}(n).$$

- OWF exists: "minimal assumption for cryptography"



Universal One-Way Hash Functions (UOWHFs) [Naor-Yung' 89]

UOWHF (also known as target collision-resistant hash function)

- A keyed hash family $C_z: \{0,1\}^m \to \{0,1\}^\ell, z \in \{0,1\}^k$
- Shrinking: $\ell < m$.
- Target collision resistance: $\forall$ PPT $A = (A_1, A_2)$

$$\Pr_{(x,st)\leftarrow A_1,\, z\leftarrow\{0,1\}^k}[A_2(x,z,st) = x' \text{ s.t. } C_z(x) = C_z(x')] \text{ is negligible.}$$

Universal One-Way Hash Functions (UOWHFs) [Naor-Yung' 89]

UOWHF (also known as target collision-resistant hash function)

- A keyed hash family $C_z: \{0,1\}^m \to \{0,1\}^\ell, z \in \{0,1\}^k$
- Shrinking: $\ell < m$.
- Target collision resistance: $\forall$ PPT $A = (A_1, A_2)$

$$\Pr_{(x,st)\leftarrow A_1,\, z\leftarrow\{0,1\}^k}[A_2(x,z,st) = x' \text{ s.t. } C_z(x) = C_z(x')] \text{ is negligible.}$$

- One-way function + UOWHF → digital signature [Naor-Yung' 89]

Universal One-Way Hash Functions (UOWHFs) [Naor-Yung' 89]

UOWHF (also known as target collision-resistant hash function)

- A keyed hash family $C_z: \{0,1\}^m \to \{0,1\}^\ell, z \in \{0,1\}^k$
- Shrinking: $\ell < m$.
- Target collision resistance: $\forall$ PPT $A = (A_1, A_2)$
  $$\Pr_{(x,st)\leftarrow A_1,\, z\leftarrow\{0,1\}^k}[A_2(x,z,st) = x' \text{ s.t. } C_z(x) = C_z(x')] \text{ is negligible.}$$

- One-way function + UOWHF → digital signature [Naor-Yung' 89]
- One-way function → UOWHF [Rompel' 90]

Universal One-Way Hash Functions (UOWHFs) [Naor-Yung' 89]

UOWHF (also known as target collision-resistant hash function)

- A keyed hash family $C_z: \{0,1\}^m \to \{0,1\}^\ell, z \in \{0,1\}^k$
- Shrinking: $\ell < m$.
- Target collision resistance: $\forall$ PPT $A = (A_1, A_2)$
$$\Pr_{(x,st)\leftarrow A_1,\, z\leftarrow\{0,1\}^k}[A_2(x,z,st) = x' \text{ s.t. } C_z(x) = C_z(x')] \text{ is negligible.}$$

- One-way function + UOWHF → digital signature [Naor-Yung' 89]
- One-way function → UOWHF [Rompel' 90]
- UOWHF can be easily constructed from a unkeyed function F that is shrinking and collision-resistant on random inputs.

Universal One-Way Hash Functions (UOWHFs) [Naor-Yung' 89]

UOWHF (also known as target collision-resistant hash function)

- A keyed hash family $C_z: \{0,1\}^m \to \{0,1\}^\ell, z \in \{0,1\}^k$
- Shrinking: $\ell < m$.
- Target collision resistance: $\forall$ PPT $A = (A_1, A_2)$

$$\Pr_{(x,st)\leftarrow A_1,\, z\leftarrow\{0,1\}^k}[A_2(x,z,st) = x' \text{ s.t. } C_z(x) = C_z(x')] \text{ is negligible.}$$

- One-way function + UOWHF → digital signature [Naor-Yung' 89]
- One-way function → UOWHF [Rompel' 90]
- UOWHF can be easily constructed from a unkeyed function F that is shrinking and collision-resistant on random inputs.

Given random $x \leftarrow \{0,1\}^m$, it is hard to find x' such that $F(x) = F(x')$.

Universal One-Way Hash Functions (UOWHFs) [Naor-Yung' 89]

UOWHF (also known as target collision-resistant hash function)

- A keyed hash family $C_z: \{0,1\}^m \to \{0,1\}^\ell, z \in \{0,1\}^k$
- Shrinking: $\ell < m$.
- Target collision resistance: $\forall$ PPT $A = (A_1, A_2)$

$$\Pr_{(x,st)\leftarrow A_1,\, z\leftarrow\{0,1\}^k}[A_2(x,z,st) = x' \text{ s.t. } C_z(x) = C_z(x')] \text{ is negligible.}$$

- One-way function + UOWHF → digital signature [Naor-Yung' 89]
- One-way function → UOWHF [Rompel' 90]
- UOWHF can be easily constructed from a unkeyed function F that is shrinking and collision-resistant on random inputs.

Given random $x \leftarrow \{0,1\}^m$, it is hard to find x' such that $F(x) = F(x')$.

