Public-Seed Pseudorandom Permutations

Stefano Tessaro
UCSB

Joint work with **Pratik Soni** (UCSB)

DIMACS Workshop
New York
June 8, 2017

DIMACS Workshop on Complexity of Cryptographic Primitives and Assumptions

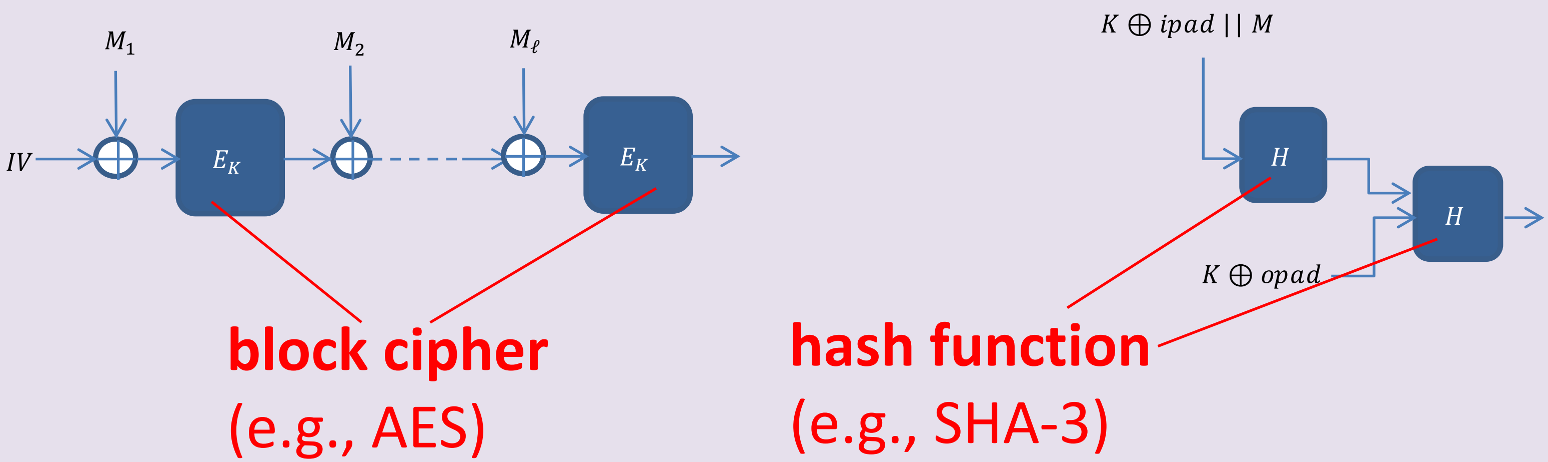
We look at existing class of cryptographic primitives and introduce/study the first "plausible" assumptions on them.

Pratik Soni, Stefano Tessaro

Public-Seed Pseudorandom Permutations

EUROCRYPT 2017

Cryptographic schemes often built from simpler **building blocks**

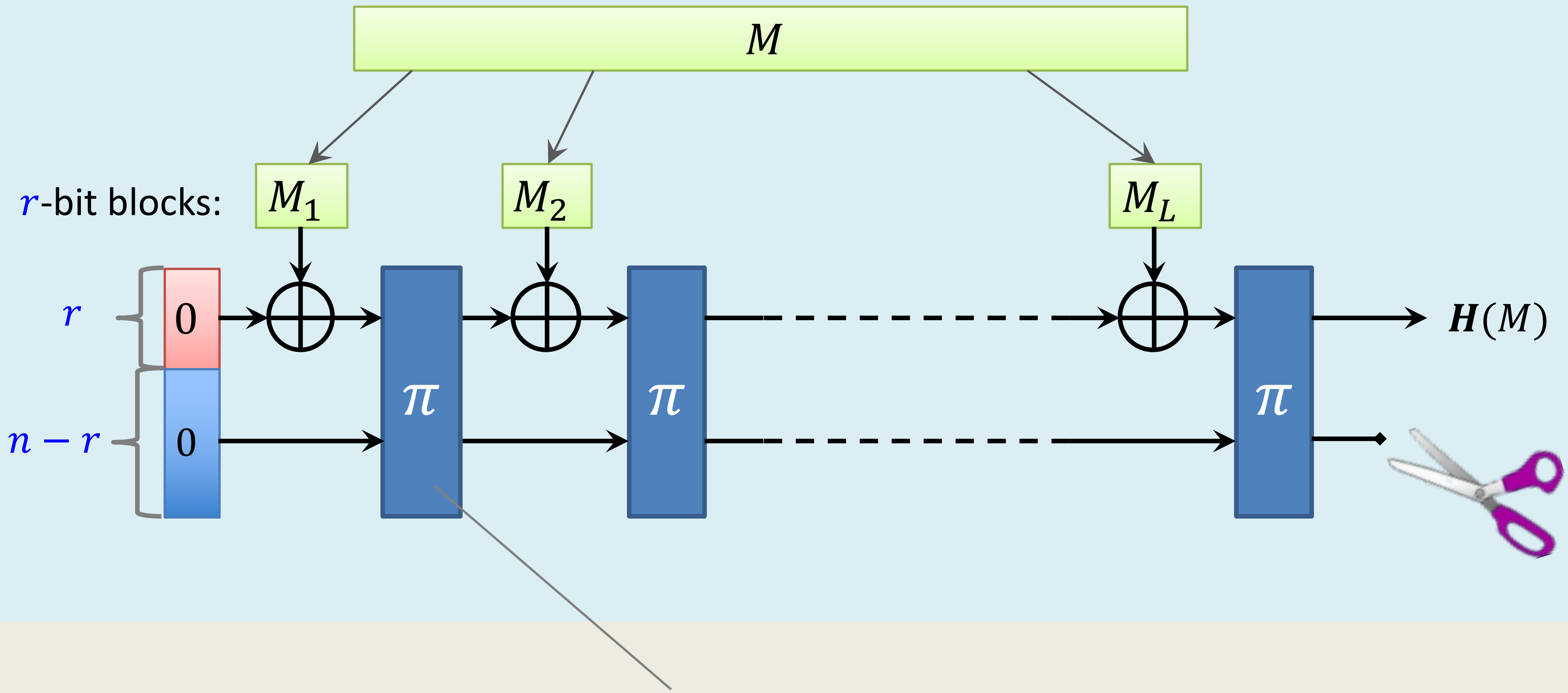


Is there a **universal** and simple building block for efficient symmetric cryptography?

Main motivation: Single object requiring optimized implementation!

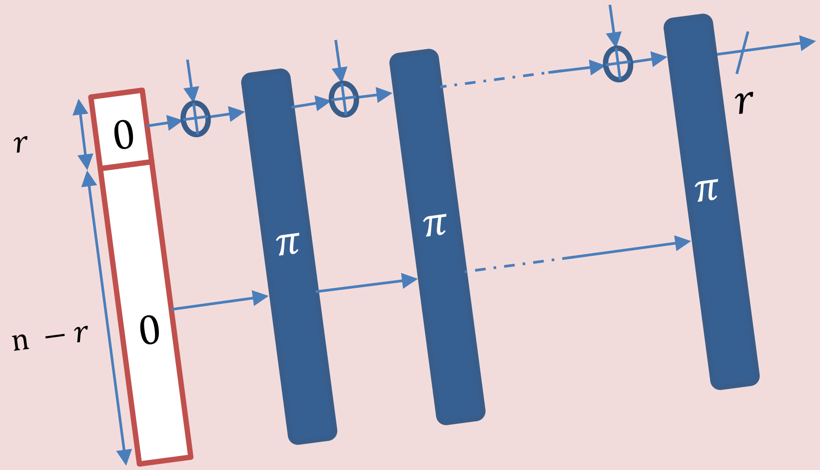
Recent trend:  = permutation

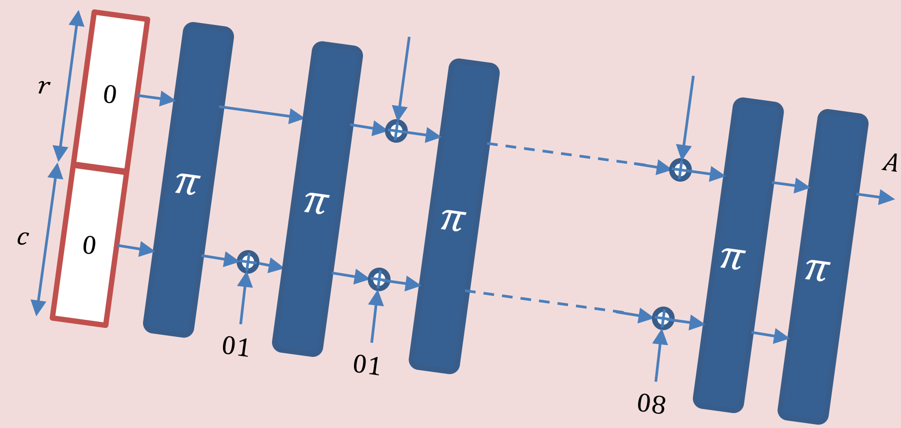
Example. Sponge construction (as in SHA-3) [BDPvA]

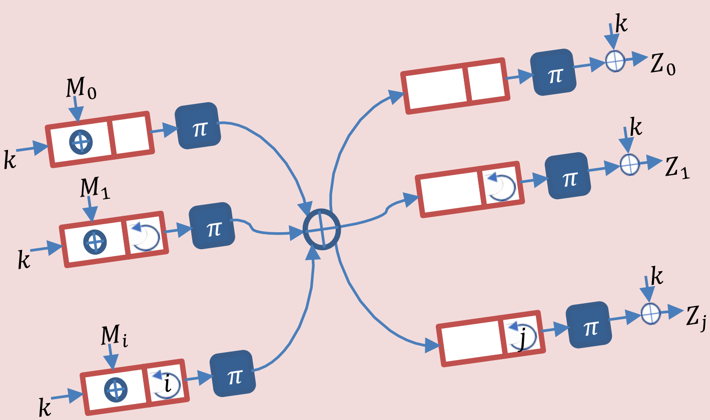


efficiently computable and invertible **permutation**

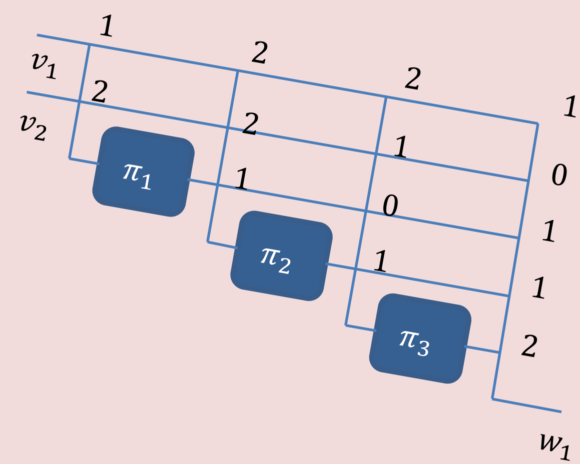
Several permutation-based constructions





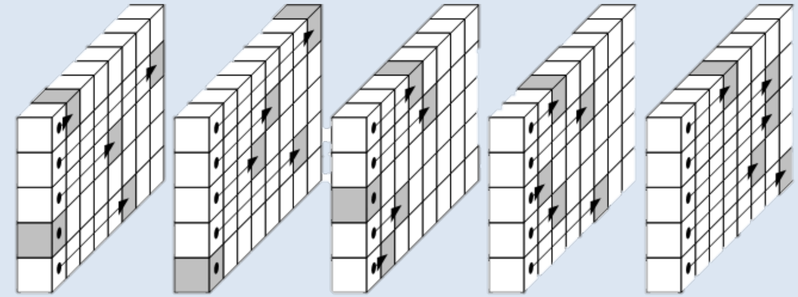


...



Hash functions, authenticated encryption schemes, PRNGs, garbling schemes …

Permutation instantiations



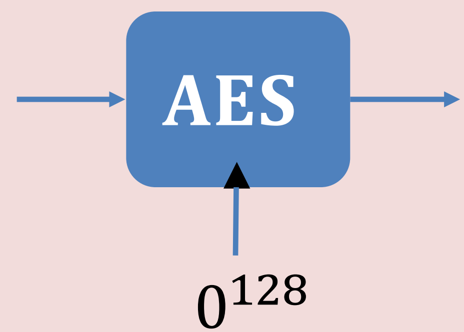
Ad-hoc designs

e.g., in SHA-3, AE schemes, …

Designed to withstand cryptanalytic attacks against constructions using them! e.g., no collision attack

Fixed-key block ciphers

e.g., $\pi : x \mapsto \mathrm{AES}(0^{128}, x)$



Faster hash functions [RS08], fast garbling [BHKR13]

Permutations assumptions

What security properties do we expect from a permutation?

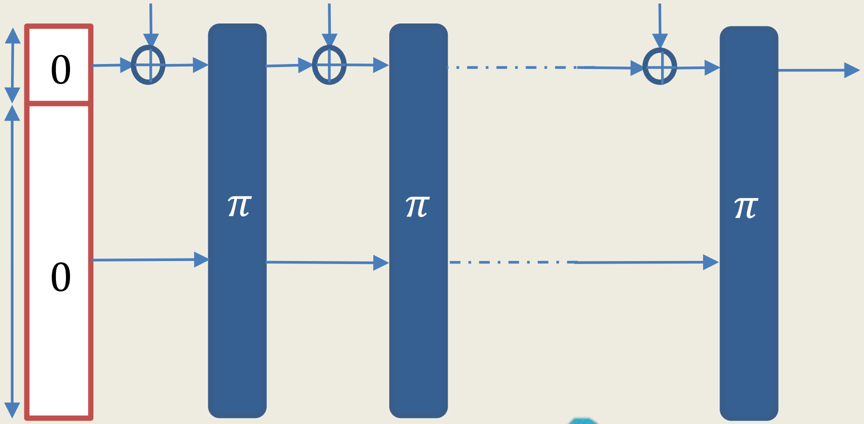
Ideal goal: Standard-model reduction!

