Using Indistinguishability Obfuscation with UCEs

ASIACRYPT, Dec 10th, 2014

Christina Brzuska
Arno Mittelbach

The results in a nutshell

- New technique to work with indistinguishability Obfuscation
  - Extension of punctured programs technique to hide punctured points

→ **Use Point Function Obfuscation within iO**

Universal Hardcore Function
(under very strong PO)

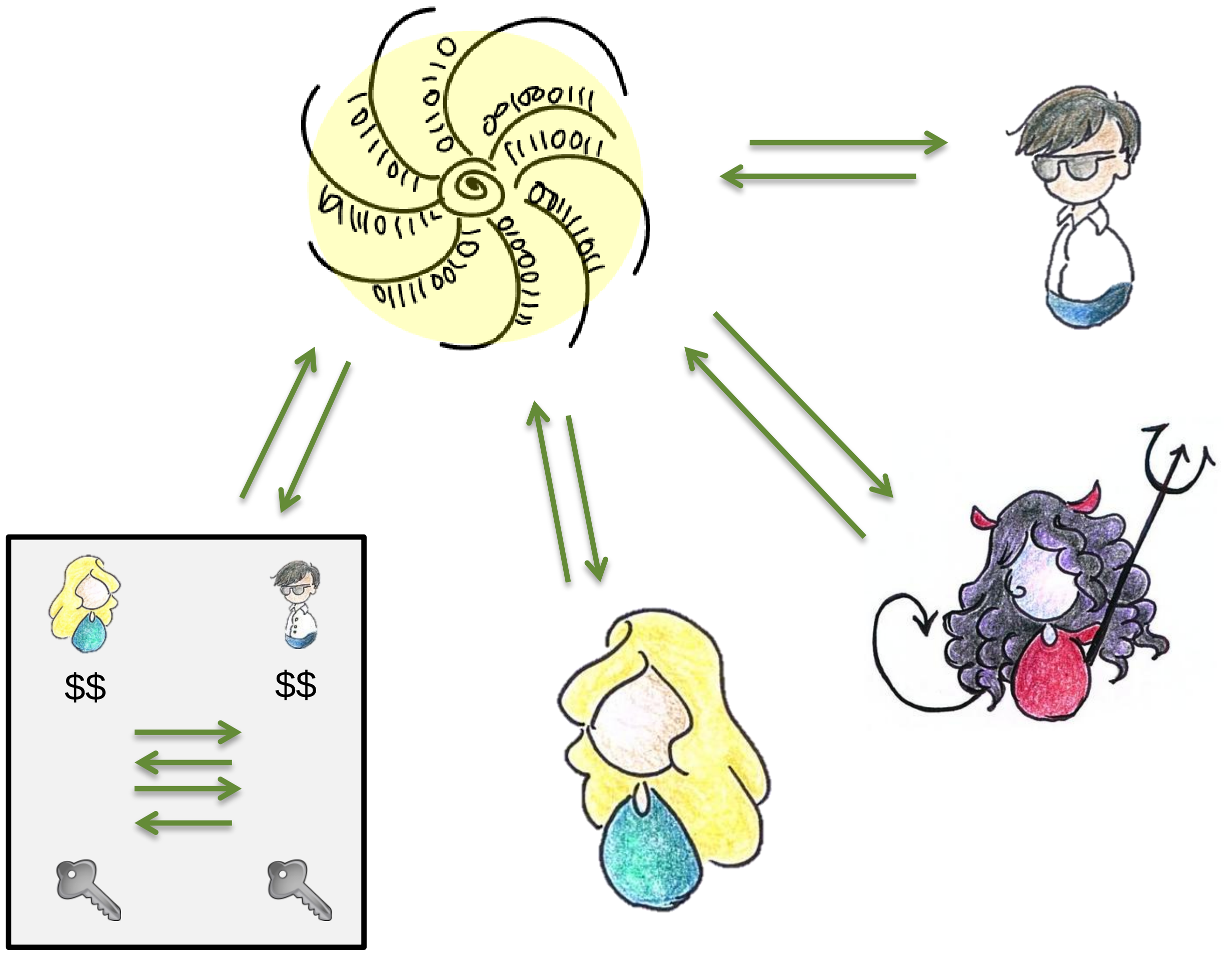
$$\mathrm{iO}\,(\mathrm{PRF}(k,\cdot))$$

q-query correlated input secure hash function
(under weaker PO)

↳ **UCE secure with respect to strong unpredictability**

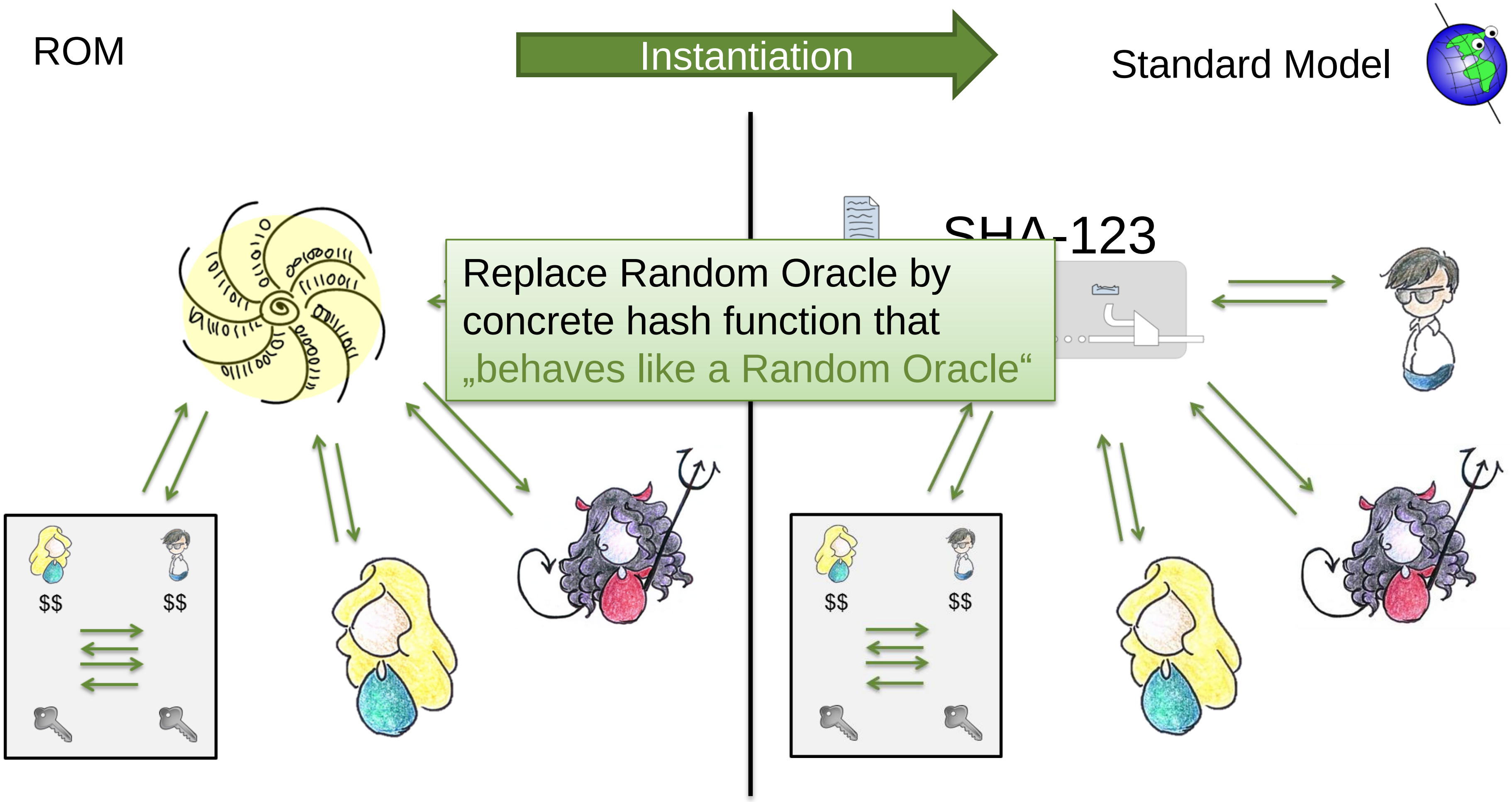
What are UCEs?