Construction:

$$C_z(x) \coloneqq F(z \oplus x)$$

The efficiency of OWF → UOWHF constructions

OWF
$f: \{0,1\}^n \to \{0,1\}^n$

UOWHF
$C_z: \{0,1\}^{m(n)} \to \{0,1\}^{\ell(n)}, z \in \{0,1\}^{k(n)}$

Efficiency Measures

- Seed length: $k(n)$
- Number of calls to the underlying OWF
- Adaptivity: whether the invocations of the OWF are dependent of the output of previous calls

The efficiency of OWF → UOWHF constructions

OWF
$f:\{0,1\}^n \to \{0,1\}^n$

UOWHF
$C_z:\{0,1\}^{m(n)} \to \{0,1\}^{\ell(n)}, z \in \{0,1\}^{k(n)}$

Efficiency Measures

- Seed length: $k(n)$
- Number of calls to the underlying OWF
- Adaptivity: whether the invocations of the OWF are dependent of the output of previous calls

	Seed length	**Number of calls**	**Non-adaptive?**
[HHRVW' 10]	$\tilde{O}(n^5 \log n)$	$\tilde{O}(n^{13})$	×
Our Construction I	$\tilde{O}(n^9 \log n)$	$\tilde{O}(n^{10})$	√

The efficiency of OWF → UOWHF constructions

OWF
$f\colon\{0,1\}^n \to \{0,1\}^n$

→

UOWHF
$C_z\colon\{0,1\}^{m(n)} \to \{0,1\}^{\ell(n)}, z \in \{0,1\}^{k(n)}$

Efficiency Measures

- Seed length: $k(n)$
- Number of calls to the underlying OWF
- Adaptivity: whether the invocations of the OWF are dependent of the output of previous calls

	Seed length	**Number of calls**	**Non-adaptive?**
[HHRVW' 10]	$\tilde{O}(n^5 \log n)$	$\tilde{O}(n^{13})$	×
Our Construction I	$\tilde{O}(n^9 \log n)$	$\tilde{O}(n^{10})$	√

- The first non-adaptive construction
- It can be implemented in $\mathbf{NC_1}$ with f-oracle gates
- Combined with [AIK' 06]→ Assuming that OWFs exist in $\mathbf{NC_1}$, there exists a UOWHF in $\mathbf{NC_0}$.

The efficiency of OWF → UOWHF constructions

OWF
$f: \{0,1\}^n \to \{0,1\}^n$

UOWHF
$C_z: \{0,1\}^{m(n)} \to \{0,1\}^{\ell(n)}, z \in \{0,1\}^{k(n)}$

Efficiency Measures

- Seed length: $k(n)$
- Number of calls to the underlying OWF
- Adaptivity: whether the invocations of the OWF are dependent of the output of previous calls

	Seed length	**Number of calls**	**Non-adaptive?**
[HHRVW' 10]	$\tilde{O}(n^5 \log n)$	$\tilde{O}(n^{13})$	×
Our Construction I	$\tilde{O}(n^9 \log n)$	$\tilde{O}(n^{10})$	√

- The first non-adaptive construction
- It can be implemented in $\mathbf{NC_1}$ with f-oracle gates
- Combined with [AIK' 06]→ Assuming that OWFs exist in $\mathbf{NC_1}$, there exists a UOWHF in $\mathbf{NC_0}$.

What does the '**right**' construction look like?

Similarity between OWF → PRG and OWF → UOWHFs

Regular OWF
$f\colon\{0,1\}^n \to \{0,1\}^n$

$\forall\, y, y' \in \mathrm{Image}(f), |f^{-1}(y)| = |f^{-1}(y')|$

[MZ' 22]

$$G\,(h, x_1, \dots, x_n) \coloneqq h(x_1, f\,(x_2)\,), h(x_2, f\,(x_3)\,), \dots, h(x_{n-1}, f\,(x_n)\,)$$

- ► $h\colon\{0,1\}^{2n} \to \{0,1\}^{n+\Delta}$ is a hash function from an appropriate hash family.
- ► Hashing out more bits: $\Delta = \log n$ → G is PRG.
- ► Hashing out fewer bits: $\Delta = -\log n$ → G' is collision-resistant on random inputs.

$$G'(h, x_1, \dots, x_n) \coloneqq f\,(x_1), G\,(h, x_1, \dots, x_t), x_n$$

The efficiency gap between OWF → PRG and OWF → UOWHFs

OWF $f:\{0,1\}^n \rightarrow \{0,1\}^n$

	Assumption	Seed Length		Number of Calls		Remarks
		PRG	UOWHF	PRG	UOWHF	
[HHR' 06] [AGV'12]	Regular OWF	$O(n)$	$O(n)$	$O(n)$	$O(n)$	Adaptive
[MZ'22]	Regular OWF	$O(n^2)$	$O(n^2)$	$O(n)$	$O(n)$	Non-adaptive
[VZ'12][HRV'10][HHRVW'10]	Arbitrary OWF	$O(n^4)$	$\tilde{O}(n^7)$	$O(n^3)$	$O(n^{13})$	Efficiency gap
Our Construction 1	Arbitrary OWF	-	$O(n^{10})$	-	$O(n^9)$	Non-adaptive

The efficiency gap between OWF → PRG and OWF → UOWHFs

OWF $f: \{0,1\}^n \to \{0,1\}^n$

	Assumption	**Seed Length**		**Number of Calls**		**Remarks**
		PRG	UOWHF	PRG	UOWHF	
[HHR' 06] [AGV'12]	Regular OWF	$O(n)$	$O(n)$	$O(n)$	$O(n)$	Adaptive
[MZ'22]	Regular OWF	$O(n^2)$	$O(n^2)$	$O(n)$	$O(n)$	Non-adaptive
[VZ'12][HRV'10][HHRVW'10]	Arbitrary OWF	$O(n^4)$	$\tilde{O}(n^7)$	$O(n^3)$	$O(n^{13})$	Efficiency gap
Our Construction 1	Arbitrary OWF	-	$O(n^{10})$	-	$O(n^9)$	Non-adaptive

No efficiency gap between PRG and UOWHF if OWF is regular!