"If π satisfies X then $C[\pi]$ satisfies Y."

e.g., $C = \mathrm{SHA{-}3}$;

$Y =$ Anything non-trivial

$X = ???$



Unfortunately: No standard-model proofs known under non-tautological assumptions!

Security of permutation-based crypto

Provable security

Random permutation model!

π is random + adversary given oracle access to π and π^{-1}

clearly unachievable **[CGH98]** …

… security against <u>generic</u> attacks!

Cryptanalysis

<u>Application</u> specific attacks

Insights are hard to recycle for new applications

Very little permutation-specific cryptanalysis

Example – OWFs from permutations

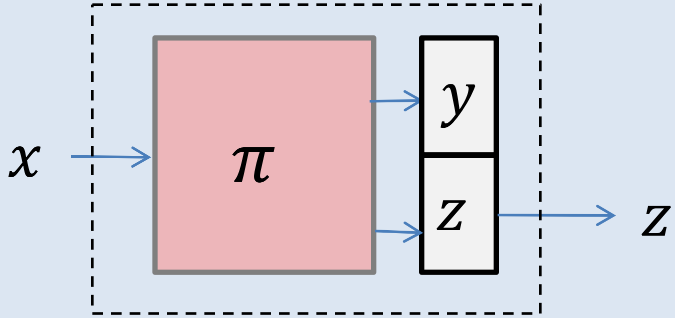
$\pi: \{0,1\}^n \to \{0,1\}^n$

$$x \to \boxed{\pi} \to y = \pi(x) \to \boxed{\pi^{-1}} \to \pi^{-1}(y)$$

Clearly: Cannot be one way!

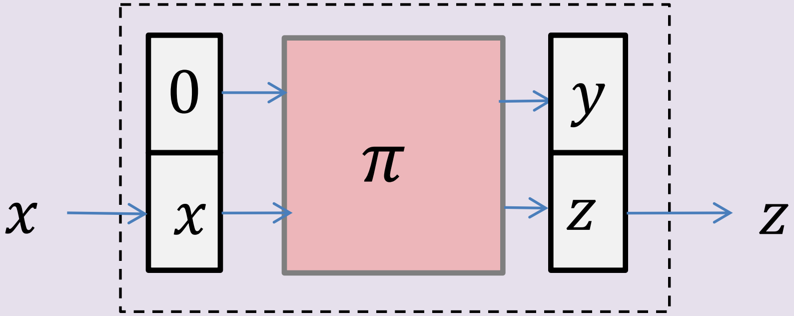
So, how do we make a one-way function out of π?

Naïve idea: **Truncation** $f: \{0,1\}^n \rightarrow \{0,1\}^{n/2}$



Not one way:
$\forall y: \pi^{-1}(y, z)$ preimage of z

Better candidate: $g: \{0,1\}^{n/2} \rightarrow \{0,1\}^{n/2}$



Conjectured one-way for π = SHA-3 permutation

Wanted: Basic (succinct, non-tautological) security property satisfied by π which implies one-wayness of g?

Permutations vs hash functions

	ideal model	standard model
Hash functions	random oracle	CRHF, OWFs, UOWHFs, CI, UCEs…
Permutations	random permutation	???

What kind of cryptographic hardness can we expect from a permutation?

This work, in a nutshell

First plausible and **useful** standard-model security assumption for permutations.

"Public-seed Pseudorandom Permutations" (psPRPs)

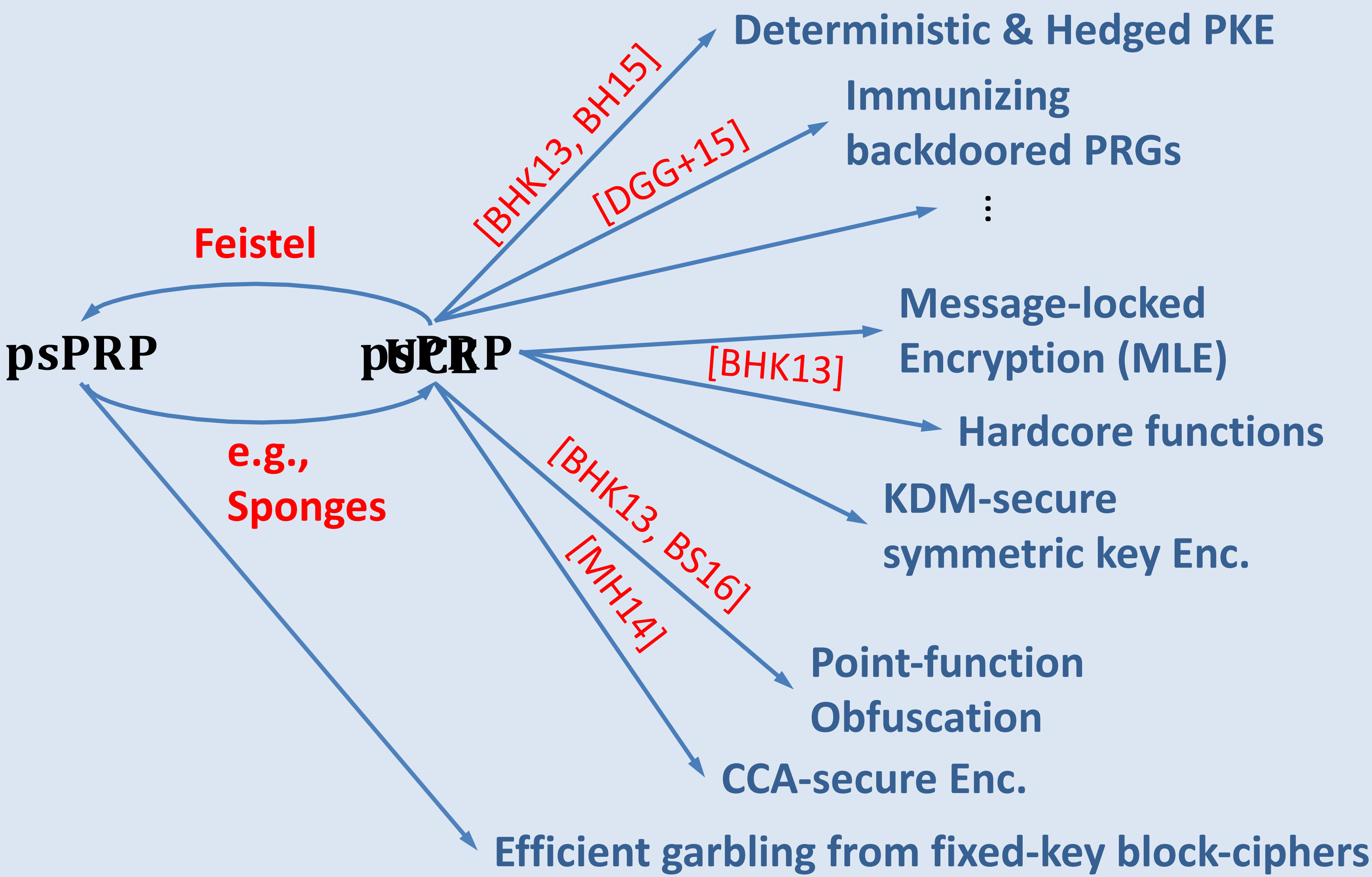
inspired by the UCE framework [BHK13]

Two main questions:

Can we get psPRPs at all?

Are psPRPs useful?

psPRPs – Landscape preview



Roadmap

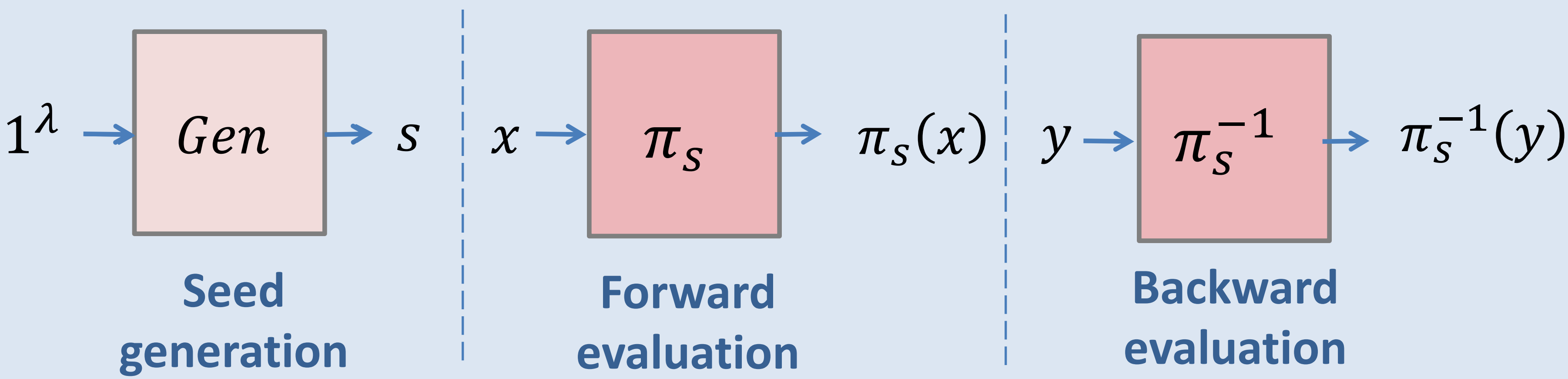
1. <u>Definitions</u>

2. Constructions & Applications

3. Conclusions

Syntax: Seeded permutations

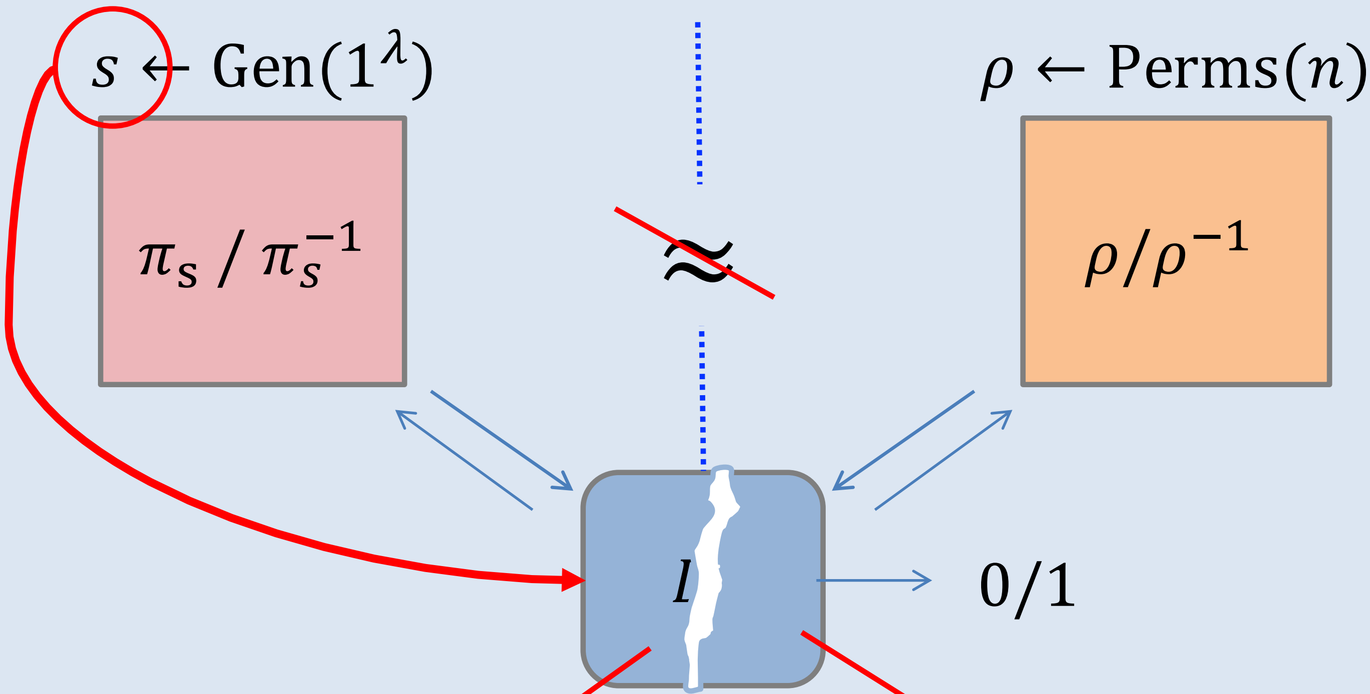
$$\cancel{\pi : \{0,1\}^n \to \{0,1\}^n} \quad \Longrightarrow \quad P = (Gen, \pi, \pi^{-1})$$



