Universal Computational Extractors and Multi-Bit AIPO From Lattice Assumptions

@Eurocrypt 2025

Yilei Chen* and Xinyu Mao**

* Tsinghua University

** University of Southern California

Instantiating Random Oracle

- Many simple and efficient schemes only have security proof in the ROM.
- Heuristic: Use cryptographic hash functions (e.g., SHA3) to replace RO.
- [CGH04]: RO is **uninstantiable** in general.
  - There exists an encryption scheme that is secure in the ROM but insecure when RO is replaced by any function.
- Belief: Counterexamples are artificially contrived.

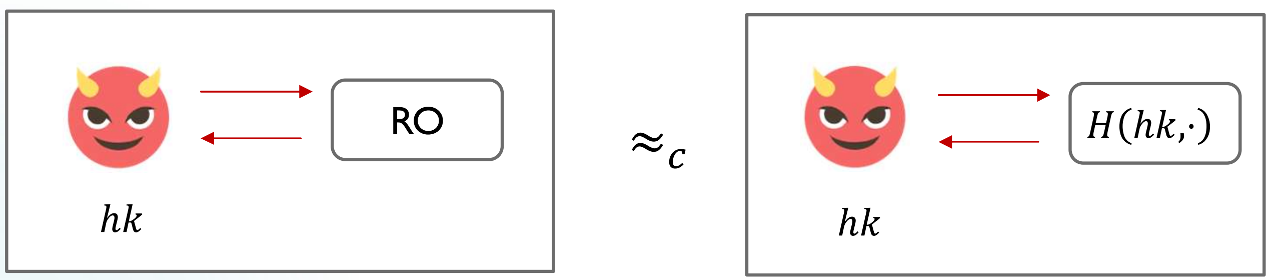
Remedy:

1. Identify **'RO-like' properties** that are sufficient for important applications.
2. Construct hash functions with such properties under well-formed assumptions.

This paper: Universal Computational Extractors and Point Obfuscation

Universal Computational Extractor [BHK13]

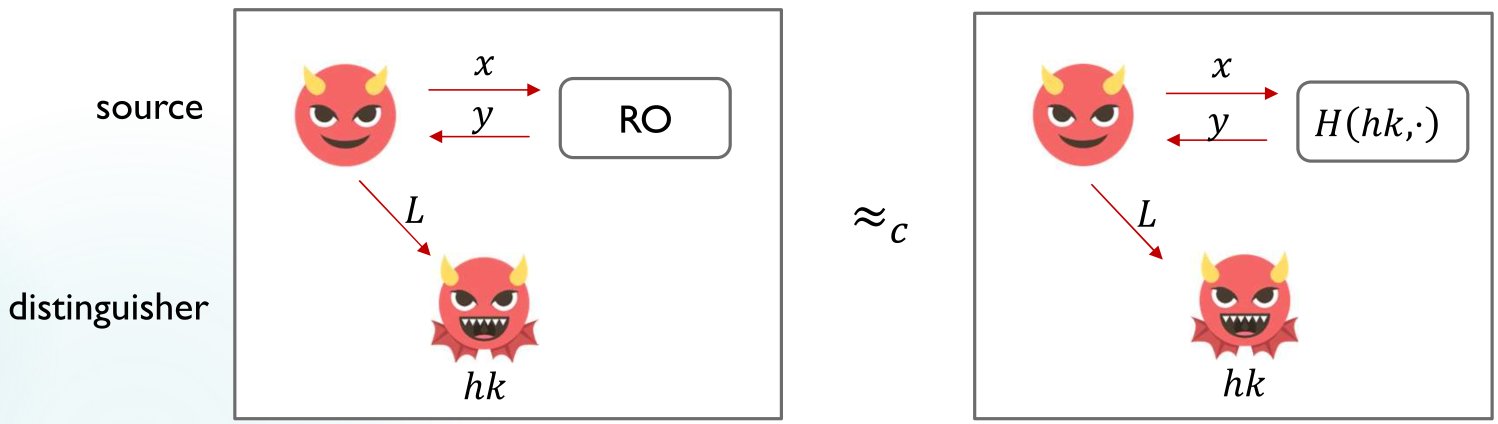
What is a ‘random-oracle-like’ hash function?



$$H: K \times X \to Y$$

- Easy to distinguish by evaluating at a single point.
- To good to be true…

Universal Computational Extractor [BHK13]



$H: K \times X \to Y$

is unpredictable if x is unpredictable given L

is strongly unpredictable if x is unpredictable given L and y

H is a UCE for (1-query) unpredictable sources if $\approx_c$ holds for all unpredictable

Point Obfuscators with Auxiliary Input (AIPO)

AIPO
(for unpredictable sources)

If x is unpredictable given aux,

then

$PO(x)$, aux $\approx_c$ $PO(null)$, aux

- $PO(x)$ outputs a program that computes the point function 1_x.
- $PO(null)$ outputs a program that computes the all-zero function.