The Random Oracle Model (ROM)



[Nice drawings by Giorgia Azzurra Marson others by Arno Mittelbach]

The Random Oracle Model (ROM)



Replace Random Oracle by concrete hash function that „behaves like a Random Oracle“

[Nice drawings by Giorgia Azzurra Marson others by Arno Mittelbach]

Random Oracles are Practical

[BR93]

Random Oracles are controversial

[CGH98,Nie02,GK03,MRH04,DOP05 ,BBP04, CGH04,BFM14]…

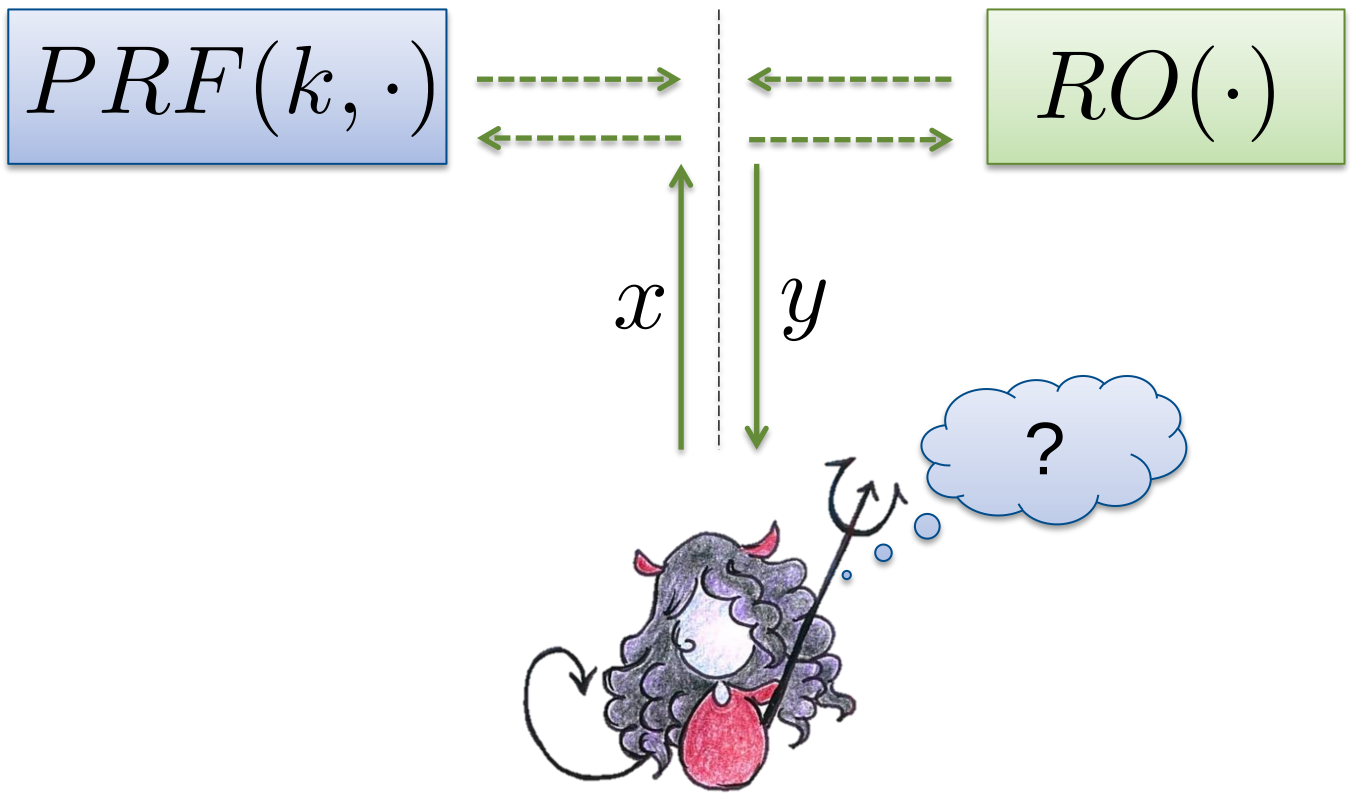
Bellare, Hoang, Keelveedhi (Crypto 2013) [BHK13]

The lack of a proof of security for the instantiated scheme is […] a consequence of an even more fundamental lack, namely that of a definition, of what it means for a family of functions to "behave like a RO"

[BHK13]

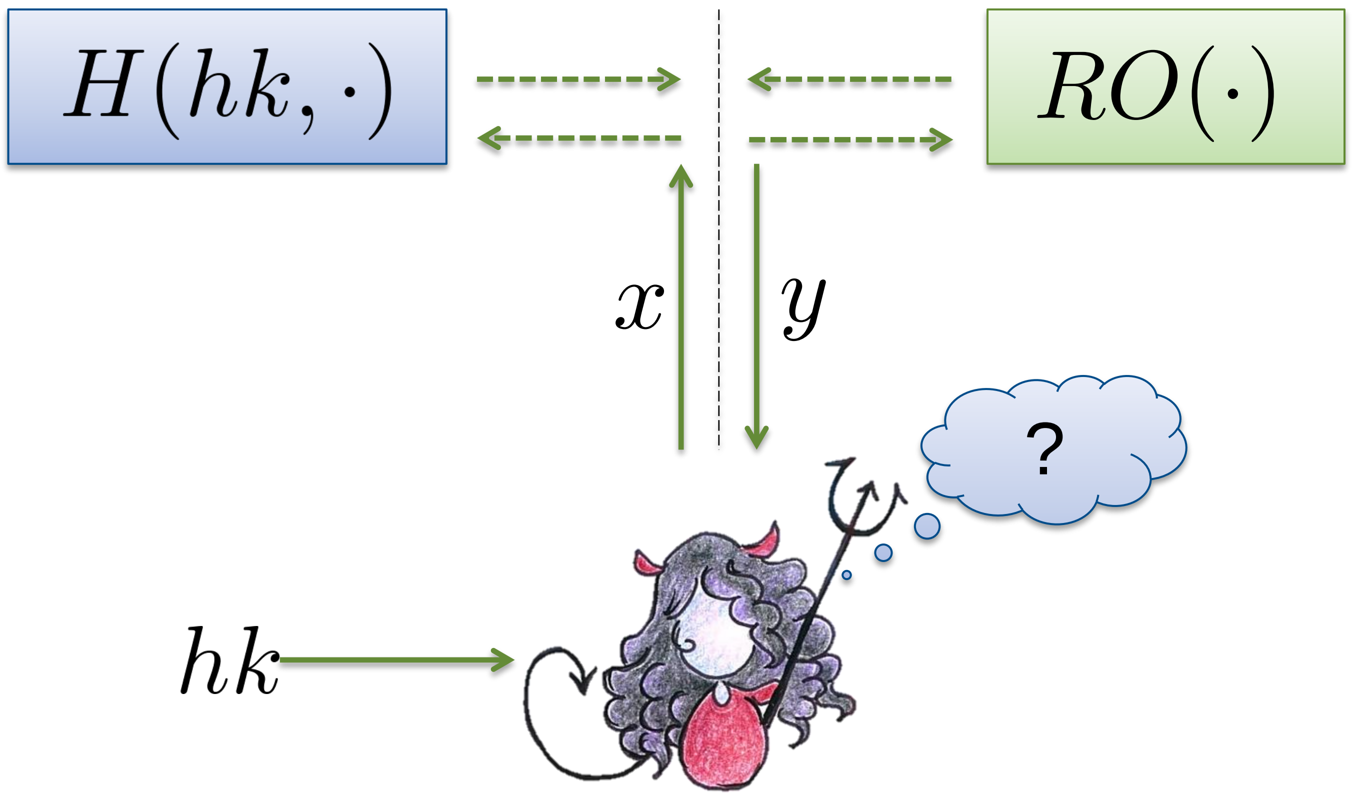
UCE Framework [BHK13]
(Universal Computational Extractors)