(1) $\pi_s : \{0,1\}^n \to \{0,1\}^n$

(2) $\forall x : \pi_s^{-1}(\pi_s(x)) = x$

Secret-seed security: **Pseudorandom permutations (PRPs)**



<u>Stage 1:</u>
- Oracle access
- Secret seed

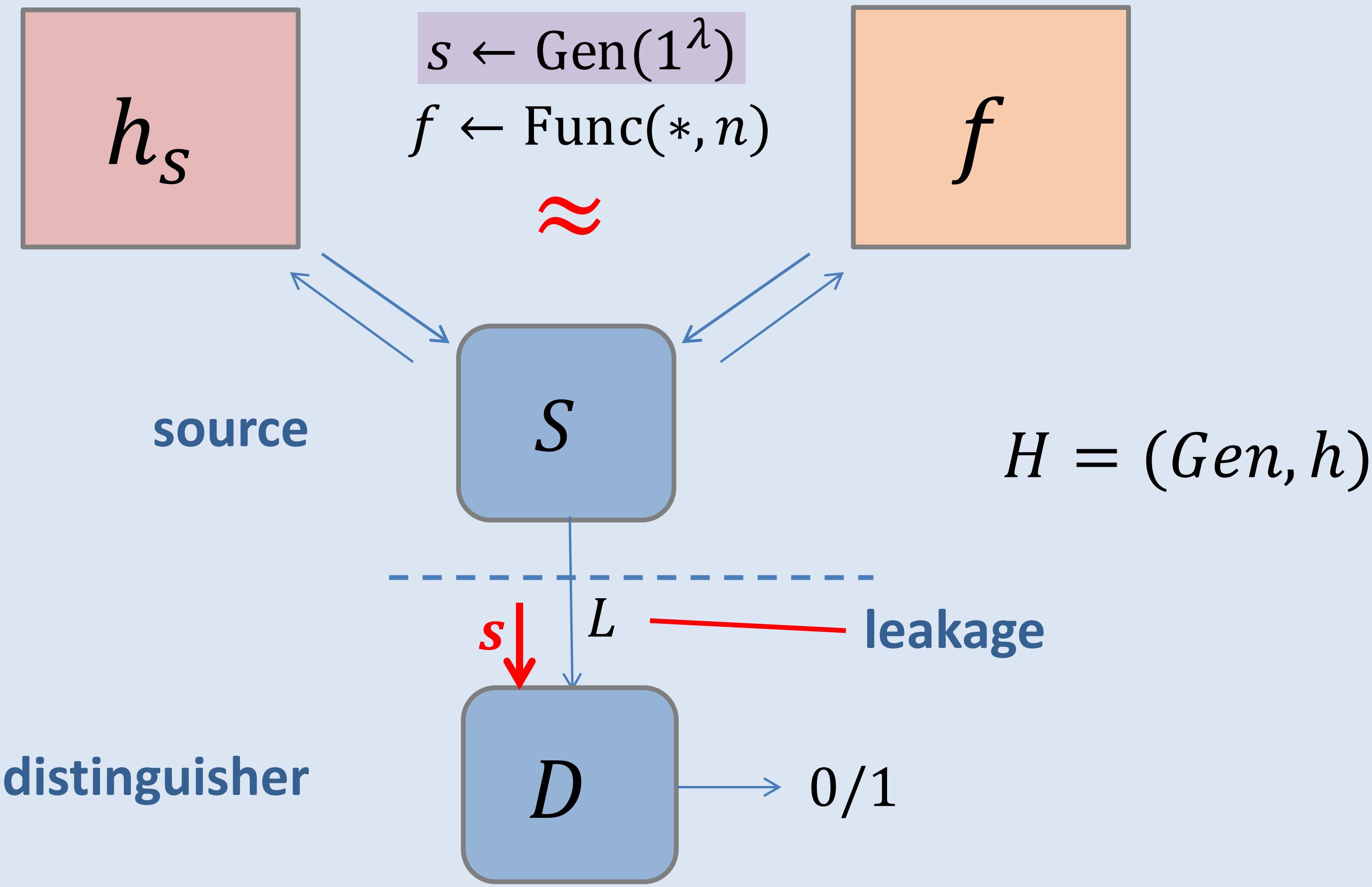
Limited information flow

<u>Stage 2:</u>
- Learns seed
- No oracle access

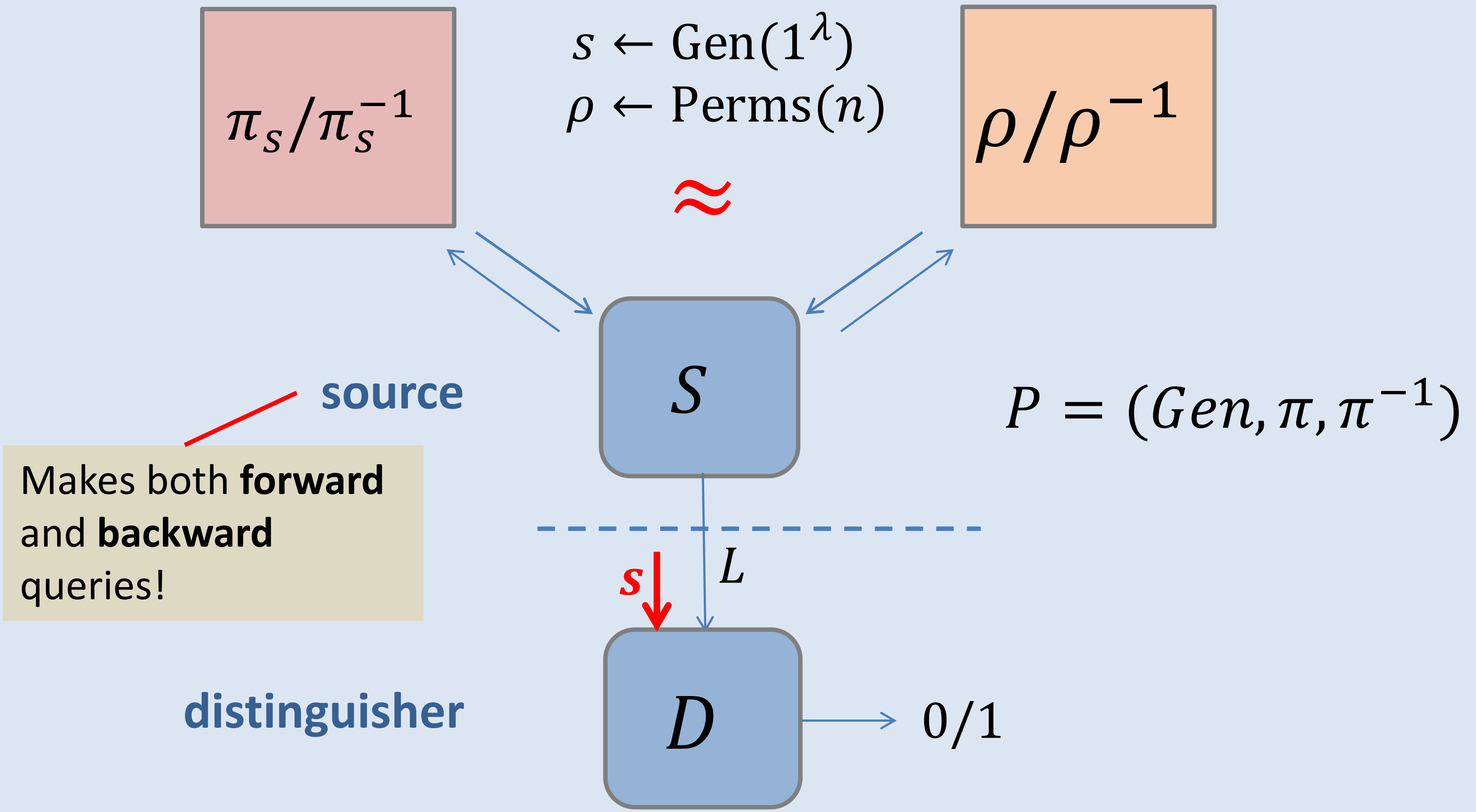
UCE security



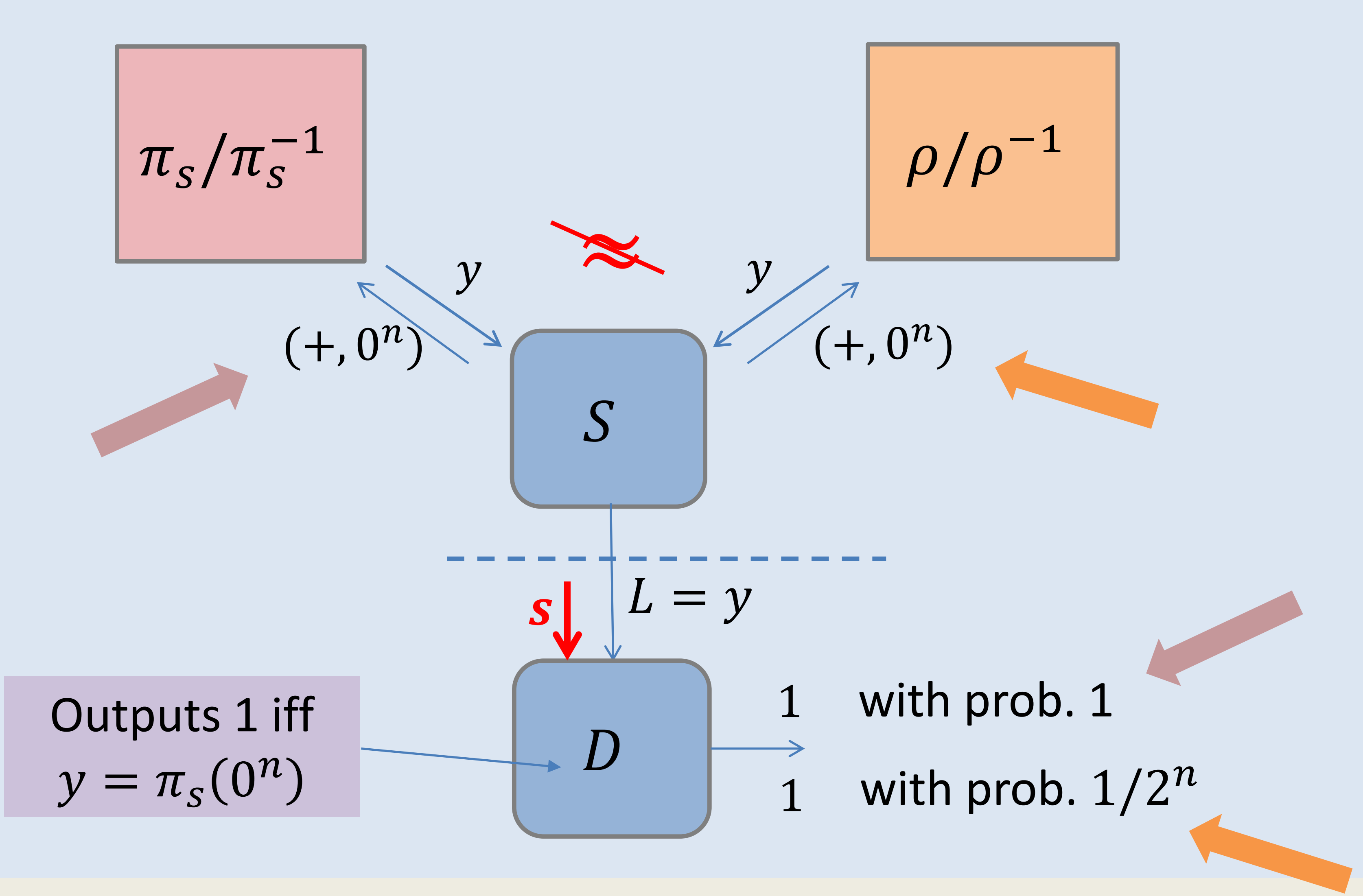
Bellare Hoang Keelveedh



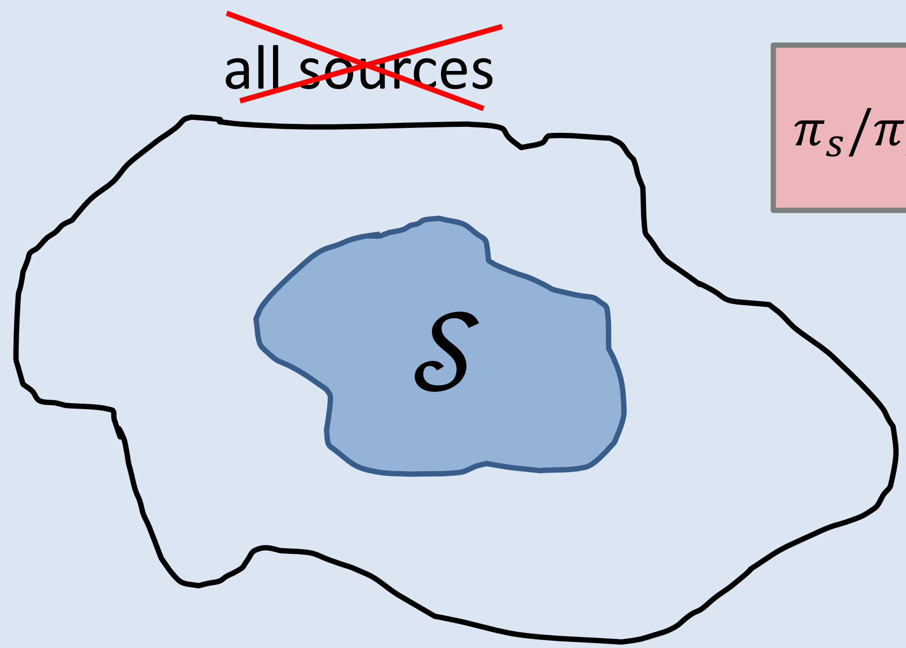
psPRP security [This work]

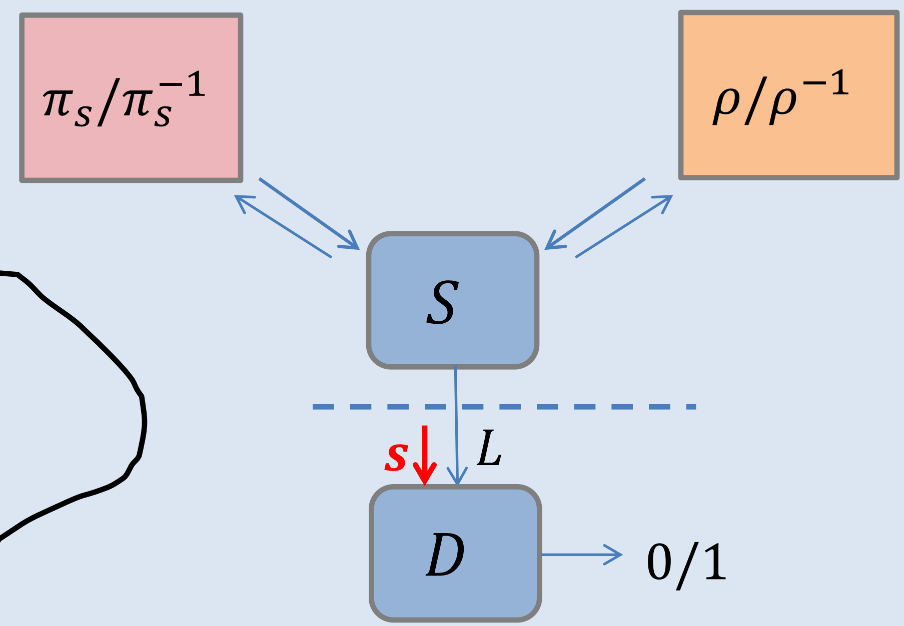


Observation: psPRP-security impossible against all PPT sources!