The efficiency gap between OWF → PRG and OWF → UOWHFs

Lower bound: $\widetilde{\Omega}(n)$ calls [HS' 12,16]

OWF $f: \{0,1\}^n \to \{0,1\}^n$

	Assumption	Seed Length		Number of Calls		Remarks
		PRG	UOWHF	PRG	UOWHF	
[HHR' 06] [AGV'12]	Regular OWF	$O(n)$	$O(n)$	$O(n)$	$O(n)$	Adaptive
[MZ'22]	Regular OWF	$O(n^2)$	$O(n^2)$	$O(n)$	$O(n)$	Non-adaptive
[VZ'12][HRV'10][HHRVW'10]	Arbitrary OWF	$O(n^4)$	$\tilde{O}(n^7)$	$O(n^3)$	$O(n^{13})$	Efficiency gap
Our Construction I	Arbitrary OWF	-	$O(n^{10})$	-	$O(n^9)$	Non-adaptive

No efficiency gap between PRG and UOWHF if OWF is regular!

The efficiency gap between OWF → PRG and OWF → UOWHFs

Lower bound: $\widetilde{\Omega}(n)$ calls [HS' 12,16]

OWF $f:\{0,1\}^n \to \{0,1\}^n$

	Assumption	Seed Length		Number of Calls		Remarks
		PRG	UOWHF	PRG	UOWHF	
[HHR' 06] [AGV'12]	Regular OWF	$O(n)$	$O(n)$	$O(n)$	$O(n)$	Adaptive
[MZ'22]	Regular OWF	$O(n^2)$	$O(n^2)$	$O(n)$	$O(n)$	Non-adaptive
[VZ'12][HRV'10][HHRVW'10]	Arbitrary OWF	$O(n^4)$	$\tilde{O}(n^7)$	$O(n^3)$	$O(n^{13})$	Efficiency gap
Our Construction I	Arbitrary OWF	-	$O(n^{10})$	-	$O(n^9)$	Non-adaptive
Our Almost-UOWHF	Arbitrary OWF	-	$\tilde{O}(n^4)$	-	$\tilde{O}(n^3)$	Non-adaptive Almost-UOWHF

No efficiency gap between PRG and UOWHF if OWF is regular!

The efficiency gap between OWF → PRG and OWF → UOWHFs

Lower bound: $\widetilde{\Omega}(n)$ calls [HS' 12,16]

OWF $f: \{0,1\}^n \to \{0,1\}^n$

	Assumption	Seed Length		Number of Calls		Remarks
		PRG	UOWHF	PRG	UOWHF	
[HHR' 06] [AGV'12]	Regular OWF	$O(n)$	$O(n)$	$O(n)$	$O(n)$	Adaptive
[MZ'22]	Regular OWF	$O(n^2)$	$O(n^2)$	$O(n)$	$O(n)$	Non-adaptive
[VZ'12][HRV'10][HHRVW'10]	Arbitrary OWF	$O(n^4)$	$\tilde{O}(n^7)$	$O(n^3)$	$O(n^{13})$	Efficiency gap
Our Construction I	Arbitrary OWF	-	$O(n^{10})$	-	$O(n^9)$	Non-adaptive
Our Almost-UOWHF	Arbitrary OWF	-	$\tilde{O}(n^4)$	-	$\tilde{O}(n^3)$	Non-adaptive Almost-UOWHF

No efficiency gap between PRG and UOWHF if OWF is regular!

Our Almost-UOWHF construction is very similar to HRV PRG construction. 🙂

Constructions

A Candidate UOWHF (the 'right' Construction)

Framework: computational entropy

Arbitrary OWF $f: \{0,1\}^n \to \{0,1\}^n$ → Computational entropy generator g → PRG, UOWHF, …

Manipulating entropy and extraction

▶ HRV PRG: $g(X)$ has next-bit pseudoentropy

▶ HRVVW UOWHF: $g(X)$ has inaccessible entropy

Write $Z := g(X) \in \{0,1\}^{\ell}$. $\exists\, Y = (Y_1, \dots, Y_\ell)$:

- $\forall\, i$: $Z_1, \dots, Z_i \approx_c Z_i, \dots, Z_{i-1}, Y_i$
- $\mathbb{E}_{I \leftarrow [\ell]}[\mathbf{H}(Y_I \mid Z_1, \dots, Z_{I-1})] \geq \frac{\mathbf{H}(Z)}{\ell} + \delta$.

($\mathbf{H}(\cdot)$: Shannon entropy)

That is, on average,

each bit exhibit δ extra pseudoentropy.