The symmetric setting:



UCE Framework [BHK13]
(Universal Computational Extractors)

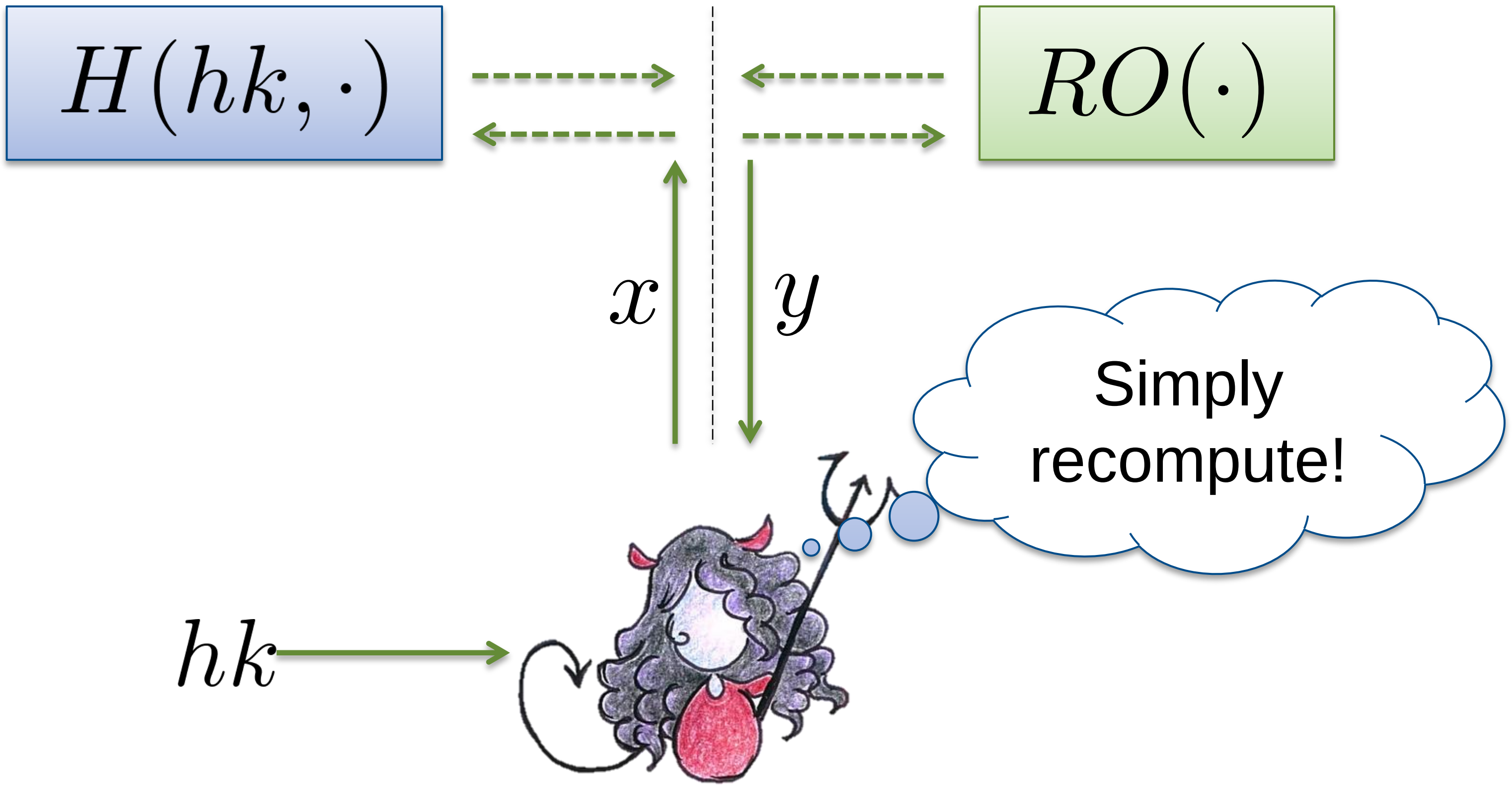
The public-key setting:



UCE Framework [BHK13]

(Universal Computational Extractors)