Multi-Bit AIPO
(for strongly unpredictable sources)

If x is unpredictable given aux and m.

then

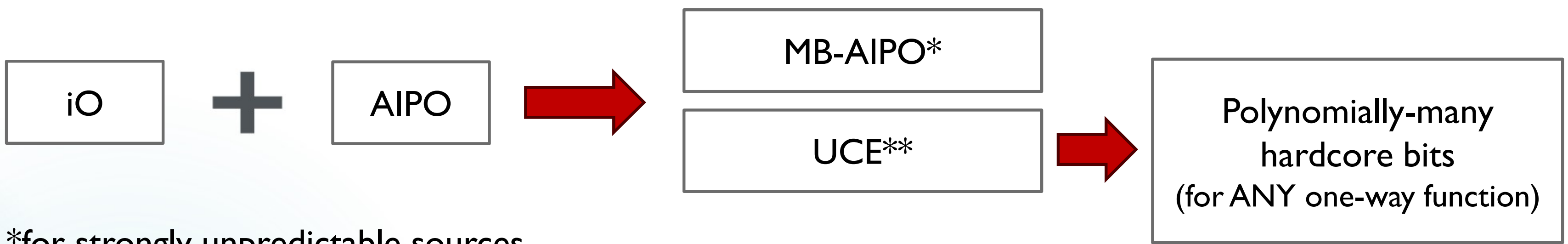
$MBPO(x, m)$, aux $\approx_c$ $MBPO(x, \$)$, aux

- $MBPO(x, m)$ outputs a program that computes the function

$$p_{x,m}(z) = \begin{cases} m & \text{if } z = x \\ \perp & \text{o.w.} \end{cases}$$

Constructions of UCE and MB-AIPO

[BM14]



*for strongly unpredictable sources

**for 1-query stronly unpredictable sources

- Assume the existence if AIPO,

Puncturable PRF

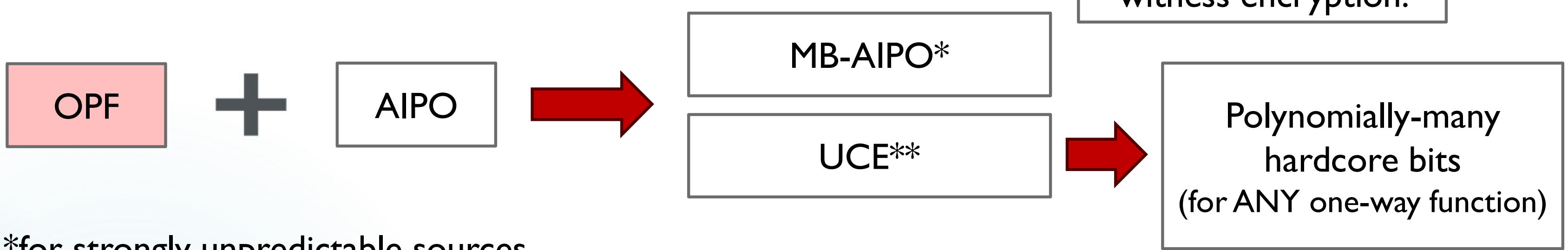
$iO(PPRF(hk,\cdot))$ is UCE**

$MBPO(x,y) \coloneqq iO\left(\begin{array}{c} \text{On input } z\text{:} \\ \text{If } z = x \text{ output } y \\ \text{else output } \perp \end{array} \right)$ is MB-AIPO*.

Our Results

Constructions from <u>Obliviously Programmable Function</u>

New abstraction somewhat similar to witness encryption.