HRV PRG : repetition + random shift, drop unpopulated columns, hash more bits

A Candidate UOWHF (the 'right' Construction)

Framework: computational entropy

Arbitrary OWF $f: \{0,1\}^n \to \{0,1\}^n$ → Computational entropy generator g → PRG, UOWHF, …

Manipulating entropy and extraction

▶ HRV PRG: $g(X)$ has next-bit pseudoentropy

▶ HRVW UOWHF: $g(X)$ has inaccessible entropy

Write $Z := g(X) \in \{0,1\}^{\ell}$. $\exists\, Y = (Y_1, \dots, Y_\ell)$:

- $\forall\, i$: $Z_1, \dots, Z_i \approx_c Z_i, \dots, Z_{i-1}, Y_i$
- $\mathbb{E}_{I \leftarrow [\ell]}[\mathbf{H}(Y_I \mid Z_1, \dots, Z_{I-1})] \geq \frac{\mathbf{H}(Z)}{\ell} + \delta$.

($\mathbf{H}(\cdot)$: Shannon entropy)
That is, on average,
each bit exhibit δ extra pseudoentropy.

q rows

$g(X_{1,1})$	$g(X_{1,2})$	$\cdots$	$g(X_{1,t})$
$g(X_{2,1})$	$g(X_{2,2})$	$\cdots$	$g(X_{2,t})$
$\vdots$	$\vdots$	$\ddots$	$\vdots$
$g(X_{2,1})$	$g(X_{2,2})$	$\cdots$	$g(X_{2,t})$

$t \cdot \ell$ columns

HRV PRG : repetition + random shift,
drop unpopulated columns, hash more bits

A Candidate UOWHF (the ‘right’ Construction)

Framework: computational entropy

Arbitrary OWF $f: \{0,1\}^n \to \{0,1\}^n$ → Computational entropy generator g → PRG, UOWHF, …

Manipulating entropy and extraction

▶ HRV PRG: $g(X)$ has next-bit pseudoentropy

▶ HRVVW UOWHF: $g(X)$ has inaccessible entropy

Write $Z := g(X) \in \{0,1\}^\ell$. $\exists\, Y = (Y_1, \dots, Y_\ell)$:

- $\forall\, i$: $Z_1, \dots, Z_i \approx_c Z_i, \dots, Z_{i-1}, Y_i$
- $\mathbb{E}_{I \leftarrow [\ell]}[\mathbf{H}(Y_I \mid Z_1, \dots, Z_{I-1})] \geq \frac{\mathbf{H}(Z)}{\ell} + \delta$.

($\mathbf{H}(\cdot)$: Shannon entropy)

That is, on average, each bit exhibit δ extra pseudoentropy.

HRV PRG : repetition + random shift, drop unpopulated columns, hash more bits

q rows

$t \cdot \ell$ columns

A Candidate UOWHF (the 'right' Construction)

Framework: computational entropy

Arbitrary OWF $f: \{0,1\}^n \to \{0,1\}^n$ → Computational entropy generator g → PRG, UOWHF, …

Manipulating entropy and extraction

- ▶ HRV PRG: $g(X)$ has next-bit pseudoentropy
- ▶ HRVVW UOWHF: $g(X)$ has inaccessible entropy

Write $Z := g(X) \in \{0,1\}^\ell$. $\exists\, Y = (Y_1, \dots, Y_\ell)$:

- $\forall\, i$: $Z_1, \dots, Z_i \approx_c Z_i, \dots, Z_{i-1}, Y_i$
- $\mathbb{E}_{I \leftarrow [\ell]}[\mathbf{H}(Y_I \mid Z_1, \dots, Z_{I-1})] \geq \frac{\mathbf{H}(Z)}{\ell} + \delta$.

($\mathbf{H}(\cdot)$: Shannon entropy)
That is, on average,
each bit exhibit δ extra pseudoentropy.

$g(X_{1,1})$	$g(X_{1,2})$	$\cdots$	$g(X_{1,t})$
$g(X_{2,1})$	$g(X_{2,2})$	$\cdots$	$g(X_{2,t})$
$\vdots$	$\vdots$	$\ddots$	$\vdots$
$g(X_{q,1})$	$g(X_{q,2})$	$\cdots$	$g(X_{q,t})$

q rows

$t \cdot \ell$ columns

HRV PRG : repetition + random shift, drop unpopulated columns, hash more bits

A Candidate UOWHF (the 'right' Construction)

Framework: computational entropy

Arbitrary OWF $f\colon \{0,1\}^n \to \{0,1\}^n$ → Computational entropy generator g → PRG, UOWHF, …

Manipulating entropy and extraction

▶ HRV PRG: $g(X)$ has next-bit pseudoentropy

▶ HRVVW UOWHF: $g(X)$ has inaccessible entropy

Write $Z := g(X) \in \{0,1\}^{\ell}$. $\exists\, Y = (Y_1, \dots, Y_\ell)$:

- $\forall\, i$: $Z_1, \dots, Z_i \approx_c Z_i, \dots, Z_{i-1}, Y_i$
- $\mathbb{E}_{I \leftarrow [\ell]}[\mathbf{H}(Y_I \mid Z_1, \dots, Z_{I-1})] \geq \frac{\mathbf{H}(Z)}{\ell} + \delta$.