Solution: Restrict class of considered sources!

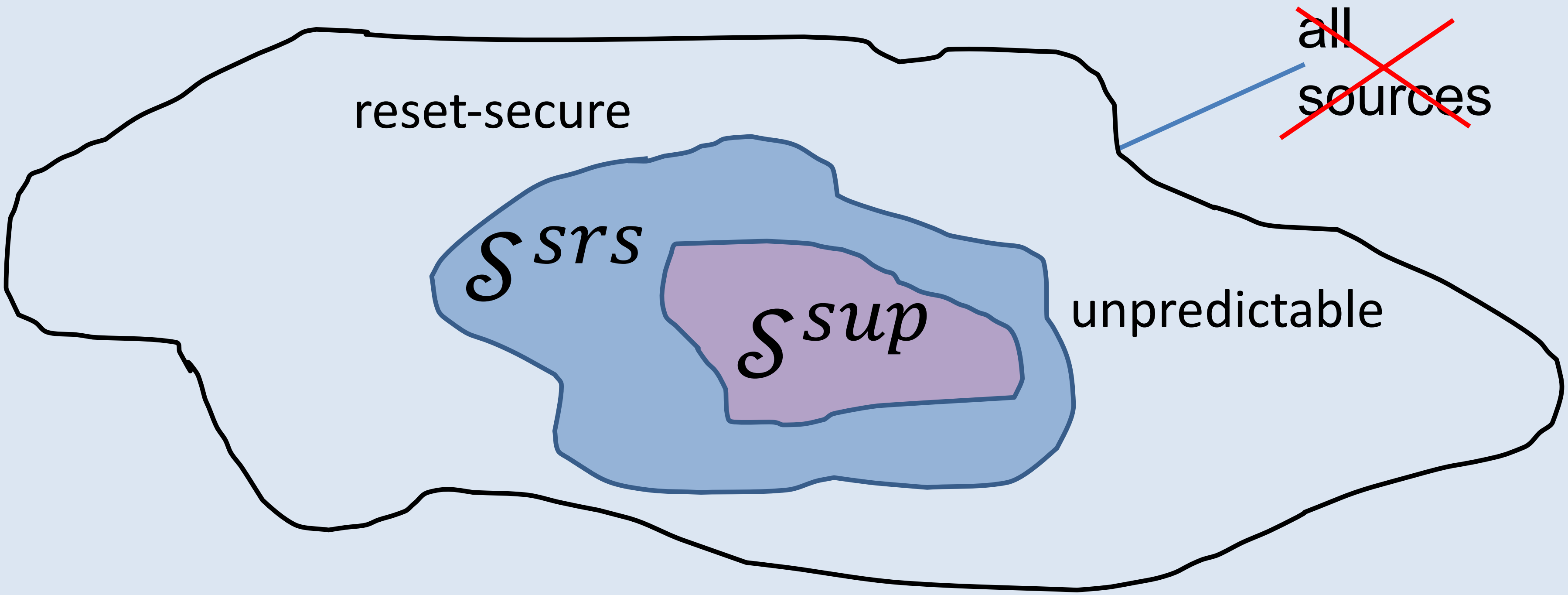




Definition. P **psPRP**$[\mathcal{S}]$**-secure**: $\forall\, S \in \mathcal{S}$, $\forall$PPT D:

$$\pi_s/\pi_s^{-1} \approx \rho/\rho^{-1}$$

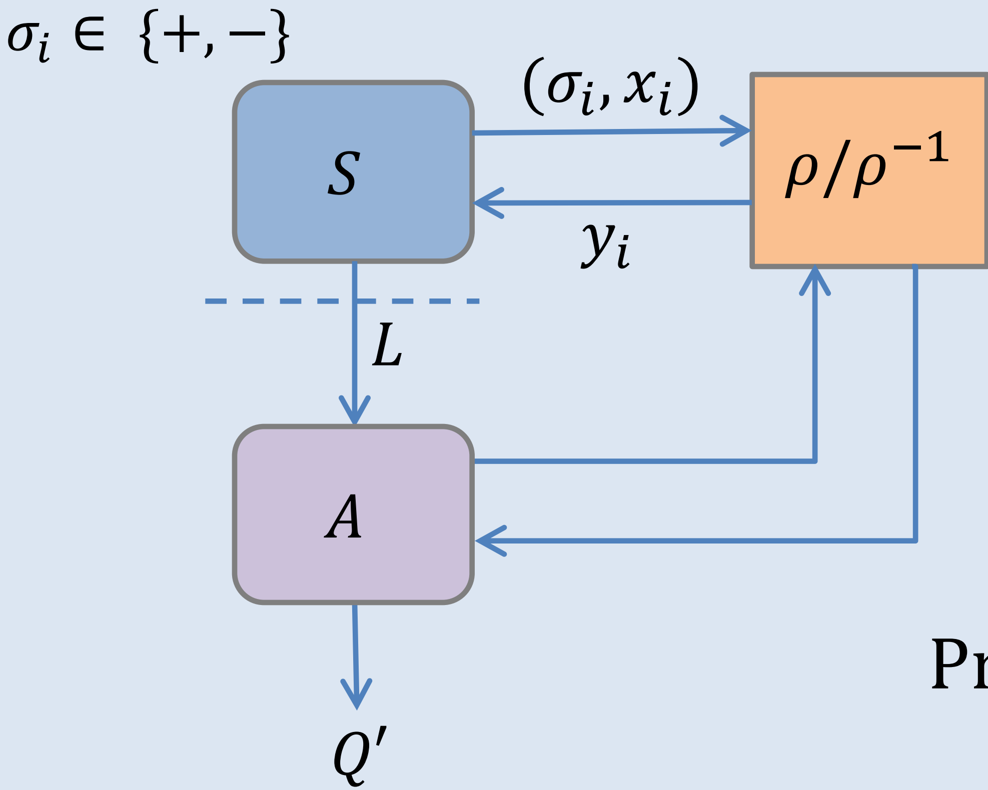
Here: unpredictable and reset-secure sources



Both restrictions capture unpredictability of source queries!

$\mathcal{S}^{sup} \subseteq \mathcal{S}^{srs} \implies$ $\mathbf{psPRP}[\mathcal{S}^{srs}]$ **stronger** assumption than $\mathbf{psPRP}[\mathcal{S}^{sup}]$

Source restrictions – unpredictability



$$Q \leftarrow Q \cup \{x_i, y_i\}$$

Goal: Must be hard for A to predict S's queries or their inverses

$$\Pr[\,Q' \cap Q \neq \phi] = \mathrm{negl}(\lambda)$$

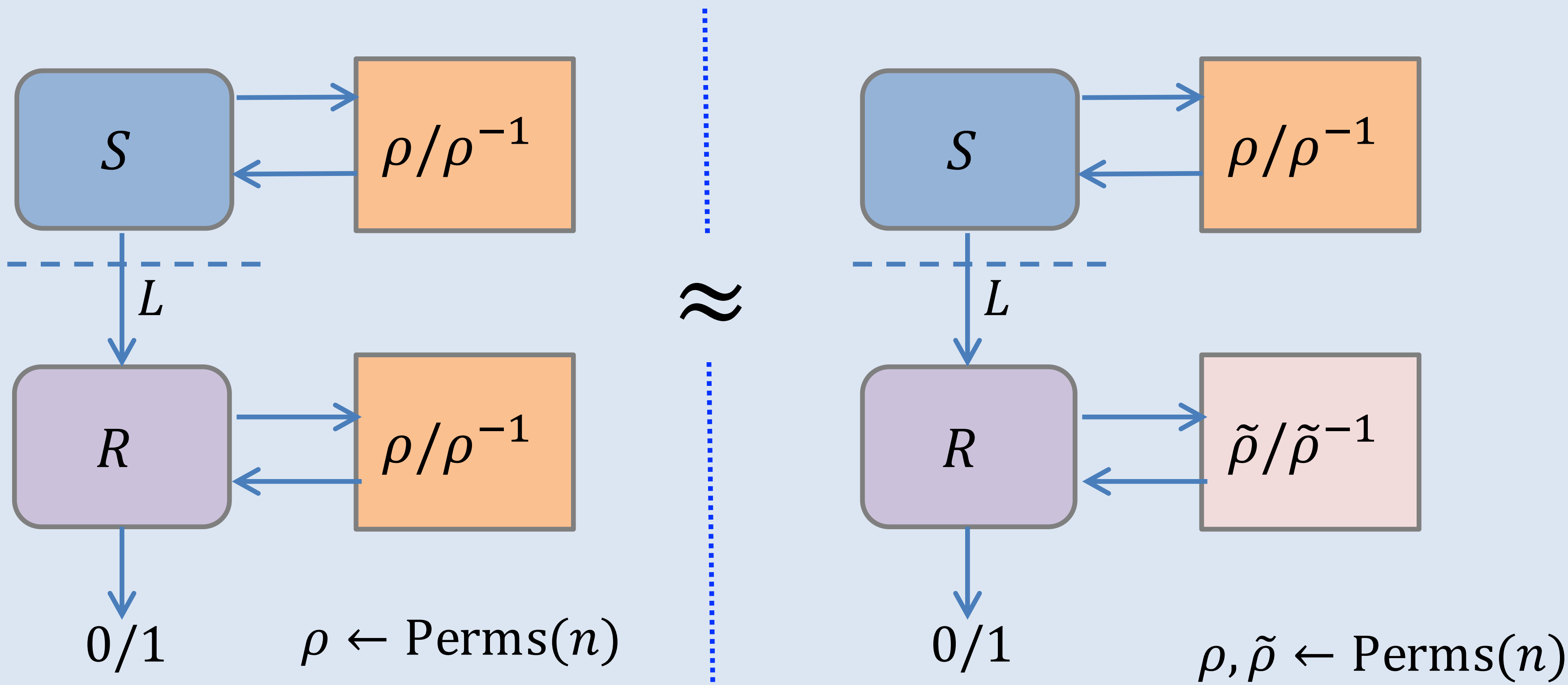
$\mathcal{S}^{sup}$: A is computationally unbounded, poly queries

$\subseteq$

~~$\mathcal{S}^{cup}$: A is PPT~~

$\boxed{\textbf{iO} \Longrightarrow \textbf{psPRP}[\mathcal{S}^{cup}] \text{ impossible [BFM14]}}$

Source restrictions – reset-security



$\mathcal{S}^{srs}$: R is computationally unbounded, poly queries

$\cup\!\!\!\!\!\!\mid$

~~$\mathcal{S}^{crs}$: R is PPT~~

Fact. $\mathcal{S}^{sup} \subseteq \mathcal{S}^{srs}$

Recap – Definitions

$$\mathbf{psPRP}[\mathcal{S}^{srs}] \downarrow \mathbf{psPRP}[\mathcal{S}^{sup}]$$

Equally useful?