*for strongly unpredictable sources
**for 1-query stronly unpredictable sources

OPF from <u>lattice assumptions</u>

- Subexponential LWE + (private-coin) evasive LWE
- Similar to CVW witness encryption candidate

Obliviously Programmable Function (OPF)

Keyed function: $OPF: K \times X \to Y$ with an algorithm $OPF.\mathrm{Program}$.

$$OPF.\mathrm{Program}(C, y^*) \mapsto k_C$$

A circuit in $\mathcal{C}$

A value $y^* \in Y$

programmed key

- **Correctness.** If C computes the point function 1_{x^*}, then $OPF(k_C, x^*) = y^*$.
- **Privacy.** k_C computationally hides C provided that
  - C computes a point function or the all-zero function
  - y^* is chosen uniformly at random.
- **Value-Hiding.** If C computes the all-zero function, then k_C computationally hides y^*.

Theorem 1 (OPF → UCE). Let $H(hk, x) \coloneqq OPF(hk, x)$.
$H.Gen$ outputs $hk \leftarrow OPF(\mathrm{AllZeroFunction}, 0)$.
If there exists AIPO in $\mathcal{C}$, then H is a UCE (for 1-query strongly unpredictable sources).

Lattice-Based OPF Construction

(based on GGH15 encoding)

GGH15 encodings

- Circuits are represented as matrix branching programs (MBPs).
- A MBP $\Gamma = \left(\mathbf{v} \in \{0,1\}^{w}, \{\mathbf{M}_{i,b}\}_{i\in[h], b\in\{0,1\}}\right)$ on input $x \in \{0,1\}^h$, output 1 if

$\mathbf{v}^\top \mathbf{M}_x := \mathbf{v}^\top \mathbf{M}_{1,x_1}\mathbf{M}_{2,x_2}\cdots\mathbf{M}_{h,x_h} = \mathbf{0}$, and output 0 otherwise.

	$\mathbf{M}_{1,0}$	$\mathbf{M}_{2,0}$		$\mathbf{M}_{h,0}$
$\mathbf{v}^\top$			$\cdots$	
	$\mathbf{M}_{1,1}$	$\mathbf{M}_{2,1}$		$\mathbf{M}_{h,1}$

To encode this MBP:

1. Construct

$$\widehat{\mathbf{S}}_{1,b} = \left(\mathbf{I}_n \mid \mathbf{v}^\top \mathbf{M}_{1,b} \otimes \mathbf{S}_{1,b}\right), \widehat{\mathbf{S}}_{i,b} = \begin{pmatrix} \mathbf{I}_n & \\ & \mathbf{M}_{i,b} \otimes \mathbf{S}_{i,b} \end{pmatrix} \text{ for } i = 2, \ldots, h, \text{ where } \mathbf{S}_{j,b} \leftarrow D_\sigma^{n\times n}$$

2. $\text{GGH.Encode}(\{\widehat{\mathbf{S}}_{i,b}\}) = \left\{ \widehat{\mathbf{S}}_{1,b}\mathbf{A}_1, \mathbf{A}_1^{-1}(\widehat{\mathbf{S}}_{2,b}\mathbf{A}_2), \ldots, \mathbf{A}_{h-1}^{-1}(\widehat{\mathbf{S}}_{h,b}\mathbf{A}_h) \right\}_{b\in\{0,1\}}$ where $\mathbf{A}_j \leftarrow \mathbb{Z}_q^{(n+nw)\times O(nw)}$

GGH15 encodings (continued)

Γ on input $x \in \{0,1\}^h$, outputs 1 if $\mathbf{v}^\top \mathbf{M}_x = \mathbf{0}$, and outputs 0 otherwise.

$$\mathbf{v}^\top \quad \begin{matrix} \boxed{\mathbf{M}_{1,0}} & \mathbf{M}_{2,0} & \cdots & \boxed{\mathbf{M}_{h,0}} \\ \mathbf{M}_{1,1} & \boxed{\mathbf{M}_{2,1}} & & \mathbf{M}_{h,1} \end{matrix}$$

To encode this MBP:

1. Construct

$$\widehat{\mathbf{S}}_{1,b} = (\mathbf{I}_n \mid \mathbf{v}^\top \mathbf{M}_{1,b} \otimes \mathbf{S}_{1,b}),\ \widehat{\mathbf{S}}_{i,b} = \begin{pmatrix} \mathbf{I}_n & \\ & \mathbf{M}_{i,b} \otimes \mathbf{S}_{i,b} \end{pmatrix} \text{ for } i = 2, \ldots, h,$$