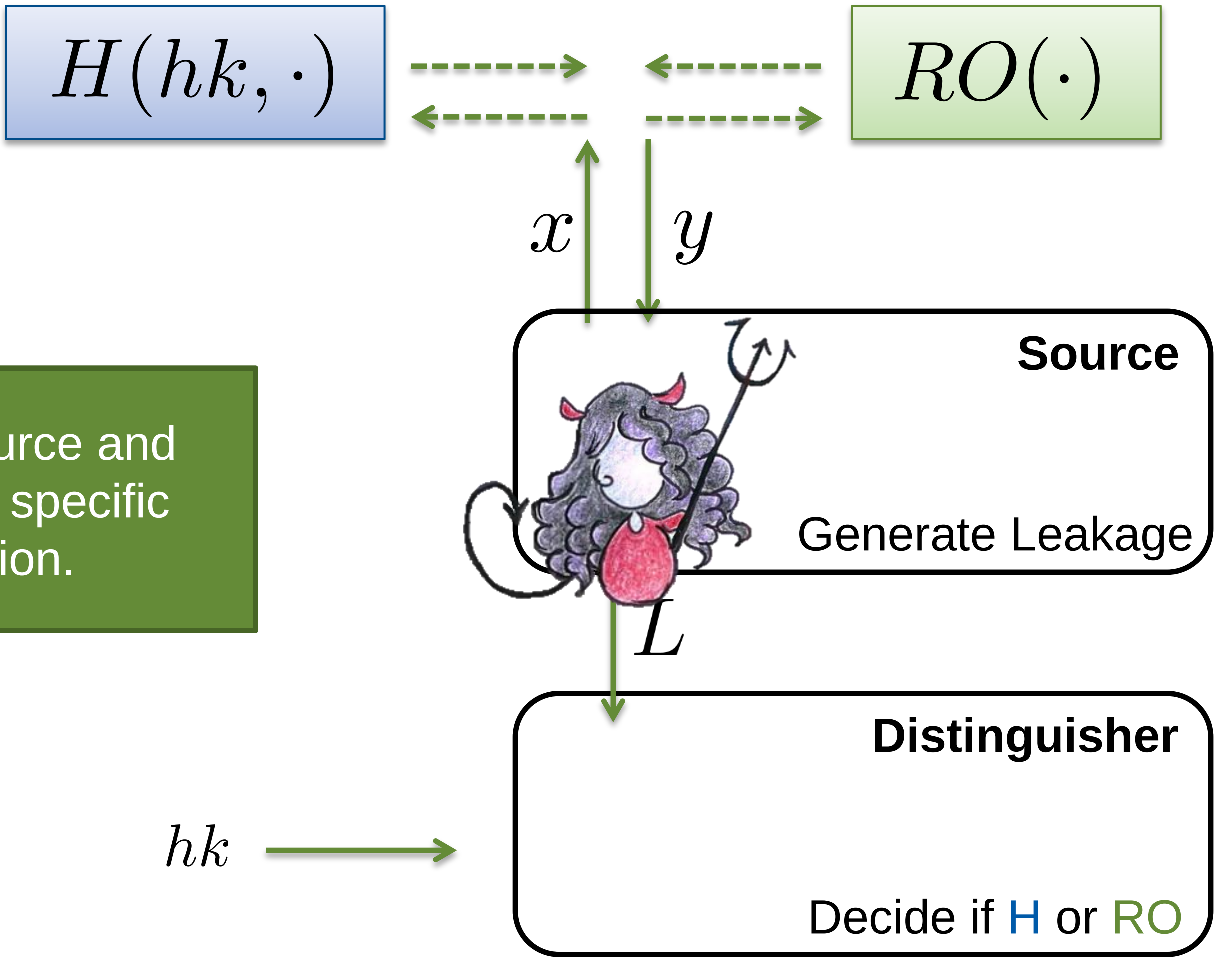
<u>The public-key setting:</u>



UCE Framework [BHK13]

(Universal Computational Extractors)

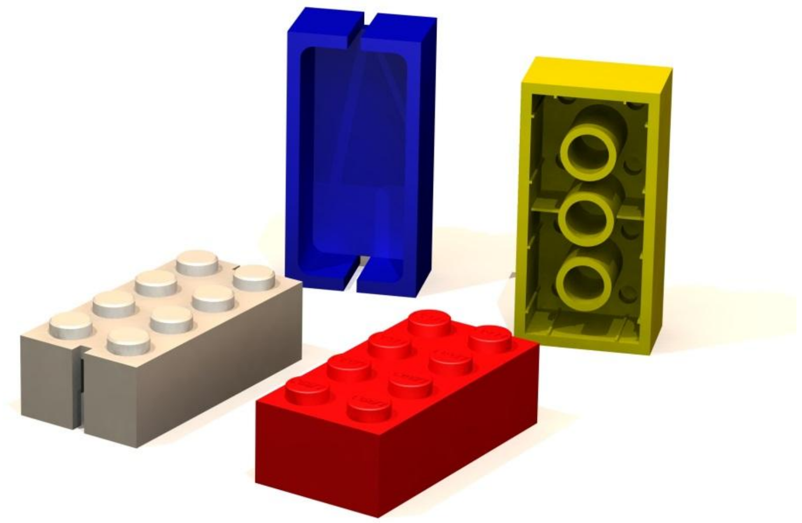
The UCE setting:



Restrictions on Source and Distinguisher yield specific UCE assumption.

UCE Framework [BHK13]
(Universal Computational Extractors)

UCE (Universal Computational Extractors) is a Framework

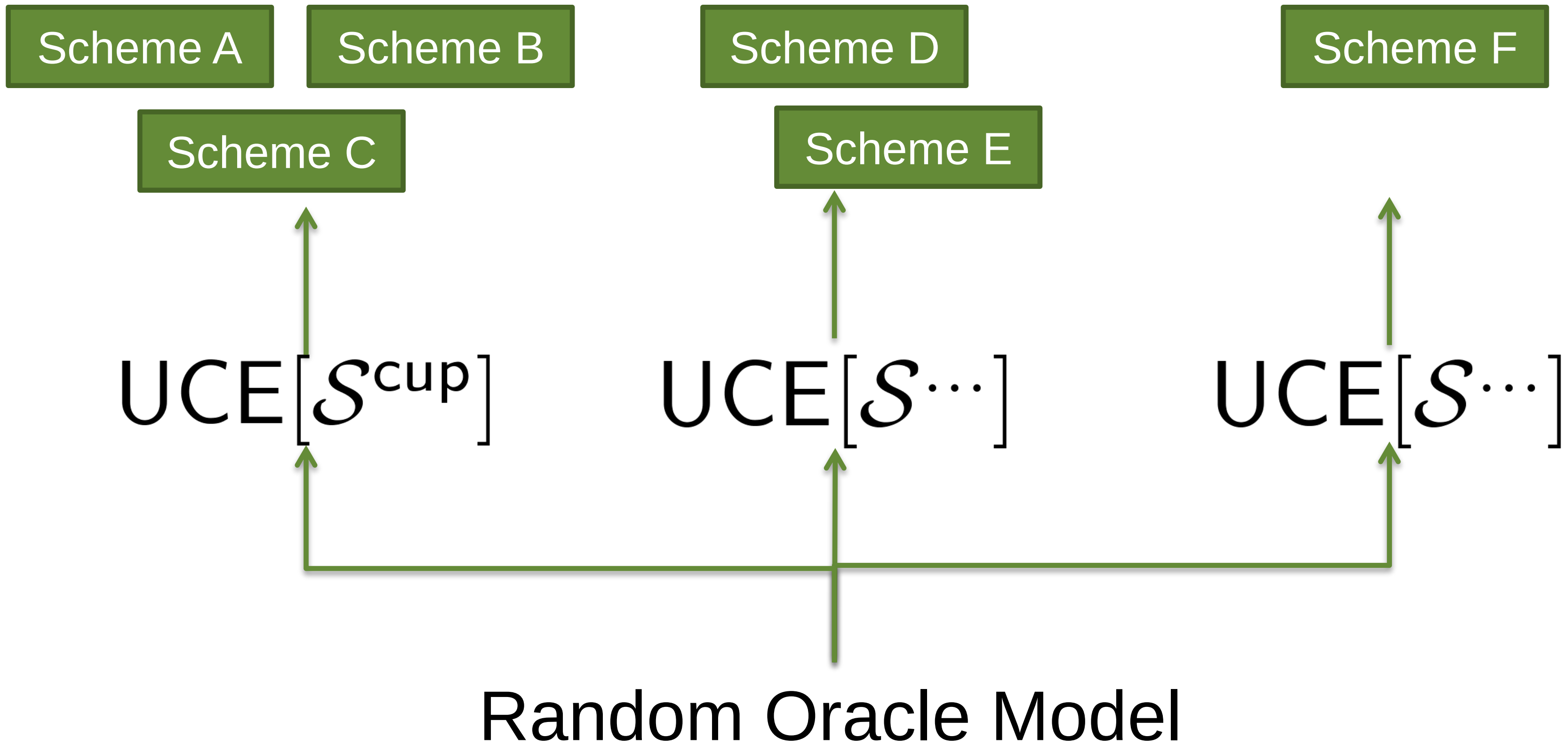


to design assumptions that describe features of a random oracle

What are good UCEs?