($\mathbf{H}(\cdot)$: Shannon entropy)

That is, on average,

each bit exhibit δ extra pseudoentropy.

q rows

$(X_{1,1})$	$g(X_{1,2})$	$\cdots$	$g(X_{1,t})$
$_{2,1})$	$g(X_{2,2})$	$\cdots$	$g(X_{2,t})$
$\vdots$	$\vdots$	$\ddots$	$\vdots$
$g(X_{q,1})$	$g(X_{q,2})$	$\cdots$	$g(X_q$

$t \cdot \ell$ columns

HRV PRG : repetition + random shift, drop unpopulated columns, hash more bits

A Candidate UOWHF (the 'right' Construction)

Framework: computational entropy

Arbitrary OWF $f: \{0,1\}^n \to \{0,1\}^n$ → Computational entropy generator g → PRG, UOWHF, …

Manipulating entropy and extraction

▶ HRV PRG: $g(X)$ has next-bit pseudoentropy

▶ HRVVW UOWHF: $g(X)$ has inaccessible entropy

Write $Z := g(X) \in \{0,1\}^\ell$. $\exists\, Y = (Y_1, \ldots, Y_\ell)$:

- $\forall\, i$: $Z_1, \ldots, Z_i \approx_c Z_i, \ldots, Z_{i-1}, Y_i$
- $\mathbb{E}_{I \leftarrow [\ell]}[\mathbf{H}(Y_I \mid Z_1, \ldots, Z_{I-1})] \geq \frac{\mathbf{H}(Z)}{\ell} + \delta.$

($\mathbf{H}(\cdot)$: Shannon entropy)
That is, on average,
each bit exhibit δ extra pseudoentropy.

$(X_{1,1})$	$g(X_{1,2})$	$\cdots$	$g(X_{1,t})$
$_{2,1})$	$g(X_{2,2})$	$\cdots$	$g(X_{2,t})$
$\vdots$	$\vdots$	$\ddots$	$\vdots$
$g(X_{q,1})$	$g(X_{q,2})$	$\cdots$	$g(X_q$

q rows

hash $h: \{0,1\}^q \to \{0,1\}^a$

$t \cdot \ell$ columns

HRV PRG : repetition + random shift, drop unpopulated columns, hash more bits

A Candidate UOWHF (the ‘right’ Construction)

Framework: computational entropy

Arbitrary OWF $f\colon \{0,1\}^n \to \{0,1\}^n$ → Computational entropy generator g → PRG, UOWHF, …

Manipulating entropy and extraction

▶ HRV PRG: $g(X)$ has next-bit pseudoentropy

▶ HRVVW UOWHF: $g(X)$ has inaccessible entropy

Next-bit version?

$g(X_{1,1})$	$g(X_{1,2})$	$\cdots$	$g(X_{1,t})$
$g(X_{2,1})$	$g(X_{2,2})$	$\cdots$	$g(X_{2,t})$
$\vdots$	$\vdots$	$\ddots$	$\vdots$
$g(X_{q,1})$	$g(X_{q,2})$	$\cdots$	$g(X_{q,t})$

q rows

hash $h\colon \{0,1\}^q \to \{0,1\}^a$

$t \cdot \ell$ columns

Write $Z := g(X) \in \{0,1\}^\ell$. $\exists\, Y = (Y_1, \dots, Y_\ell)$:

- $\forall\, i$: $Z_1, \dots, Z_i \approx_c Z_i, \dots, Z_{i-1}, Y_i$
- $\mathbb{E}_{I \leftarrow [\ell]}[\mathbf{H}(Y_I \mid Z_1, \dots, Z_{I-1})] \geq \frac{\mathbf{H}(Z)}{\ell} + \delta$.

($\mathbf{H}(\cdot)$: Shannon entropy)

That is, on average,

each bit exhibit δ extra pseudoentropy.

HRV PRG : repetition + random shift,

drop unpopulated columns, hash more bits

A Candidate UOWHF (the ‘right’ Construction)

Similar to HRV PRG

Framework: computational entropy

Arbitrary OWF $f: \{0,1\}^n \to \{0,1\}^n$ → Computational entropy generator g → PRG, UOWHF, …

Manipulating entropy and extraction

▶ HRV PRG: $g(X)$ has next-bit pseudoentropy

▶ HRVVW UOWHF: $g(X)$ has inaccessible entropy

Next-bit version?

Write $Z := g(X) \in \{0,1\}^\ell$. $\exists\, Y = (Y_1, \dots, Y_\ell)$:

- $\forall\, i$: $Z_1, \dots, Z_i \approx_c Z_i, \dots, Z_{i-1}, Y_i$
- $\mathbb{E}_{I \leftarrow [\ell]}[\mathbf{H}(Y_I \mid Z_1, \dots, Z_{I-1})] \geq \frac{\mathbf{H}(Z)}{\ell} + \delta$.