2. $\text{GGH.Encode}(\{\widehat{\mathbf{S}}_{i,b}\}) = \left\{ \widehat{\mathbf{S}}_{1,b}\mathbf{A}_1, \mathbf{A}_1^{-1}(\widehat{\mathbf{S}}_{2,b}\mathbf{A}_2), \ldots, \mathbf{A}_{h-1}^{-1}(\widehat{\mathbf{S}}_{h,b}\mathbf{A}_h) \right\}_{b \in \{0,1\}}$

- ▶ **Functionality**: Given the encodings, one can approximate $\widehat{\mathbf{S}}_x \mathbf{A}_h \coloneqq \widehat{\mathbf{S}}_{1,x_1} \widehat{\mathbf{S}}_{2,x_2} \cdots \widehat{\mathbf{S}}_{h,x_h} \mathbf{A}_h$.
- ▶ **Security**: Encodings are pseudorandom.

- ▶ Programmed key k_Γ := encodings
- ▶ $OPF(k_\Gamma, x) \coloneqq$ the approx. of $\widehat{\mathbf{S}}_x \mathbf{A}_h$ given by the encodings.

How to program the value at x^*?

Constructing OPF

OPF construction:
- Programmed key k_Γ := encodings
- $OPF(k_\Gamma, x)$:= the approx. of $\hat{\mathbf{S}}_x\mathbf{A}_h$ given by the encodings.

$$\widehat{\mathbf{S}}_{1,b} = \left(\mathbf{I}_n \mid \mathbf{v}^\top \mathbf{M}_{1,b} \otimes \mathbf{S}_{1,b}\right), \widehat{\mathbf{S}}_{i,b} = \begin{pmatrix} \mathbf{I}_n & \\ & \mathbf{M}_{i,b} \otimes \mathbf{S}_{i,b} \end{pmatrix} \text{ for } i = 2, \ldots, h,$$

- $OPF(k_\Gamma, x) \approx \hat{\mathbf{S}}_x\mathbf{A}_h = (\mathbf{I}_n \mid \mathbf{v}^\top\mathbf{M}_x \otimes \mathbf{S}_x)\,\mathbf{A}_h = \overline{\mathbf{A}_h} + (\mathbf{v}^\top\mathbf{M}_x \otimes \mathbf{S}_x) \cdot \underline{\mathbf{A}_h}$
- $\mathbf{v}^\top\mathbf{M}_{x^*} = 0 \rightarrow OPF(k_\Gamma, x^*) \approx \overline{\mathbf{A}_h}$
- We can program the value $OPF(k_\Gamma, x^*)$ by controlling $\overline{\mathbf{A}_h}$!

Top n rows of $\mathbf{A}_h$

For security, we prove that the encodings are pseudorandom.

Security: Reduction via evasive LWE

$$\widehat{\mathbf{S}}_{1,b} = \left(\mathbf{I}_n \mid \mathbf{v}^\top \mathbf{M}_{1,b} \otimes \mathbf{S}_{1,b}\right), \widehat{\mathbf{S}}_{i,b} = \begin{pmatrix} \mathbf{I}_n & \\ & \mathbf{M}_{i,b} \otimes \mathbf{S}_{i,b} \end{pmatrix} \text{ for } i = 2, \ldots, h,$$

$$\text{GGH.Encode}(\{\widehat{\mathbf{S}}_{i,b}\}) = \left\{ \underset{\sim}{\widehat{\mathbf{S}}_{1,b}\mathbf{A}_1}, \mathbf{A}_1^{-1}(\underset{\sim}{\widehat{\mathbf{S}}_{2,b}\mathbf{A}_2}), \ldots, \mathbf{A}_{h-1}^{-1}(\underset{\sim}{\widehat{\mathbf{S}}_{h,b}\mathbf{A}_h}) \right\}_{b \in \{0,1\}}$$

The encodings are pseudorandom.

[VWW22] , relying on evasive LWE

The evaluated products $\left\{\hat{\mathbf{S}}_x \mathbf{A}_h + \mathbf{E}_x\right\}_{x \in \{0,1\}^h}$ are pseudorandom.