UCE Framework [BHK13]

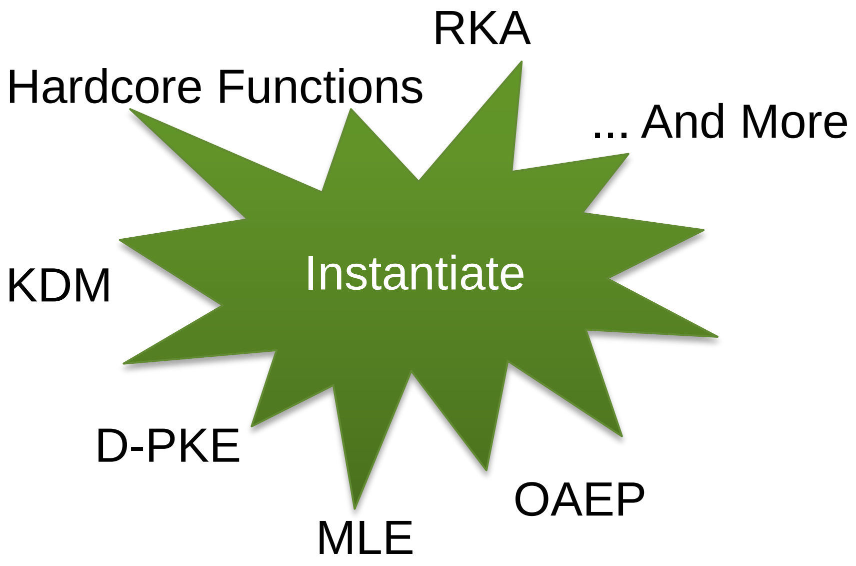
Layered Cryptography Paradigm

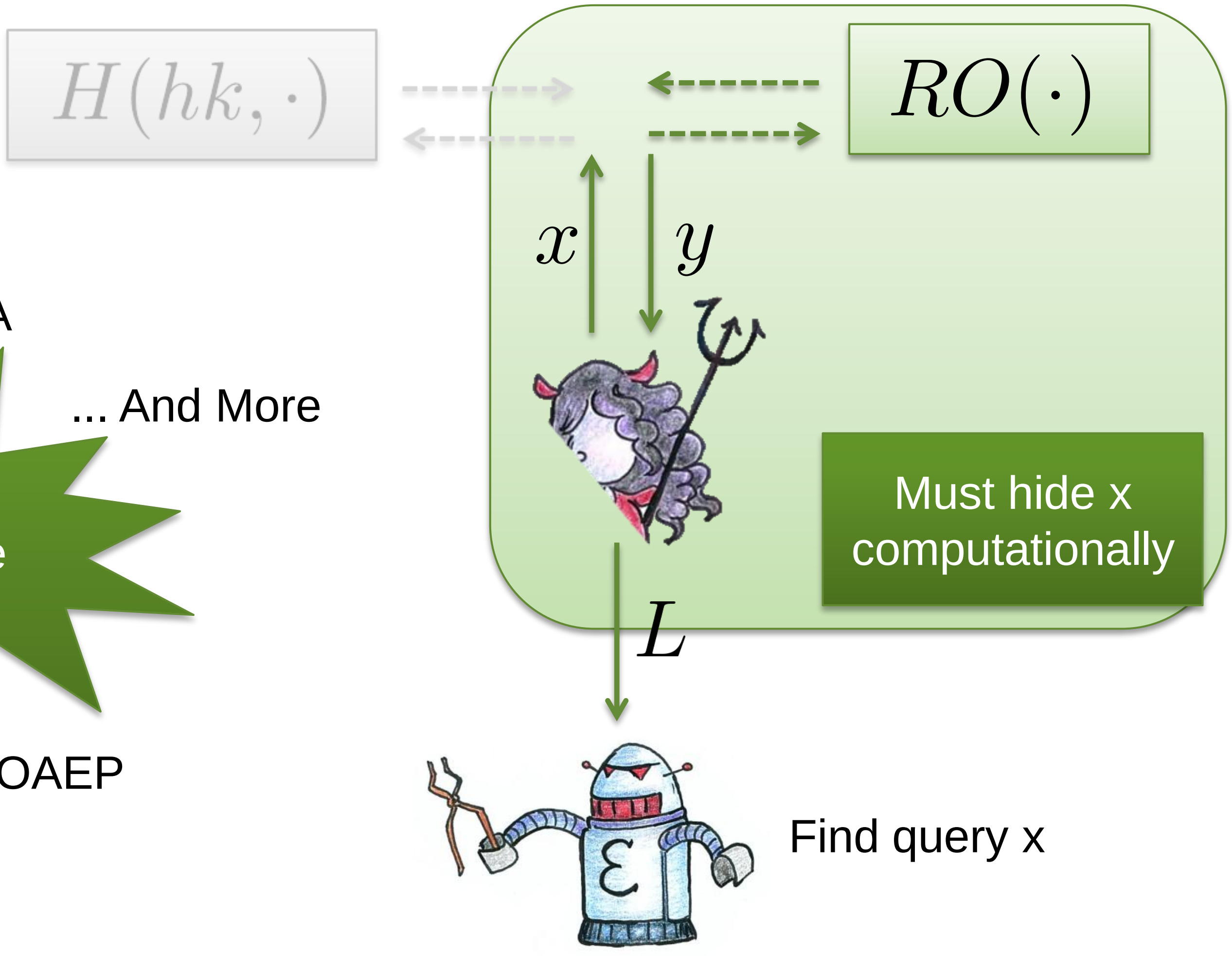


Scheme A
Scheme B
Scheme C
Scheme D
Scheme E
Scheme F
$\mathrm{UCE}[\mathcal{S}^{\mathrm{cup}}]$
$\mathrm{UCE}[\mathcal{S}^{\cdots}]$
$\mathrm{UCE}[\mathcal{S}^{\cdots}]$
Random Oracle Model

UCE Framework [BHK13]

UCE1=UCE[S^{cup}]: Computational Unpredictability





Find query x

UCE vs. iO [BrzuskaFarshimMittelbach14]

$\mathsf{UCE}[\mathcal{S}^{\mathsf{cup}}]$ and indistinguishability obfuscation are mutually exclusive [BFM14]

- Split sources: $\mathsf{UCE}[\mathcal{S}^{\mathsf{splt}}]$
- ~~Bounded Parallel Sources: $\mathsf{UCE}[\mathcal{S}^{\mathsf{prl}}_{\tau,\sigma,q}]$~~ [BFM14]
- Statistical Sources: $\mathsf{UCE}[\mathcal{S}^{\mathsf{sup}}]$
- ...

However, all assumptions validated only in the ROM

What are good UCEs?