$$\mathbf{UCE}[\mathcal{S}^{srs}] \downarrow \mathbf{UCE}[\mathcal{S}^{sup}]$$

Central assumptions in UCE theory

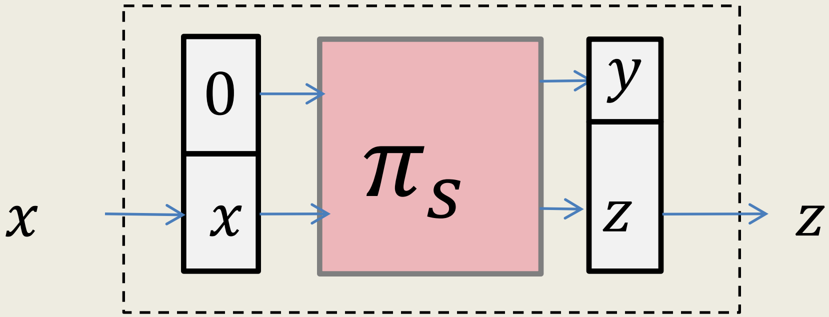
Roadmap

1. Definitions

2. Constructions & Applications

3. Conclusions

Example – Truncation



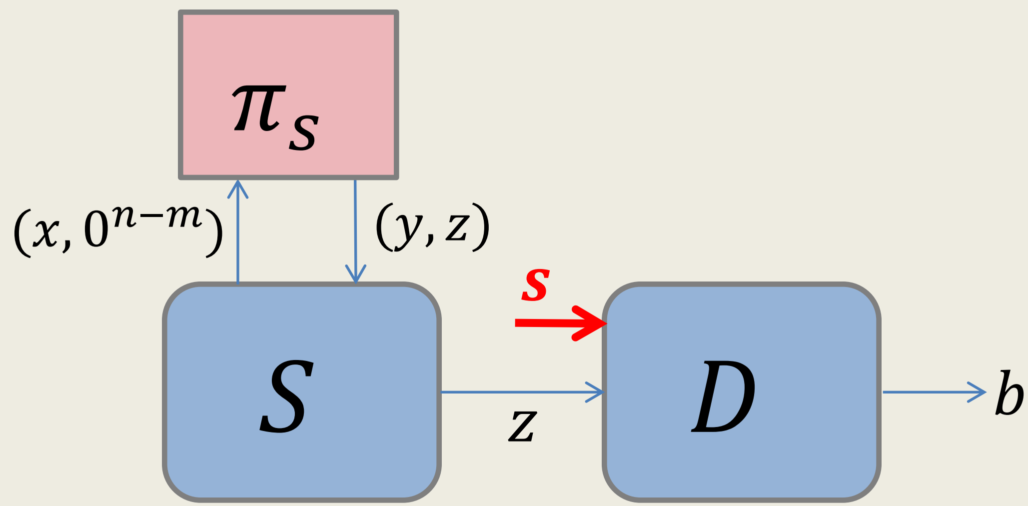
$$g_s\colon \{0,1\}^m \to \{0,1\}^k$$

$$g_s(x) = \pi_s(x, 0^{n-m})[1..k]$$

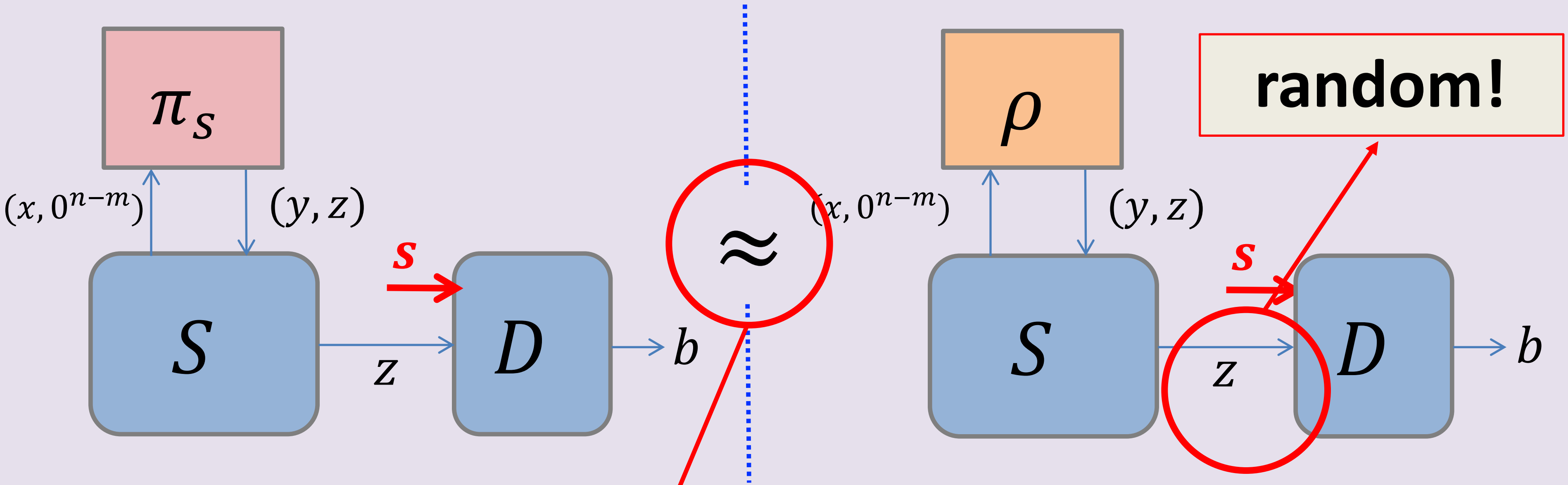
Lemma. If π $\mathbf{psPRP}[\mathcal{S}^{sup}]$-secure and $m + \omega(\log\lambda) \leq k \leq n - \omega(\log\lambda)$, then g is **PRG.**

Thus, also a **OWF** ...

$s \leftarrow \mathrm{Gen}(1^\lambda)$
$x \leftarrow \{0,1\}^{n-m}$
$(y, z) \leftarrow \pi_s(x, 0)$
$b \leftarrow D(s, z)$



Proof – Cont'd

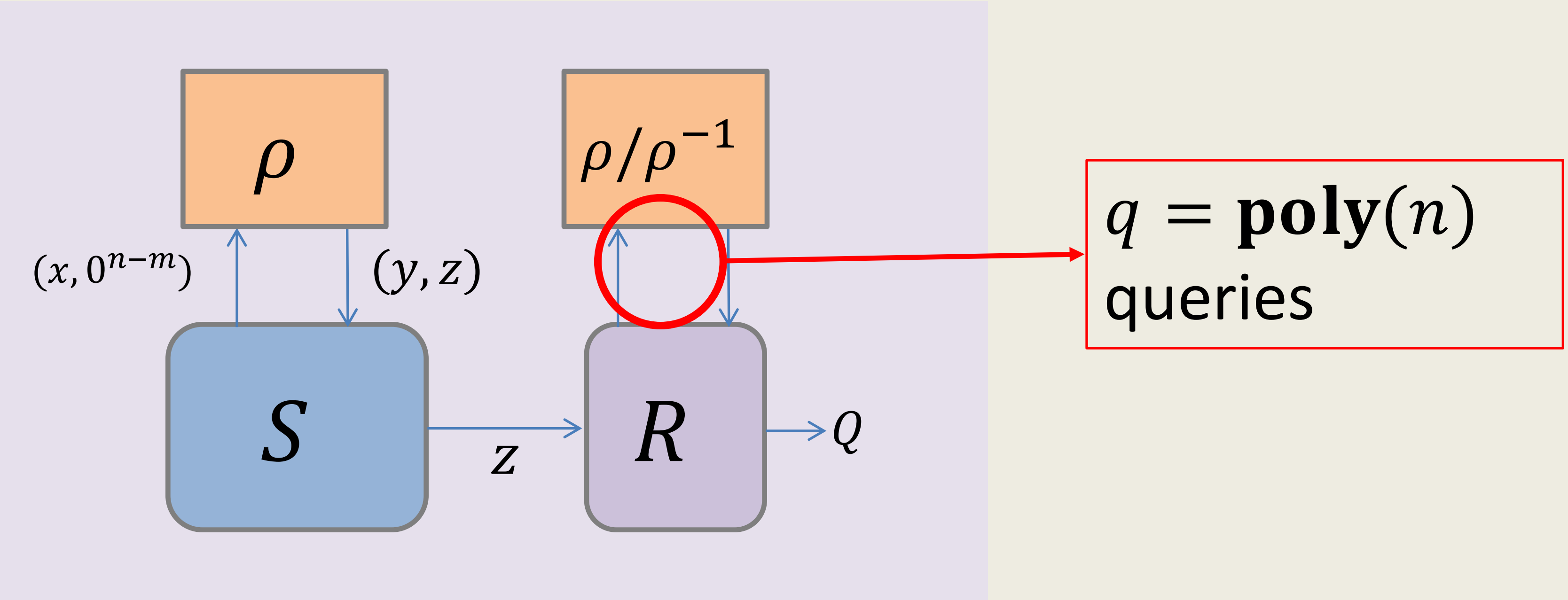


$s \leftarrow \mathrm{Gen}(1^\lambda)$
$x \leftarrow \{0,1\}^{n-m}$
$(y, z) \leftarrow \pi_s(x, 0)$
$b \leftarrow D(s, z)$

if $S \in \mathcal{S}^{sup}$

$s \leftarrow \mathrm{Gen}$
$z \leftarrow \{0,1\}^k$
$b \leftarrow D(s, z)$

Proof – Unpredictability of S



Fact. $\Pr[\{(x, 0^{n-m}), (y, z)\} \cap \ Q \neq \phi] \leq \frac{q}{2^m} + \frac{q}{2^{n-k}}$

Next

Can we get psPRPs at all?

Are psPRPs useful?

Constructions from UCEs

Constructions of UCEs

Heuristic Instantiations

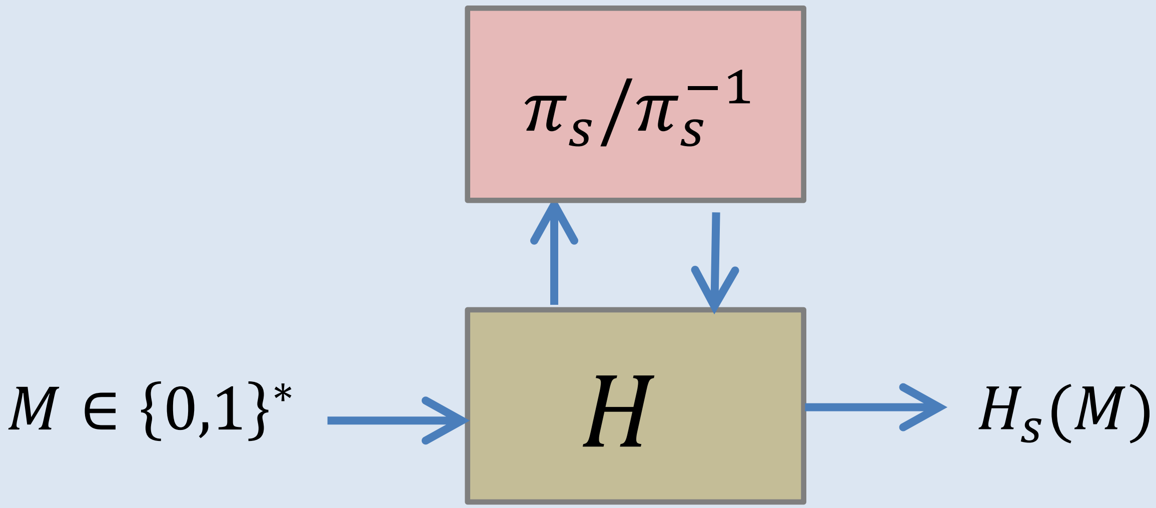
Direct applications

Garbling from fixed-key block ciphers

Common denominator: CP-sequential indifferentiability

How to build UCEs from psPRPs?

$H[P]$

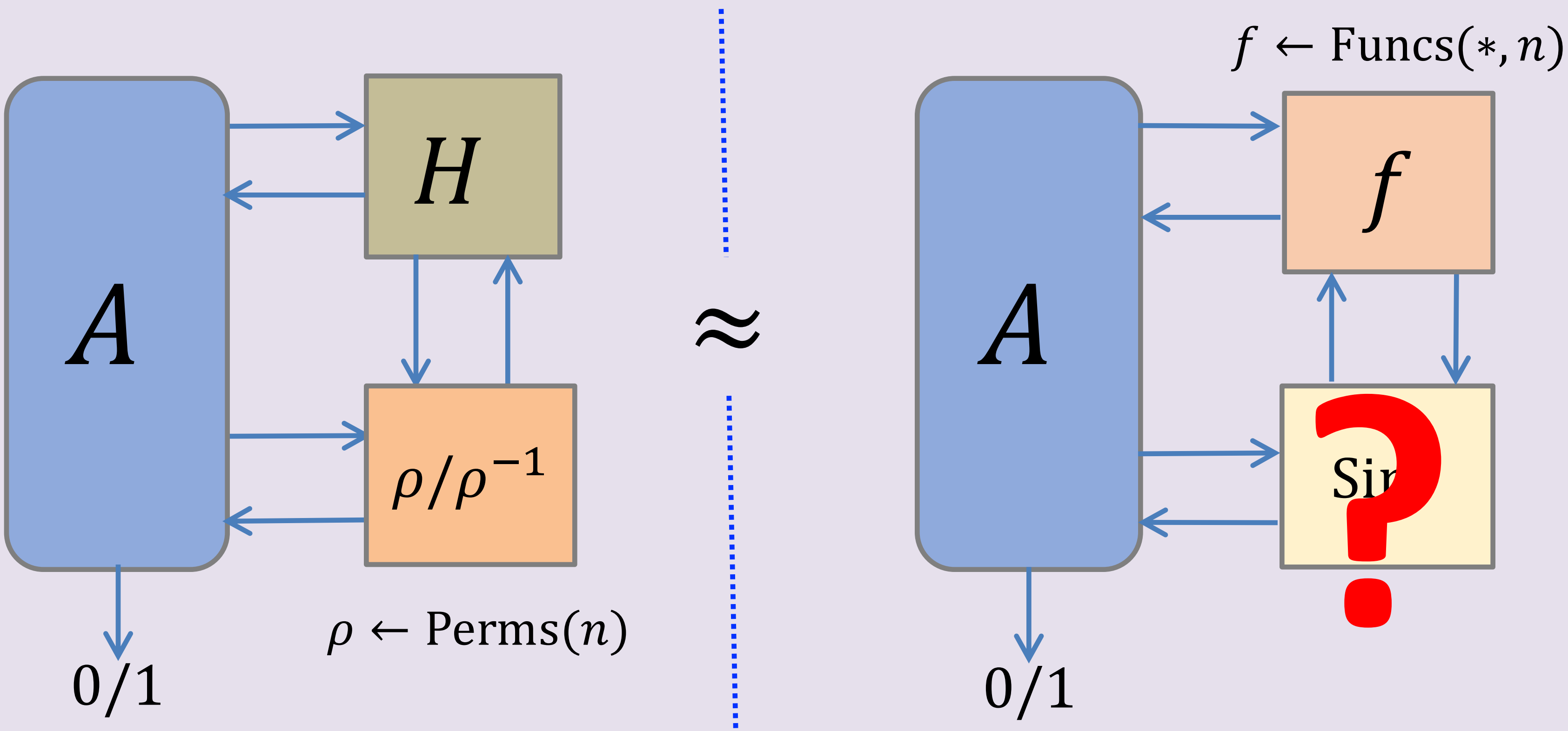


Ideal theorem.