($\mathbf{H}(\cdot)$: Shannon entropy)

That is, on average, each bit exhibit δ extra pseudoentropy.

q rows

$(X_{1,1})$	$g(X_{1,2})$	$\cdots$	$g(X_{1,t})$
$_{2,1})$	$g(X_{2,2})$	$\cdots$	$g(X_{2,t})$
$\vdots$	$\vdots$	$\ddots$	$\vdots$
$g(X_{q,1})$	$g(X_{q,2})$	$\cdots$	$g(X_q$

hash $h: \{0,1\}^q \to \{0,1\}^a$

$t \cdot \ell$ columns

HRV PRG : repetition + random shift, drop unpopulated columns, hash more bits

A Candidate UOWHF (the 'right' Construction) cont'd

Framework: computational entropy

Arbitrary OWF $f: \{0,1\}^n \to \{0,1\}^n$ → Computational entropy generator g → PRG, UOWHF, …

Manipulating entropy and extraction

▶ HRV PRG: $g(X)$ has next-bit pseudoentropy

▶ HRVVW UOWHF: $g(X)$ has inaccessible entropy

Repetition + Random shift

$X_{1,1})$	$g(X_{1,2})$	$\cdots$	$g(X_{1,t})$
$_{2,1})$	$g(X_{2,2})$	$\cdots$	$g(X_{2,t})$
$\vdots$	$\vdots$	$\ddots$	$\vdots$
$g(X_{q,1})$	$g(X_{q,2})$	$\cdots$	$g(X_{q,}$

hash h

Drop unpopulated columns, hash more bits → HRV PRG

A Candidate UOWHF (the ‘right’ Construction) cont’d

Framework: computational entropy

Arbitrary OWF $f: \{0,1\}^n \to \{0,1\}^n$ → Computational entropy generator g → PRG, UOWHF, …

Manipulating entropy and extraction

- ▶ HRV PRG: $g(X)$ has next-bit pseudoentropy
- ▶ HRVVW UOWHF: $g(X)$ has inaccessible entropy

Repetition + Random shift

$g(X_{1,1})$	$g(X_{1,2})$	$\cdots$	$g(X_{1,t})$
$g(X_{2,1})$	$g(X_{2,2})$	$\cdots$	$g(X_{2,t})$
$\vdots$	$\vdots$	$\ddots$	$\vdots$
$g(X_{q,1})$	$g(X_{q,2})$	$\cdots$	$g(X_{q,t})$

hash h

Drop unpopulated columns, hash more bits → HRV PRG

A Candidate UOWHF (the ‘right’ Construction) cont’d

Framework: computational entropy

Arbitrary OWF $f: \{0,1\}^n \to \{0,1\}^n$ → Computational entropy generator g → PRG, UOWHF, …

Manipulating entropy and extraction

- ▶ HRV PRG: $g(X)$ has next-bit pseudoentropy
- ▶ HRVVW UOWHF: $g(X)$ has inaccessible entropy

Repetition + Random shift

$g(X_{1,1})$	$g(X_{1,2})$	$\cdots$	$g(X_{1,t})$
$g(X_{2,1})$	$g(X_{2,2})$	$\cdots$	$g(X_{2,t})$
$\vdots$	$\vdots$	$\ddots$	$\vdots$
$g(X_{q,1})$	$g(X_{q,2})$	$\cdots$	$g(X_{q,t})$

hash h

Drop unpopulated columns, hash more bits → HRV PRG

Output unpopulated columns, hash fewer bits ? → UOWHF

A Candidate UOWHF (the ‘right’ Construction) cont’d

Framework: computational entropy

Arbitrary OWF $f: \{0,1\}^n \to \{0,1\}^n$ → Computational entropy generator g → PRG, UOWHF, …

Manipulating entropy and extraction

- ▶ HRV PRG: $g(X)$ has next-bit pseudoentropy
- ▶ HRVVW UOWHF: $g(X)$ has inaccessible entropy

Repetition + Random shift

$g(X_{1,1})$	$g(X_{1,2})$	$\cdots$	$g(X_{1,t})$
$g(X_{2,1})$	$g(X_{2,2})$	$\cdots$	$g(X_{2,t})$
$\vdots$	$\vdots$	$\ddots$	$\vdots$
$g(X_{q,1})$	$g(X_{q,2})$	$\cdots$	$g(X_{q,t})$

hash h

Drop unpopulated columns, hash more bits → HRV PRG

Output unpopulated columns, hash fewer bits ? → UOWHF

We introduce next-bit unreachable entropy and show that:

→ almost-UOWHF

Next-bit unreachable entropy

We say $g: \{0,1\}^m \to \{0,1\}^\ell$ has next-bit unreachable entropy Δ if for every $i \in [\ell]$, there exists a set $\mathcal{U}_i \subseteq \{0,1\}^m$, such that:

- It is hard to flip the i-th bit **while staying inside** $\mathcal{U}_i$: $\forall$ PPT A
$$\Pr[g(X)_{<I} = g(X')_{<I} \wedge g(X)_I \neq g(X')_I \wedge X' \in \mathcal{U}_I] = \mathrm{negl}(n).$$
- $\mathcal{U}$ is large: $\Pr[X_I \in \mathcal{U}_I] \geq \frac{\ell - m + \Delta}{\ell}$
- Hard to get inside $\mathcal{U}$: $\forall$ PPT A
$$\Pr[g(X)_{<I} = g(X')_{<I} \wedge X \notin \mathcal{U}_I \wedge X' \in \mathcal{U}_I] = \mathrm{negl}(n).$$

$$\boxed{X \leftarrow \{0,1\}^m, I \leftarrow [\ell], X' \leftarrow A(X, I).}$$

Next-bit unreachable entropy

We say $g: \{0,1\}^m \to \{0,1\}^\ell$ has next-bit unreachable entropy Δ if for every $i \in [\ell]$, there exists a set $\mathcal{U}_i \subseteq \{0,1\}^m$, such that:

- It is hard to flip the i-th bit **while staying inside** $\mathcal{U}_i$: $\forall$ PPT A
  $$\Pr[g(X)_{<I} = g(X')_{<I} \wedge g(X)_I \neq g(X')_I \wedge X' \in \mathcal{U}_I] = \mathrm{negl}(n).$$
- $\mathcal{U}$ is large: $\Pr[X_I \in \mathcal{U}_I] \geq \frac{\ell - m + \Delta}{\ell}$
- Hard to get inside $\mathcal{U}$: $\forall$ PPT A
  $$\Pr[g(X)_{<I} = g(X')_{<I} \wedge X \notin \mathcal{U}_I \wedge X' \in \mathcal{U}_I] = \mathrm{negl}(n).$$

$$X \leftarrow \{0,1\}^m, I \leftarrow [\ell], X' \leftarrow A(X, I).$$

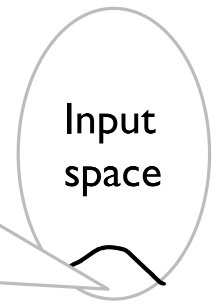
HRV next−bit pseudoentropy generator: $g(h, x) \coloneqq (f(x), h(x), h)$

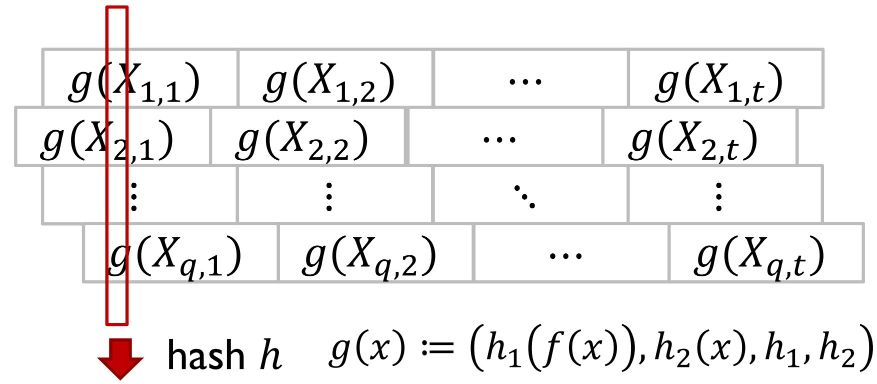
Our next−bit unreachable entropy generator: $g(h_1, h_2, x) \coloneqq \big(h_1(f(x)), h_2(x), h_1, h_2\big)$

$^*h,\ h_1,\ h_2$ are from proper hash families

Almost-UOWHF: What's the point?

Almost-UOWHF:
$\exists$ a negligible fraction of inputs $\mathcal{B}$
such that any adversary can find
collision x' only from $\mathcal{B}$.

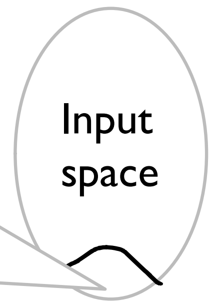




- Our construction is very similar to the HRV PRG construction.
- The HRV PRG construction is actually an "Almost-PRG".
- Fortunately, Almost-PRG = PRG.

Almost-UOWHF: What's the point?

Almost-UOWHF:
$\exists$ a negligible fraction of inputs $\mathcal{B}$ such that any adversary can find collision x' only from $\mathcal{B}$.



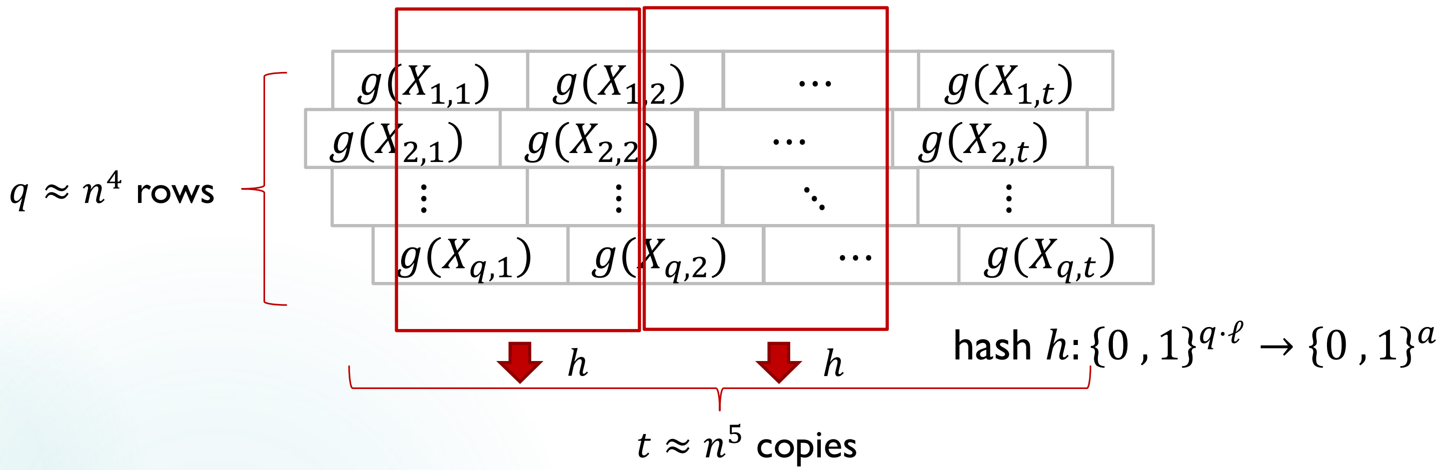
$g(X_{1,1})$	$g(X_{1,2})$	$\cdots$	$g(X_{1,t})$
$g(X_{2,1})$	$g(X_{2,2})$	$\cdots$	$g(X_{2,t})$
$\vdots$	$\vdots$	$\ddots$	$\vdots$
$g(X_{q,1})$	$g(X_{q,2})$	$\cdots$	$g(X_{q,t})$