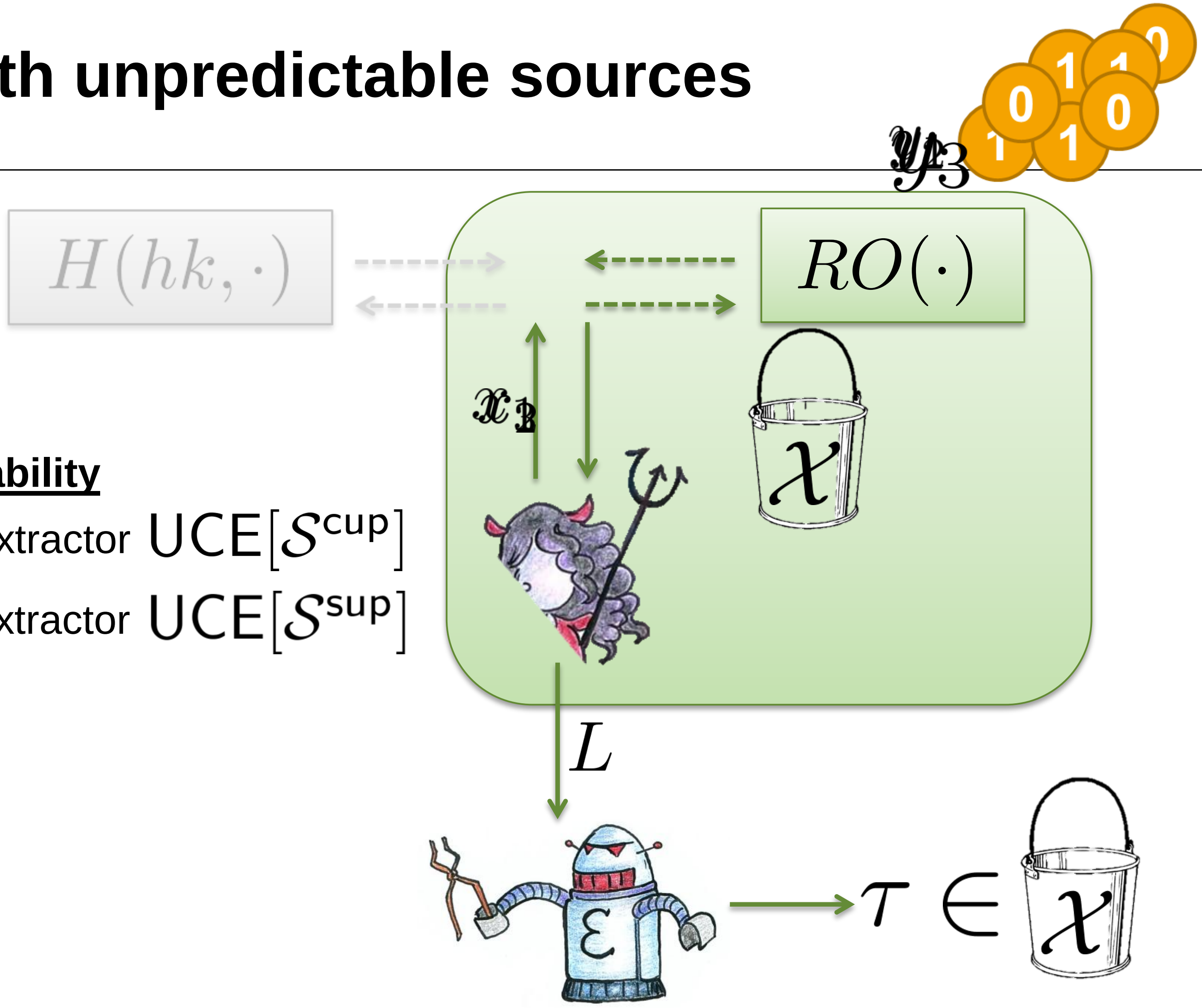
One Definition: Good UCEs are those that strike the right balance between being powerful and feasible.

Candidate Construction in Standard Model

Nice Applications

UCEs with Strongly Unpredictable Sources

UCE with unpredictable sources

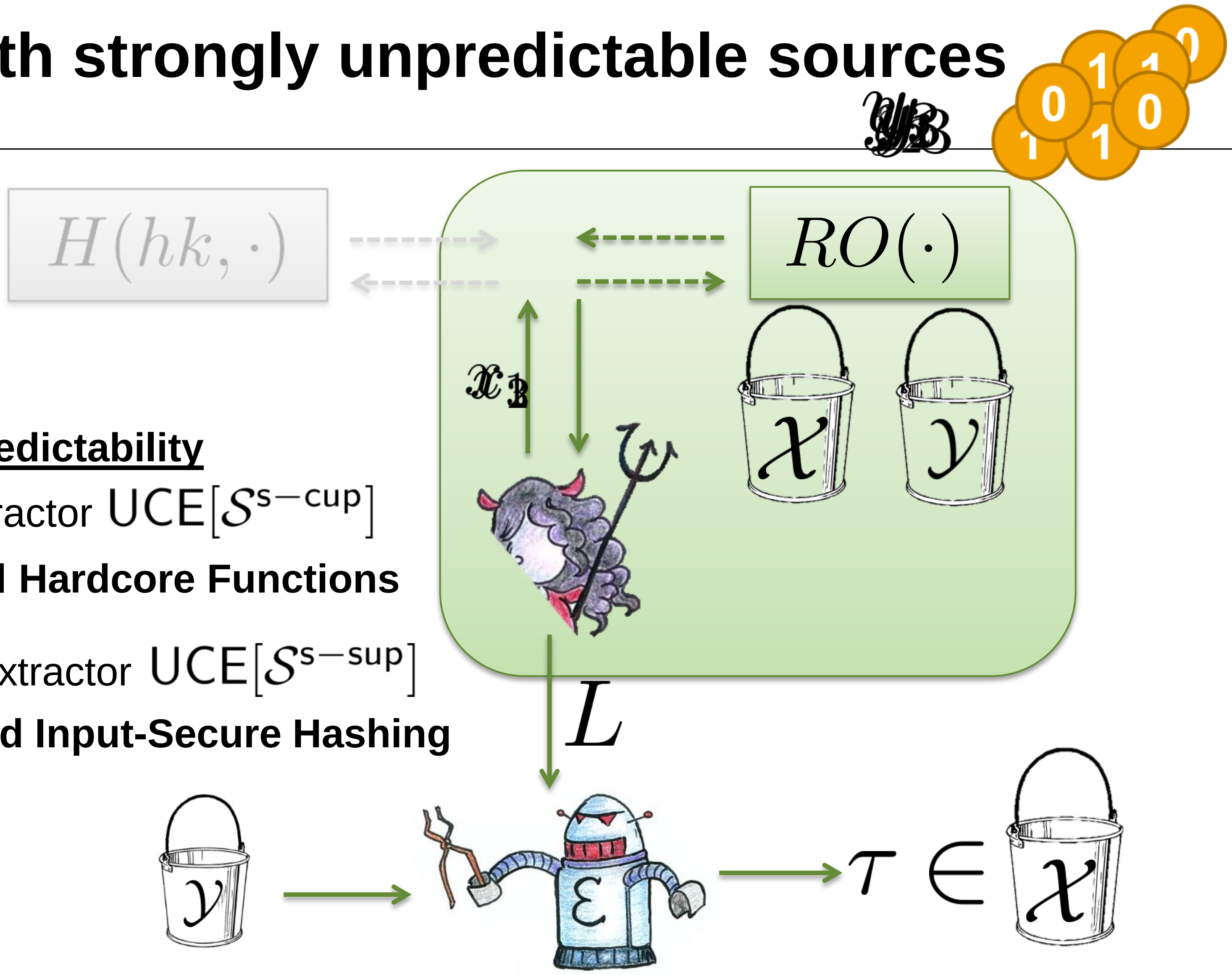


unpredictability

Efficient extractor $\mathsf{UCE}[\mathcal{S}^{\mathsf{cup}}]$

Unbounded extractor $\mathsf{UCE}[\mathcal{S}^{\mathsf{sup}}]$

UCE with strongly unpredictable sources



Strong unpredictability

Efficient extractor $\mathsf{UCE}[\mathcal{S}^{\mathsf{s-cup}}]$

Universal Hardcore Functions

Unbounded extractor $\mathsf{UCE}[\mathcal{S}^{\mathsf{s-sup}}]$

Correlated Input-Secure Hashing

UCEs with strongly unpredictable sources In the standard model

Indistinguishability Obfuscation

Strong Point Obfuscation

Computational unpredictability for single query: $\mathrm{UCE}[\mathcal{S}^{\mathsf{s-cup}} \cap \mathcal{S}^{1-\mathsf{query}}]$