P $\mathbf{psPRP}[\mathcal{S}^{srs}]$-secure $\Longrightarrow$ $H[P]$ $\mathbf{UCE}[\mathcal{S}^{srs}]$-secure.

What does H need to satisfy for this to be true?

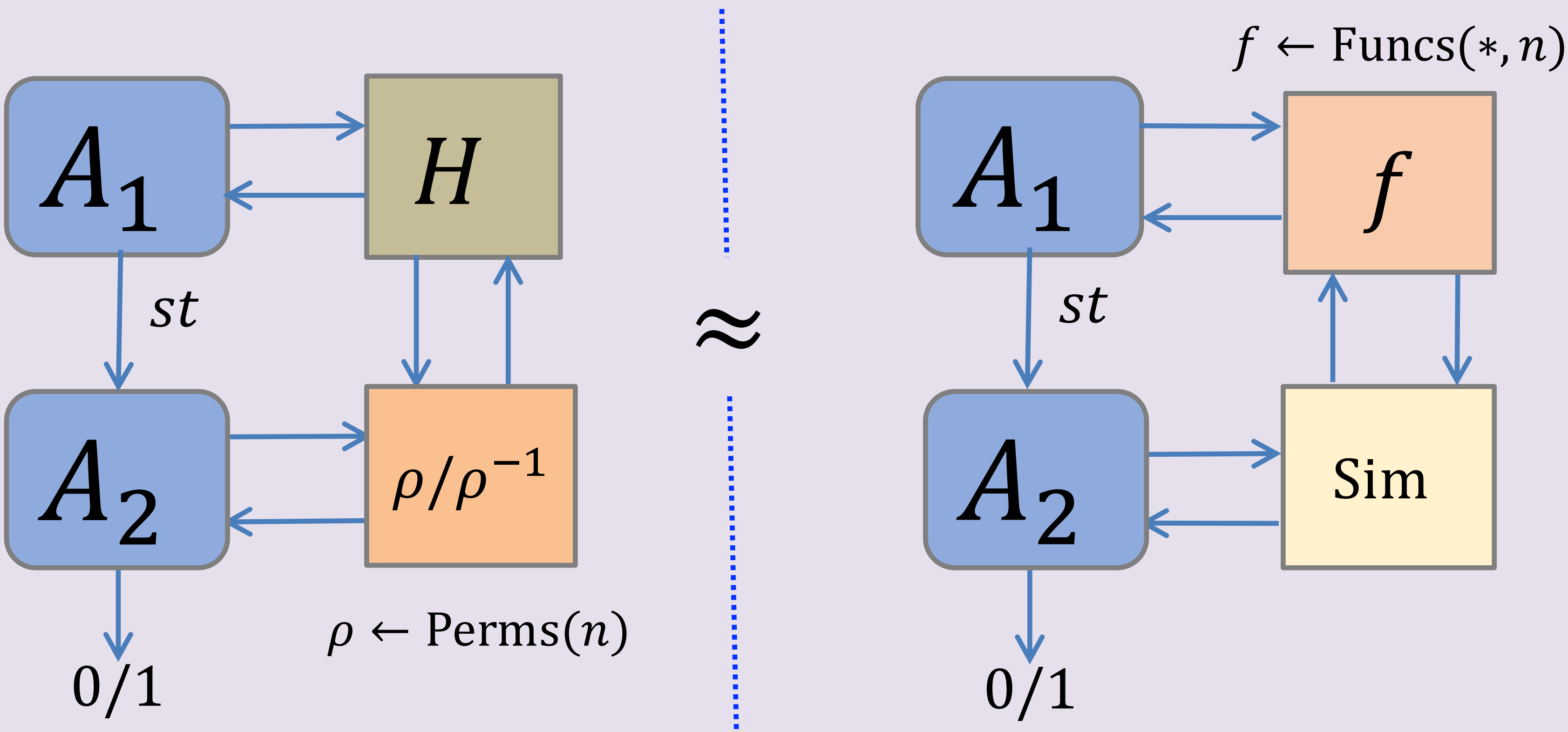
Indifferentiability [MRH04]



Definition. H **indiff. from RO** if $\exists$ PPT Sim $\forall$ PPT A:

$$H + \rho/\rho^{-1} \approx f + \text{Sim}$$

CP-sequential indifferentiability



Def. H **CP-indiff. from RO** if $\exists$ PPT Sim $\forall$ PPT (A_1, A_2):

$$H + \rho/\rho^{-1} \approx f + \mathrm{Sim}$$

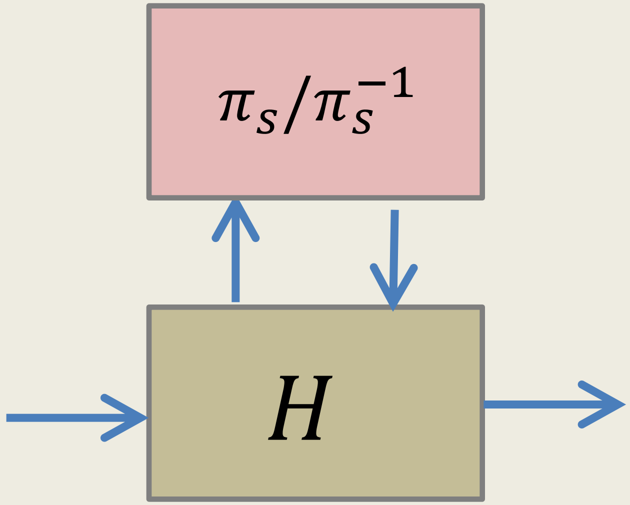
From psPRPs to UCEs

Theorem.

P $\mathbf{psPRP}[\mathcal{S}^{srs}]$-secure
H CP-indiff from RO
$\Longrightarrow$ $H[P]$ $\mathbf{UCE}[\mathcal{S}^{srs}]$-secure.

Similar to [BHK14]. But:

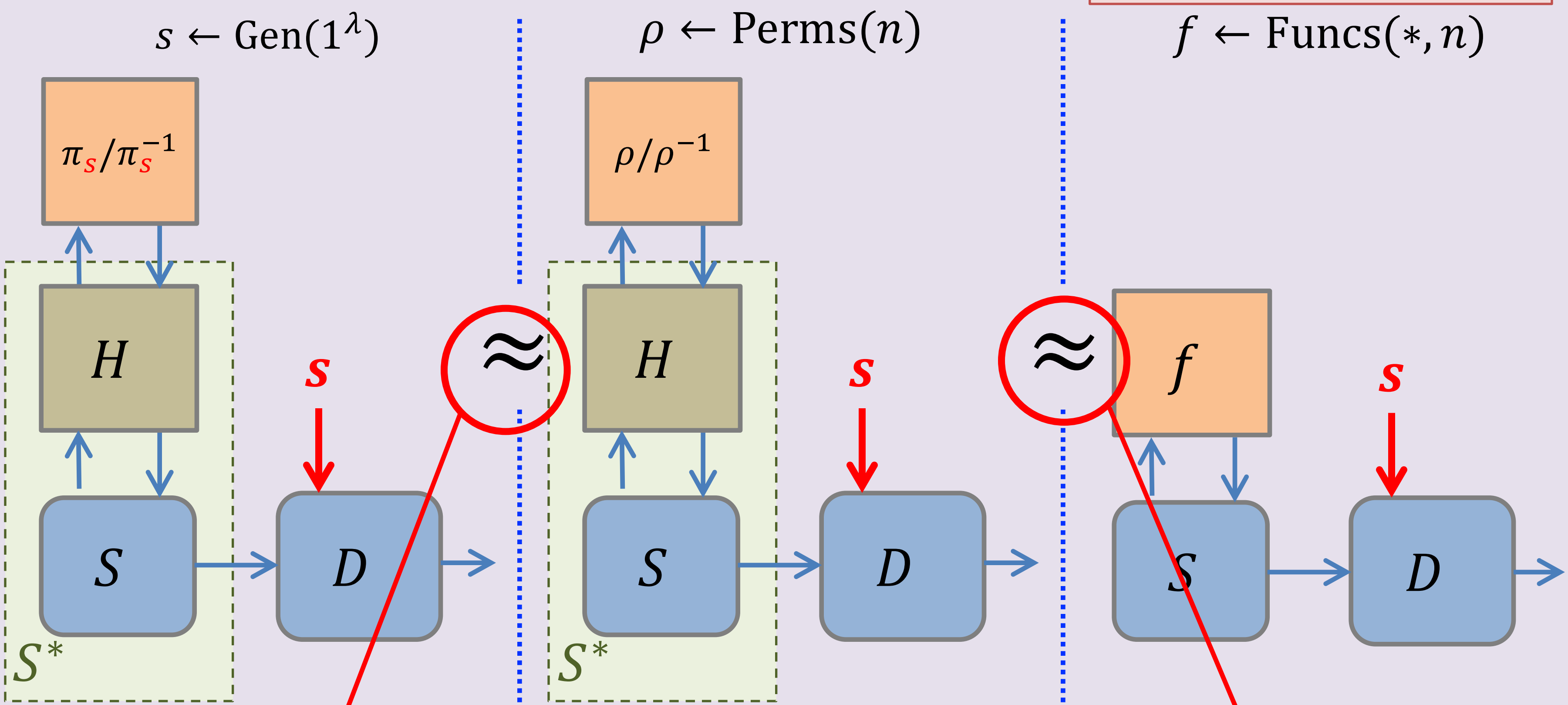
- Needs **full indifferentiability**
- **UCE domain extension**



Corollary. Every perm-based indiff. hash-function transforms a psPRP into a UCE!

From psPRPs to UCEs – Proof

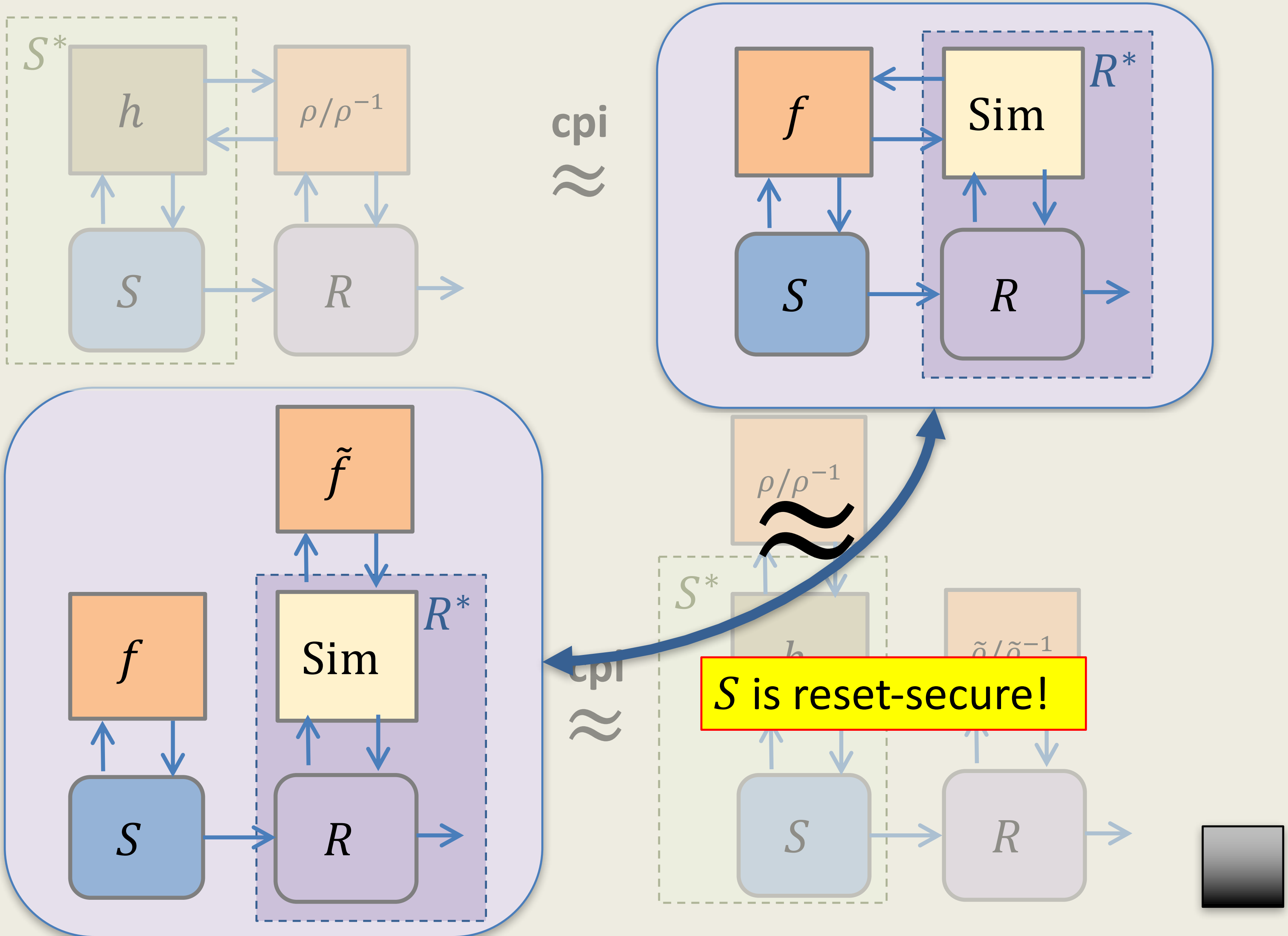
S reset-secure
H is CP-indiff from RO



by $\mathbf{psPRP}[\mathcal{S}^{srs}]$-security if $S^* \in \mathcal{S}^{srs}$

by CP-indiff.