This is exactly the privacy property we want!

$$\begin{aligned} \hat{\mathbf{S}}_x \mathbf{A}_h + \mathbf{E}_x &= \overline{\mathbf{A}_h} + (\mathbf{v}^\top \mathbf{M}_x \otimes \mathbf{S}_x) \cdot \underline{\mathbf{A}_h} + \mathbf{E}_x \\ &= \overline{\mathbf{A}_h} + (\mathbf{v}^\top \mathbf{M}_x \otimes \mathbf{I}) \cdot (\mathbf{I} \otimes \mathbf{S}_x) \cdot \underline{\mathbf{A}_h} + \mathbf{E}_x \\ &\approx \overline{\mathbf{A}_h} + (\mathbf{v}^\top \mathbf{M}_x \otimes \mathbf{I}) \cdot [(\mathbf{I} \otimes \mathbf{S}_x) \cdot \underline{\mathbf{A}_h} + \mathbf{E}_x'] \end{aligned}$$

Pseudorandom by LWE

The encoding is pseudorandom if:

- ▶ Γ computes a point function or the all-zero function;
- ▶ $\overline{\mathbf{A}_h}$ is chosen uniformly at random.

Putting it together

Theorem 2 (OPF from GGH15 encodings).
Assuming subexponential LWE and evasive LWE, there exists an OPF for $\mathbf{NC^1}$.

Theorem 1 (OPF → UCE). Let $H(hk, x) := OPF(hk, x)$.
$H.Gen$ outputs $hk \leftarrow OPF(\text{AllZeroFunction}, 0)$.
If there exists AIPO in $\mathcal{C}$, then H is a UCE*.

Main Theorem. There exist UCE* and MB-AIPO** under the following assumptions:

1. Subexponential LWE;
2. (private-coin) evasive LWE;
3. the existence of AIPO in $\mathbf{NC^1}$.

* for 1-query strongly unpredictable sources
** for strongly unpredictable sources

Discussion

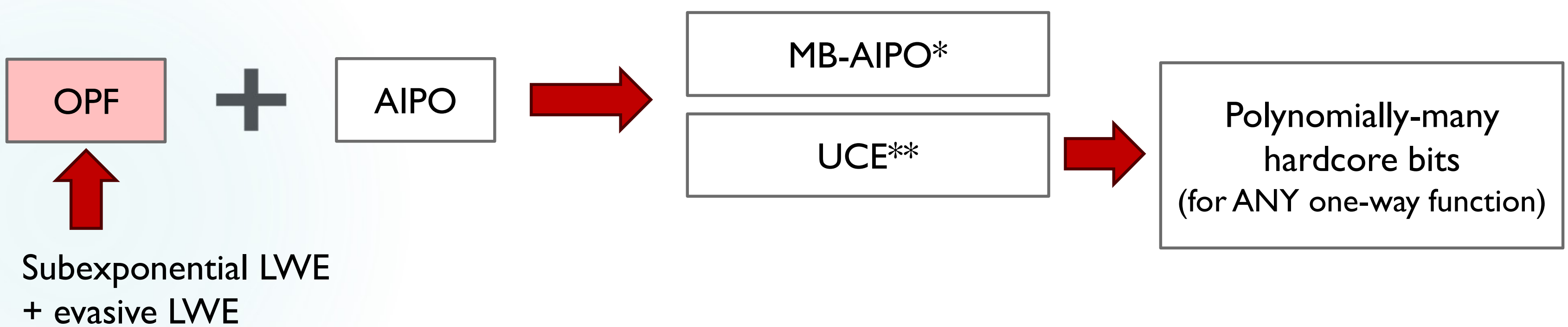
- Programming for all circuits? Programming on more points?
- Can we base the security on standard LWE?
  - Closely related to CVW witness encryption.
- Can we Use OPF to instantiate RO in other applications?
  - E.g., full domain hash signatures.

Suppose that we have some joint distributions over matrices $\mathbf{P}, \mathbf{S}$ and auxiliary information aux. Private-coin evasive LWE assumption postulates that, for a uniformly random (and secret) matrix $\mathbf{B}$,

$$\begin{array}{ll}\textbf{if} & (\mathbf{SB}+\mathbf{E}, \mathbf{SP}+\mathbf{E}', \text{aux}) \approx_c (\mathbf{U}, \mathbf{U}', \text{aux}) \\ \textbf{then} & (\mathbf{SB}+\mathbf{E}, \mathbf{B}^{-1}(\mathbf{P}), \text{aux}) \approx_c (\mathbf{U}, \mathbf{B}^{-1}(\mathbf{P}), \text{aux})\end{array}$$

where $\mathbf{U}, \mathbf{U}'$ are uniformly random matrices, and $\mathbf{E}, \mathbf{E}'$ are chosen from the LWE error distribution.

Thank you for listening! ☺



*for strongly unpredictable sources
**for 1-query stronly unpredictable sources