Statistical unpredictability for poly many queries: $\mathrm{UCE}[\mathcal{S}^{\mathsf{s-sup}} \cap \mathcal{S}^{q-\mathsf{query}}]$

The Construction

Indistinguishability Obfuscation

Puncturable Pseudorandom Function

$$\mathsf{iO}\left(\mathsf{PRF}(k,\cdot)\right)$$

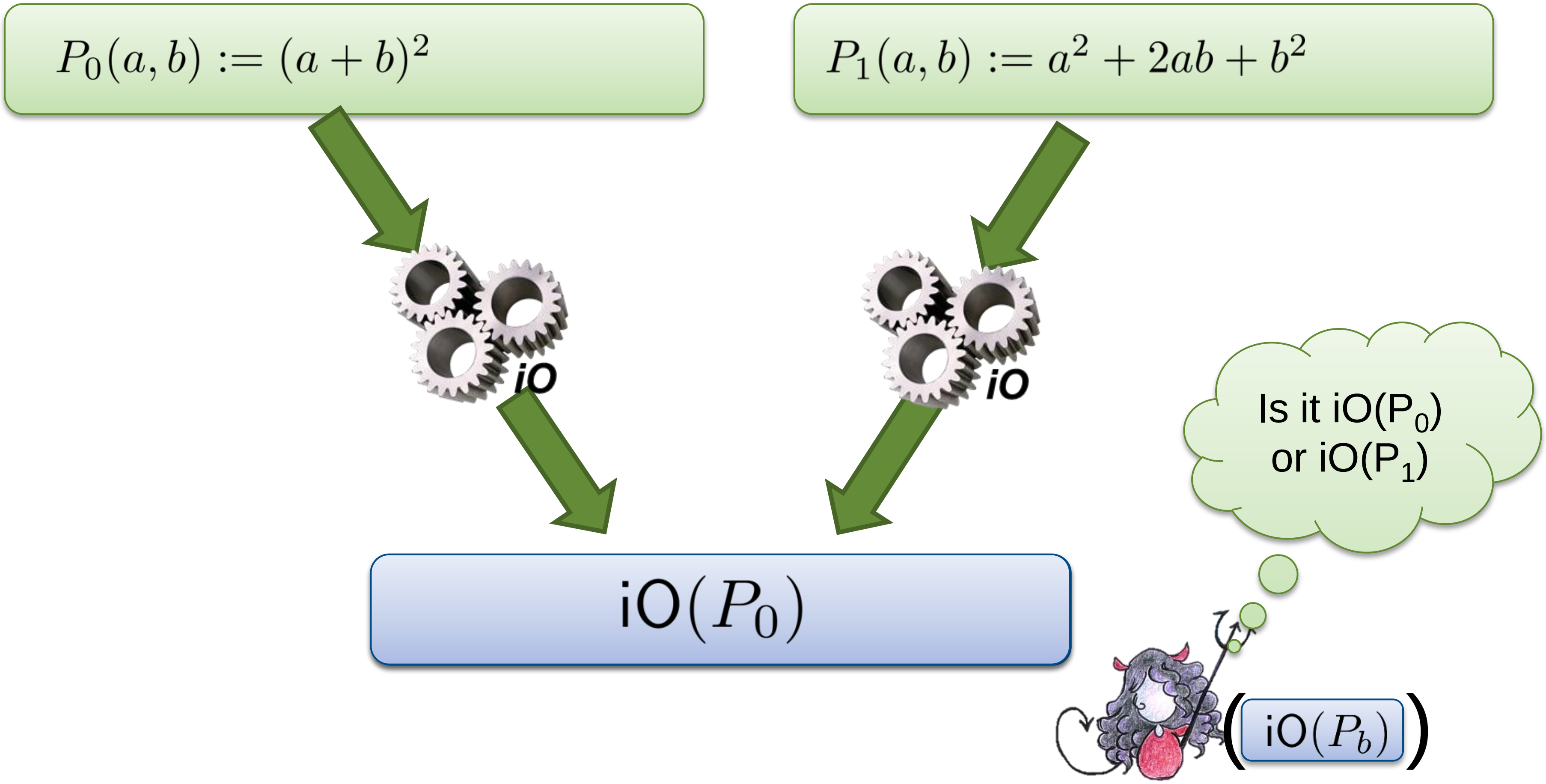
Puncturable Pseudorandom Function

$k^*_{x^*} := \mathsf{puncture}(k, x^*)$

$k^*_{x^*}$ allows to evaluate $\mathsf{PRF}(k,\cdot)$ on all points except for x^*.

$$(k^*_{x^*}, \mathsf{PRF}(k, x^*)) \approx (k^*_{x^*}, \$)$$

Indistinguishability Obfuscation (iO)



The Construction

Indistinguishability Obfuscation

Puncturable Pseudorandom Function

$$\mathsf{iO}\left(\mathsf{PRF}(k,\cdot)\right)$$

[BST14] (previous talk)

The above construction is hardcore for an injective one-way function if padded sufficiently before obfuscation.

The Construction

Indistinguishability Obfuscation

Puncturable Pseudorandom Function

$$\mathsf{iO}\left(\underbrace{\qquad \mathsf{PRF}(k,\cdot) \qquad}_{\text{Pad Before Obfuscation}}\right)$$

Padding depends on number of adversarial queries.

UCEs with strongly unpredictable sources In the standard model

Indistinguishability Obfuscation

Strong Point Obfuscation

Hang On!

- Where is the Point Obfuscation?

The Construction

Hang On!

- Where is the Point Obfuscation?

$$\mathsf{iO}\left(\mathsf{PRF}(k,\cdot)\right)$$

Point Obfuscation

- Only used within the proof
- AIPO: Point obfuscation secure in the presence of auxiliary information

```
$b \leftarrow \{0,1\}$
$(z, x_0) \leftarrow \mathcal{B}_1(1^\lambda)$
$x_1 \leftarrow \{0,1\}^\lambda$
$p \leftarrow \mathsf{AIPO}(x_b)$
$b' \leftarrow \mathcal{B}_2(1^\lambda, p, z)$
return $b = b'$
```

AIPOs have been built from non-standard assumptions [C97,BP12]

z hides x_0 computationally $\Rightarrow \mathsf{UCE}[\mathcal{S}^{\mathsf{s-cup}} \cap \mathcal{S}^{1-\mathsf{query}}]$

z hides x_0 statistically $\Rightarrow \mathsf{UCE}[\mathcal{S}^{\mathsf{s-sup}} \cap \mathcal{S}^{q-\mathsf{query}}]$

Point Obfuscation with iO

A new proof technique

Point obfuscation allows to hide where puncturing takes place.