Reset-security of S^*?



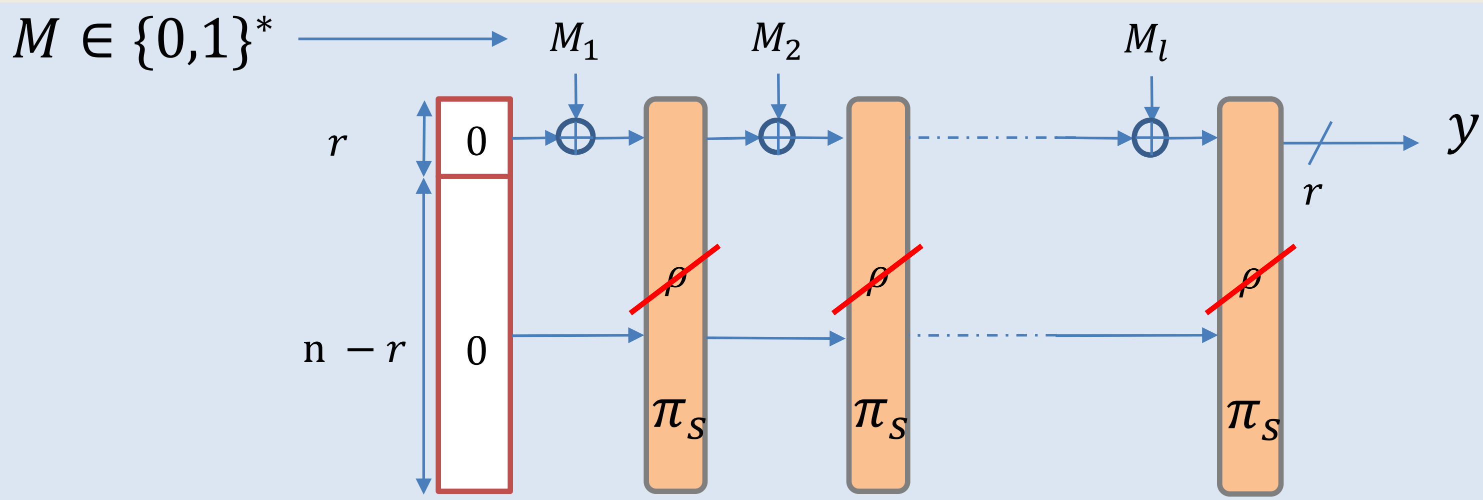
Good news #1

Corollary. Every perm-based indiff. hash-function transforms a psPRP into a UCE!

Many practical hash designs from permutations are indifferentiable from RO!

UCE is a meaningful security target – several applications!

Examples – Sponges



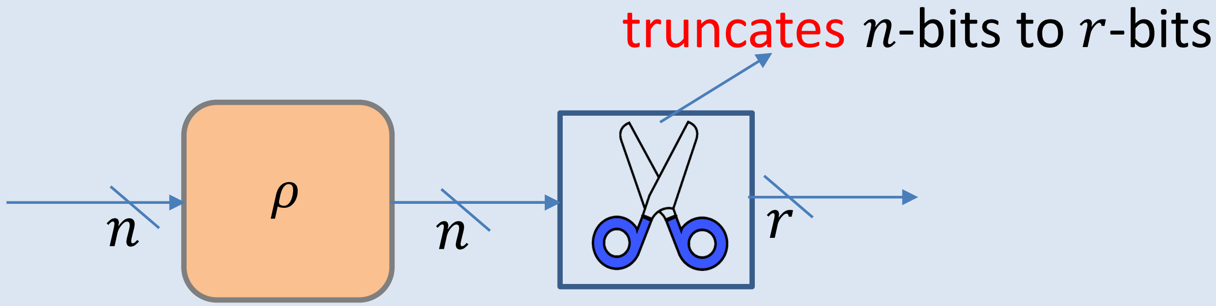
Theorem. [BDVP08] Sponge indifferentiable from RO.

Corollary, P $\mathbf{psPRP}[\mathcal{S}^{srs}]$-secure $\Longrightarrow$ $\mathrm{Sponge}[P]$ $\mathbf{UCE}[\mathcal{S}^{srs}]$-secure.

Validates the Sponge paradigm for UCE applications!

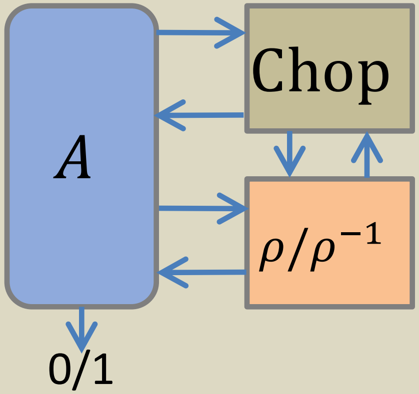
Good news #2 – No need for full indifferentiability

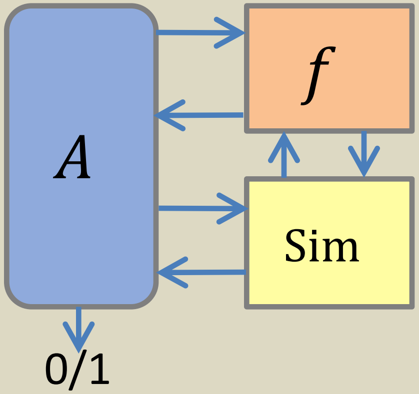
Chop



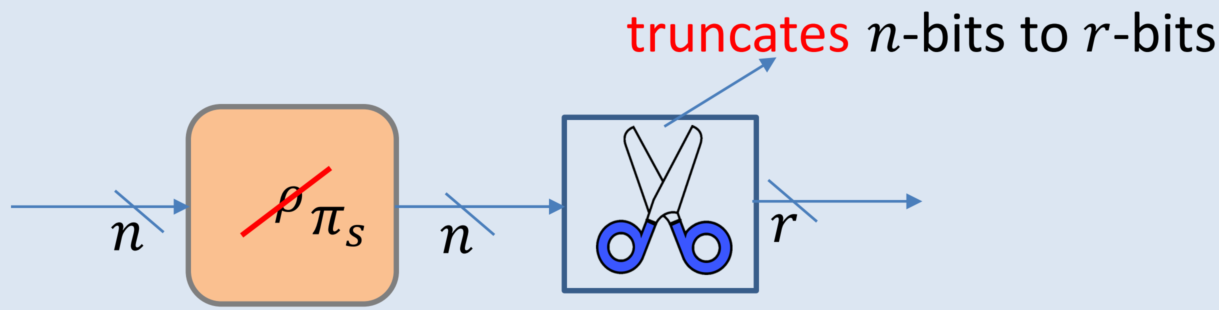
Not indifferentiable!

- For random y, get $x = \rho^{-1}(y)$
- Query construction on x, check consistency with first r bits of y





Chop – Cont'd

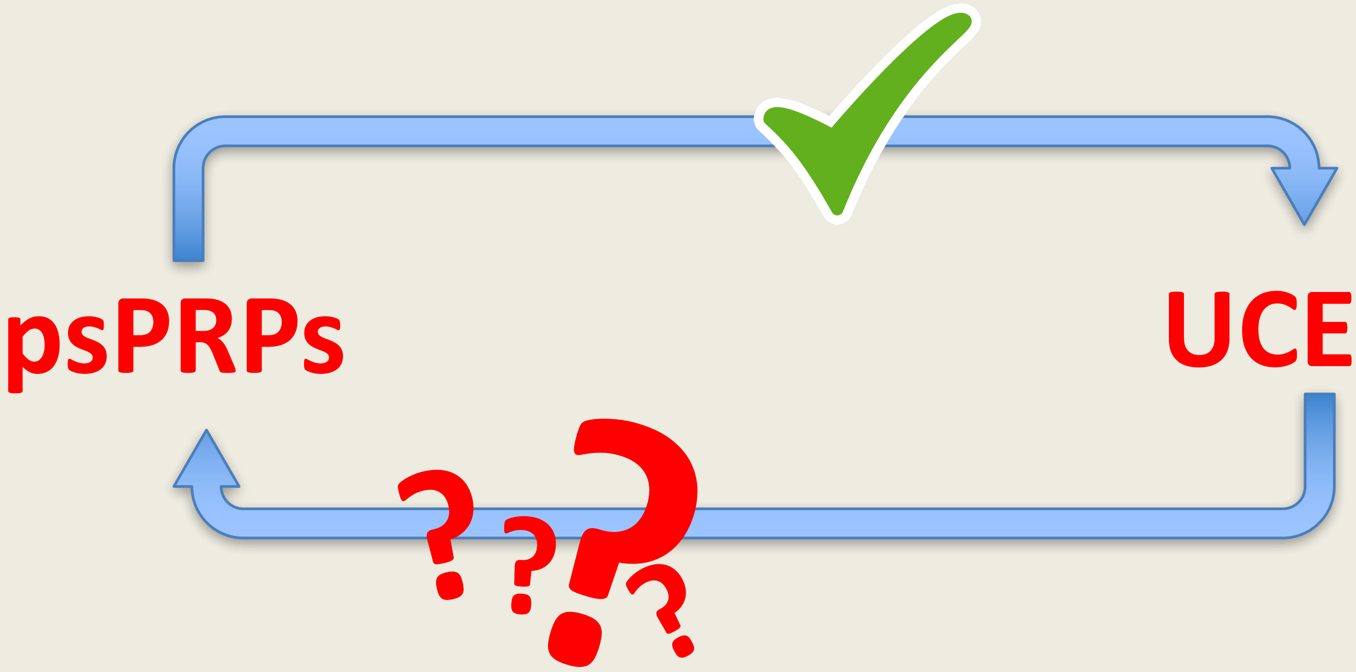


Theorem. Chop is CP-indiff from RO when $n - r \in \omega(\log \lambda)$.

Corollary. P $\cancel{\mathbf{psPRP}[\mathcal{S}^{srs}]}$ $\mathbf{psPRP}[\mathcal{S}^{sup}]$-secure $\Longrightarrow$ $\mathrm{Chop}[P]$ $\cancel{\mathbf{UCE}[\mathcal{S}^{srs}]}$ $\mathbf{UCE}[\mathcal{S}^{sup}]$-secure.

From $\mathrm{Chop}[P]$ to VIL UCE: Domain extension techniques [BHK14]

What about the converse?



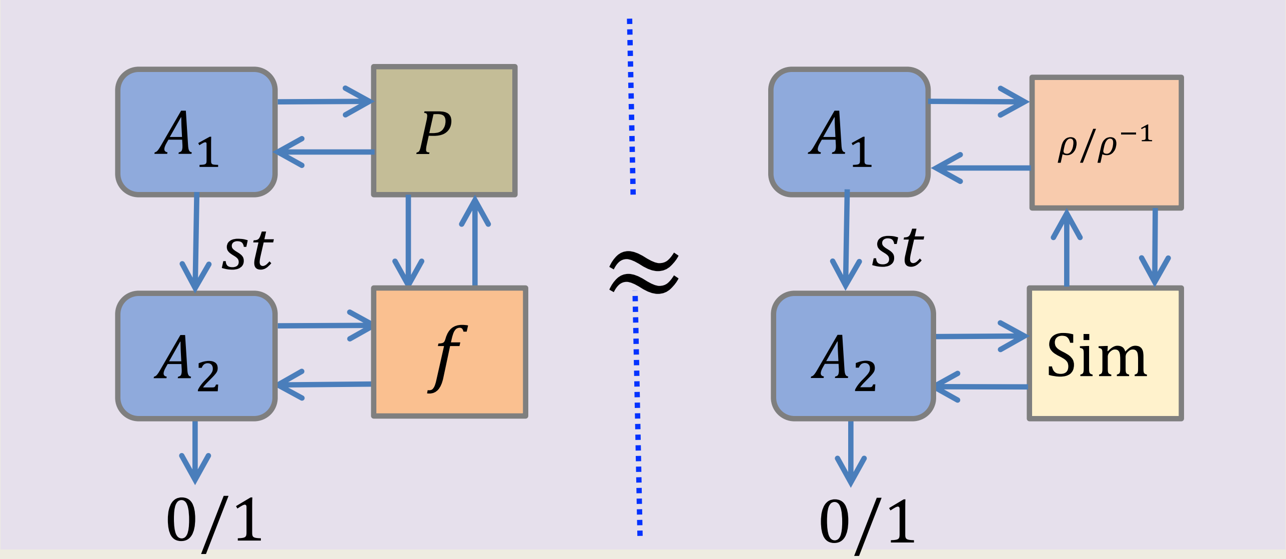
psPRPs from UCEs

Theorem.