hash h $\quad g(x) \coloneqq \big(h_1(f(x)), h_2(x), h_1, h_2\big)$

- Our construction is very similar to the HRV PRG construction.
- The HRV PRG construction is actually an "Almost-PRG".
- Fortunately, Almost-PRG = PRG.

Almost-PRG:
$G(U|_{U \notin \mathcal{B}}) \approx_c$ *uniform random bits*, where $\mathcal{B}$ contains negligible fraction of inputs.

Non-adaptive UOWHF



Modifications towards a full-fledged UOWHF

- ▶ Use large q, t
- ▶ Hash a $\ell \cdot q$ block instead of hashing a single column

→ Collision-resistant on random inputs* ✔

*In order to get a simpler proof by existing techniques,
we actually prove that an equivalent construction is UOWHF.

Open Questions

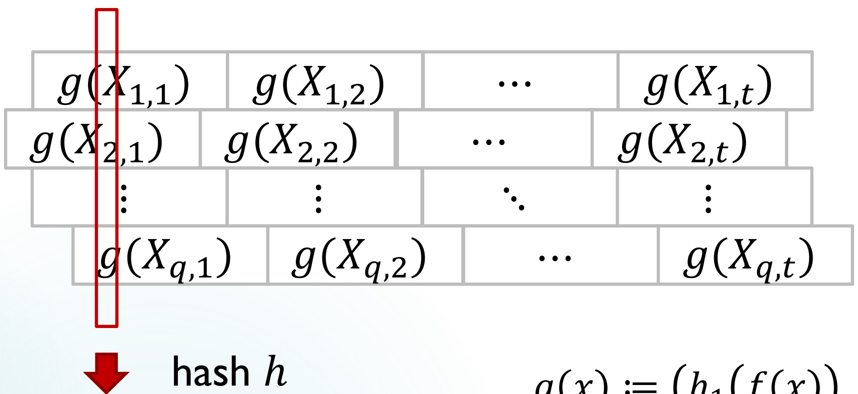
Open Questions

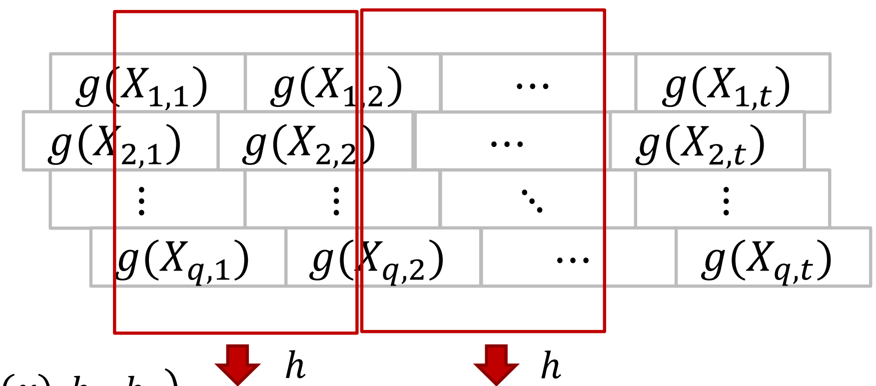
- Conjecture. Our Almost-UOWHF construction is a full-fledged UOWHF.
  - Do we need to modify our next-bit unreachable entropy definition?
  - Even with a more natural computational entropy generator: $g(x) \coloneqq (f(x), x)$
    - This is used in [VZ'12] to construct PRG.

Open Questions

- Conjecture. Our Almost-UOWHF construction is a full-fledged UOWHF.
  - Do we need to modify our next-bit unreachable entropy definition?
  - Even with a more natural computational entropy generator: $g(x) \coloneqq (f(x), x)$
    - This is used in [VZ'12] to construct PRG.
- Lower bounds on black-box constructions from OWF:
  - seed length
  - number of calls
  - Both PRG and UOWHFs

Thank you!





$$g(x) \coloneqq \big(h_1(f(x)), h_2(x), h_1, h_2\big)$$

	Seed length	**Number of calls**	**Non-adaptive?**
[HHRVW' 10]	$\tilde{O}(n^5)$	$\tilde{O}(n^{13})$	×
Our UOWHF	$\tilde{O}(n^{10})$	$\tilde{O}(n^9)$	√
Our Almost-UOWHF	$\tilde{O}(n^4)$	$\tilde{O}(n^3)$	√