Point Obfuscation with iO

A new proof technique

1. „Standard Puncturing“ [SW13]

```
$C_1[k](x)$
    return $\mathsf{PRF}(k, x)$
```

```
$C_2[k^* \leftarrow \mathsf{PRF.puncture}(k, x^*),$
    $x^*, y^* \leftarrow \mathsf{PRF}(k, x^*)](x)$
  if $x = x^*$ then
    return $y^*$
  return $\mathsf{PRF}(k^*, x)$
```

```
$C_3[k^* \leftarrow \mathsf{PRF.puncture}(k, x^*),$
    $x^* \boxed{, y^* \leftarrow \$}](x)$
  if $x = x^*$ then
    return $y^*$
  return $\mathsf{PRF}(k^*, x)$
```

```
$C_4\boxed{[k, x^*, y^* \leftarrow \$]}(x)$
  if $x = x^*$ then
    return $y^*$
  return $\mathsf{PRF}(k, x)$
```

$$\begin{tikzcd} C_1 \arrow[d, "\text{iO}"] & C_4 \\ C_2 \arrow[r, "\text{PRF}"] & C_3 \arrow[u, "\text{iO}"] \end{tikzcd}$$

Point Obfuscation with iO

A new proof technique

2. „Hide Punctured Point“

```
$C_4[k, x^*, y^* \leftarrow \$](x)$
    if $x = x^*$ then
        return $y^*$
    return $\mathsf{PRF}(k, x)$
```

```
$C_5[k, \boxed{p_{x^*} \leftarrow \mathsf{(AI)PO}(x^*)}, y^* \leftarrow \$](x)$
    if $\boxed{p_{x^*}(x) \neq \bot}$ then
        return $y^*$
    return $\mathsf{PRF}(k, x)$
```

```
$C_6[k](x)$
    return $\mathsf{PRF}(k, x)$
```

$$\begin{tikzcd}
C_4 \arrow[d, "\text{iO}"] & \\
C_5 \arrow[r, bend left, "\text{PO+iO+[BCP14]}"] \arrow[r, "\text{diO}"'] & C_6
\end{tikzcd}$$

Proof Overview

$$\begin{tikzcd}
\mathsf{Game}_1(\lambda) \arrow[r, bend left, "\mathsf{iO}"] & \mathsf{Game}_2(\lambda) \arrow[r, bend left, "\mathsf{PRF}"] & \mathsf{Game}_3(\lambda) \arrow[r, bend left, "\mathsf{iO}"] & \mathsf{Game}_4(\lambda) \arrow[r, bend left, "\mathsf{AIPO} + \mathsf{iO} + \text{[BCP14]}"] & \mathsf{Game}_5(\lambda)
\end{tikzcd}$$

```
$\mathsf{Game}_1(\lambda)$
  $x^*, y^* \leftarrow \perp$
  $k \leftarrow\$ G.\mathsf{KGen}(1^\lambda)$
  $L \leftarrow\$ \mathsf{S}^{\textsc{Hash}}(1^\lambda)$
  $\mathsf{hk} \leftarrow\$ \mathsf{iO}(C_1[k])$
  $b' \leftarrow\$ \mathsf{D}(1^\lambda, \mathsf{hk}, L)$
  return $(1 = b')$

$\textsc{Hash}(x)$
  $x^* \leftarrow x$
  $y^* \leftarrow G.\mathsf{Eval}(k, x)$
  return $y^*$
```

```
$\mathsf{Game}_2(\lambda)$
  $x^*, y^* \leftarrow \perp$
  $k \leftarrow\$ G.\mathsf{KGen}(1^\lambda)$
  $L \leftarrow\$ \mathsf{S}^{\textsc{Hash}}(1^\lambda)$
  $p \leftarrow\$ \mathsf{AIPO}(x^*)$
  $k^* \leftarrow G.\mathsf{Puncture}(k, x^*)$
  $\mathsf{hk} \leftarrow\$ \mathsf{iO}(C_2[k^*, p, y^*])$
  $b' \leftarrow\$ \mathsf{D}(1^\lambda, \mathsf{hk}, L)$
  return $(1 = b')$

$\textsc{Hash}(x)$
  $x^* \leftarrow x$
  $y^* \leftarrow G.\mathsf{Eval}(k, x)$
  return $y^*$
```

```
$\mathsf{Game}_3(\lambda)$
  $x^*, y^* \leftarrow \perp$
  $k \leftarrow\$ G.\mathsf{KGen}(1^\lambda)$
  $L \leftarrow\$ \mathsf{S}^{\textsc{Hash}}(1^\lambda)$
  $p \leftarrow\$ \mathsf{AIPO}(x^*)$
  $k^* \leftarrow G.\mathsf{Puncture}(k, x^*)$
  $\mathsf{hk} \leftarrow\$ \mathsf{iO}(C_2[k^*, p, y^*])$
  $b' \leftarrow\$ \mathsf{D}(1^\lambda, \mathsf{hk}, L)$
  return $(1 = b')$

$\textsc{Hash}(x)$
  $x^* \leftarrow x$
  $y^* \leftarrow\$ \{0,1\}^{\mathsf{H.ol}(\lambda)}$
  return $y^*$
```

```
$\mathsf{Game}_4(\lambda)$
  $x^*, y^* \leftarrow \perp$
  $k \leftarrow\$ G.\mathsf{KGen}(1^\lambda)$
  $L \leftarrow\$ \mathsf{S}^{\textsc{Hash}}(1^\lambda)$
  $p \leftarrow\$ \mathsf{AIPO}(x^*)$
  $\mathsf{hk} \leftarrow\$ \mathsf{iO}(C_3[k, p, y^*])$
  $b' \leftarrow\$ \mathsf{D}(1^\lambda, \mathsf{hk}, L)$
  return $(1 = b')$

$\textsc{Hash}(x)$
  $x^* \leftarrow x$
  $y^* \leftarrow\$ \{0,1\}^{\mathsf{H.ol}(\lambda)}$
  return $y^*$
```

```
$\mathsf{Game}_5(\lambda)$
  $x^*, y^* \leftarrow \perp$
  $k \leftarrow\$ G.\mathsf{KGen}(1^\lambda)$
  $L \leftarrow\$ \mathsf{S}^{\textsc{Hash}}(1^\lambda)$
  $\mathsf{hk} \leftarrow\$ \mathsf{iO}(C_4[k])$
  $b' \leftarrow\$ \mathsf{D}(1^\lambda, \mathsf{hk}, L)$
  return $(1 = b')$

$\textsc{Hash}(x)$
  $x^* \leftarrow x$
  $y^* \leftarrow\$ \{0,1\}^{\mathsf{H.ol}(\lambda)}$
  return $y^*$
```

```
$\textsc{Circuit}\ C_1[k](x)$
  return $G.\mathsf{Eval}(k, x)$
```

```
$\textsc{Circuit}\ C_2[k^*, p, y^*](x)$
  if $p(x) = \perp$ then    // if $x \neq x^*$
    return $G.\mathsf{Eval}(k^*, x)$
  return $y^*$
```

```
$\textsc{Circuit}\ C_3[k, p, y^*](x)$
  if $p(x) = \perp$ then
    return $G.\mathsf{Eval}(k, x)$
  return $y^*$
```

```
$\textsc{Circuit}\ C_4[k](x)$
  return $G.\mathsf{Eval}(k, x)$
```

Summary

- Propose UCE with strong unpredictability
  - statistical $\mathsf{UCE}[\mathcal{S}^{\mathsf{s-sup}}] \longrightarrow$ correlated input security
  - computational $\mathsf{UCE}[\mathcal{S}^{\mathsf{s-cup}}] \longrightarrow$ hardcore functions

- Standard Model Constructions from iO and AIPO
  - $\mathsf{UCE}[\mathcal{S}^{\mathsf{s-cup}} \cap \mathcal{S}^{1-\mathsf{query}}] \longrightarrow$ (universal) hardcore functions
  - $\mathsf{UCE}[\mathcal{S}^{\mathsf{s-sup}} \cap \mathcal{S}^{q-\mathsf{query}}] \longrightarrow$ q-query correlated input secure hashes

- New iO proof technique: use Point Obfuscation

  Extension of punctured programs technique to hide punctured point