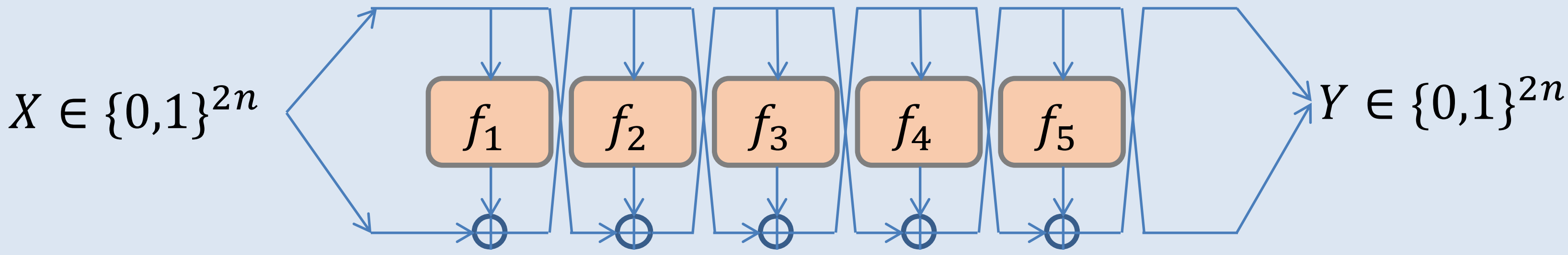
H $\mathbf{UCE}[\mathcal{S}^{srs}]$-secure

P CP-indiff from RP

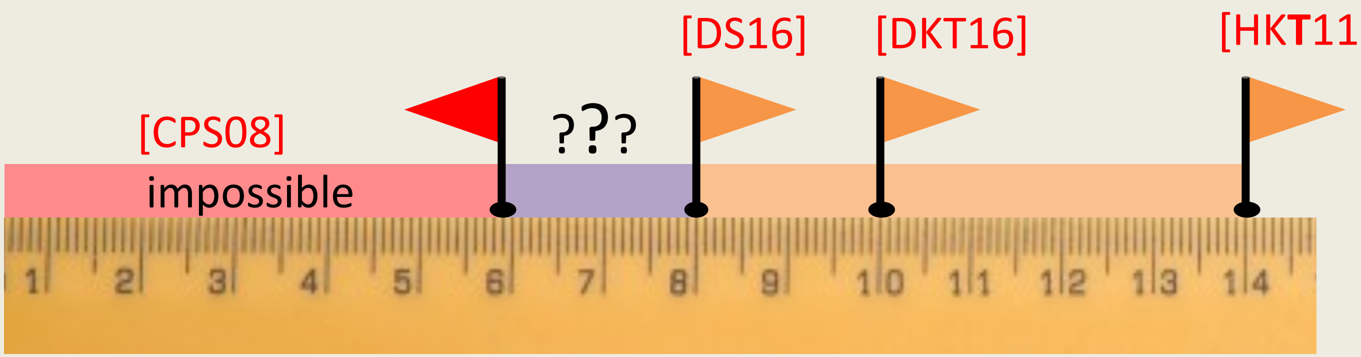
$\Longrightarrow$ $P[H]$ $\mathbf{psPRP}[\mathcal{S}^{srs}]$-secure.



From UCEs to psPRPs – Feistel



$\psi_5[\boldsymbol{f}]$

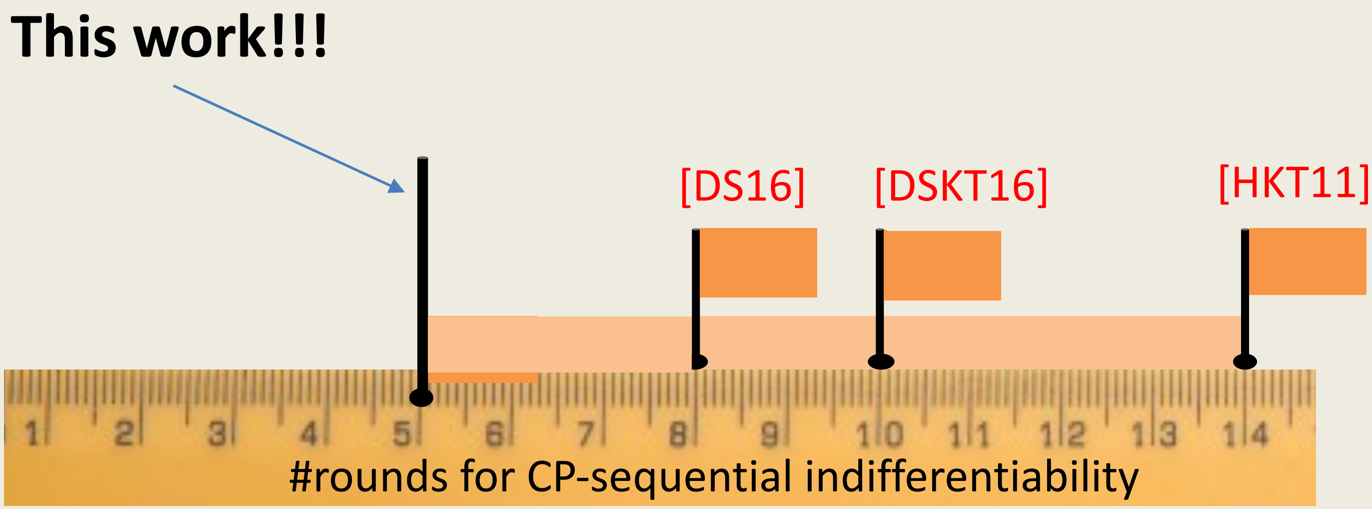


#rounds for indifferentiability

Corollary. psPRPs **exist** iff UCEs exist!!!*

* wrt reset-secure sources

Round-complexity of Feistel for UCE-to-psPRP transformation?



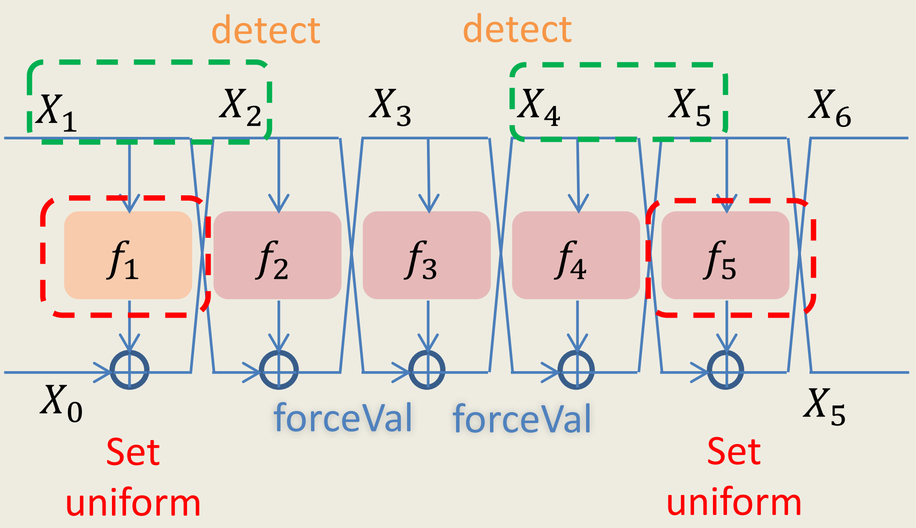
Theorem. 5-round Feistel is CP-indiff from RP

Corollary. $\boldsymbol{H}$ $\mathbf{UCE}[\mathcal{S}^{srs}]$-secure $\Longrightarrow \psi_5[\boldsymbol{H}]$ $\mathbf{psPRP}[\mathcal{S}^{srs}]$-secure.

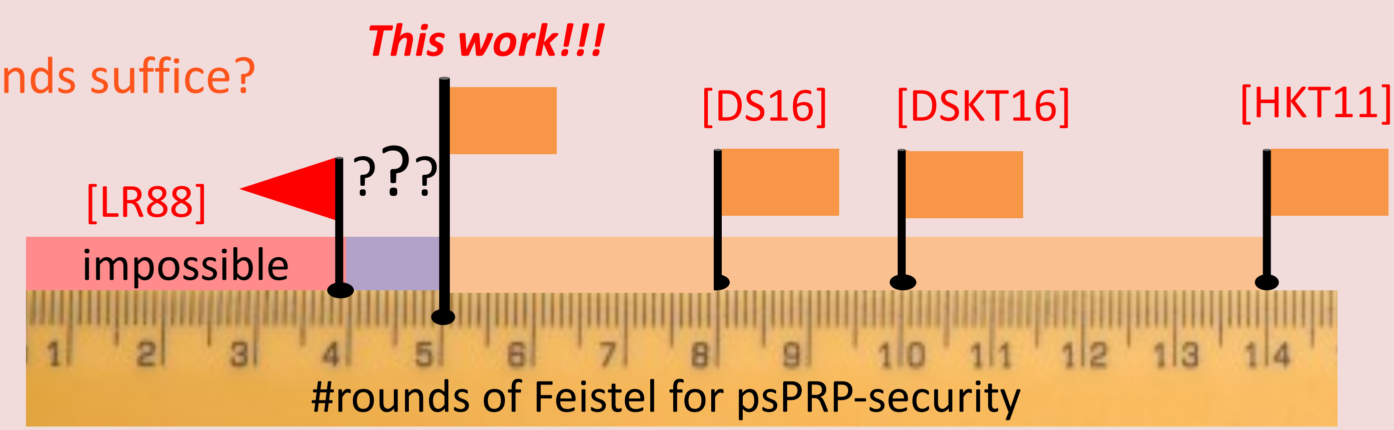
5-round proof is quite involved!

Our 5-round Sim:

- Relies on **chain completion** techniques
- Heavily exploits **query ordering**
- Very different chain-completion strategy from previous works, **no recursion** needed



Open: Do 4-rounds suffice?



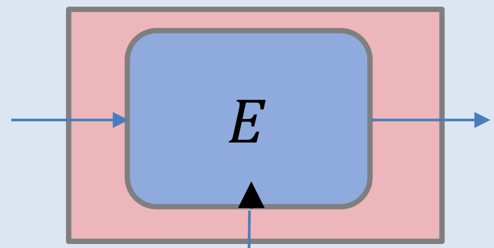
A couple of extra results!

(In passing!)

Heuristic Instantiations

From block ciphers:

$$\pi_s(x) = E(s,x)$$



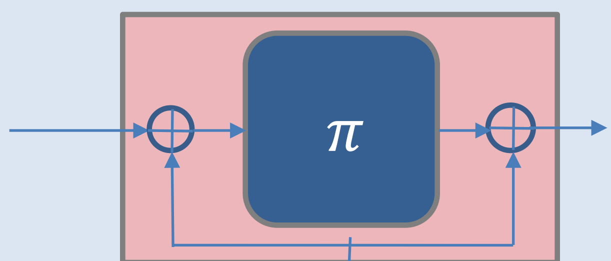
Gen: $s \leftarrow \{0,1\}^k$

psPRP[$\mathcal{S}^{srs}$]-secure

Ideal-cipher model

From <u>seedless</u> permutations:

$$\pi_s(x) = s \oplus \pi(s \oplus x)$$

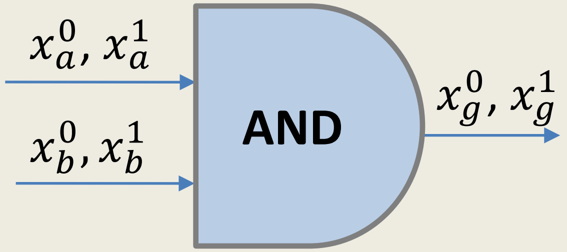


Gen: $s \leftarrow \{0,1\}^k$

psPRP[$\mathcal{S}^{sup}$]-secure

RP model

Fast Garbling from psPRPs



Garbling scheme from [BHKR13]

- Only calls fixed-key block cipher
  $x \to E(0^k, x)$
- **Very fast** – no key re-schedule
- **Proof in RP model**

Garbled **AND**-Gate
$E(0^n, x_a^0 \oplus x_b^0) \oplus x_a^0 \oplus x_b^0 \oplus x_g^0$
$E(0^n, x_a^0 \oplus x_b^1) \oplus x_a^0 \oplus x_b^1 \oplus x_g^0$
$E(0^n, x_a^1 \oplus x_b^0) \oplus x_a^1 \oplus x_b^0 \oplus x_g^0$
$E(0^n, x_a^1 \oplus x_b^1) \oplus x_a^1 \oplus x_b^1 \oplus x_g^1$

Our variant: $E(0^k, x) \Rightarrow \pi_s(x)$, fresh seed s generated upon each garbling operation!

Theorem. Secure when π_s is $\mathbf{psPRP}[\mathcal{S}^{sup}]$.

Roadmap

1. Definitions

2. Constructions & Applications

3. <u>Conclusions</u>

Conclusion

	ideal model	standard model
Hash functions	random oracle	CRHF, OWFs, UOWHFs, CI, UCEs...
Permutations	RP	psPRPs

First (useful) standard model assumptions on permutations

Constructions

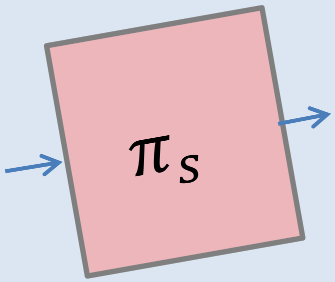
psPRPs

Applications

(Some) open questions

More on psPRPs:

- More efficient constructions from UCEs?
- Weaker assumptions?
- Cryptanalysis?



ps-Pseudorandomness as a paradigm:

- **UCE** = **psPRF**
- Applications of psX?

Beyond psPRPs:

- Simpler assumptions on permutations?

Is SHA-3 a CRHF under any non-trivial assumption?

Thank you!

Paper on ePrint really soon …

For now: http://www.cs.ucsb.edu/~tessaro/papers/SonTes17.pdf