Lower-Bounds on Public-Key Operations in PIR

Mohammad Hajiabadi (U Waterloo)

(Based on join work with Jesko Dujmovic, CISPA)

Disclaimer

- I will use the word "public-key operations" loosely in the first half of my talk, but I will define it later. Be patient!

Private Information Retrieval (PIR) [CGKS95,KO97]

1. Bob shouldn't learn anything about index i.
2. Bob-to-Alice Communication < n

NOT Required:
- Bob Privacy: Alice shouldn't learn more than m_i
- Total communication less than n.

Database of n bits

m_1

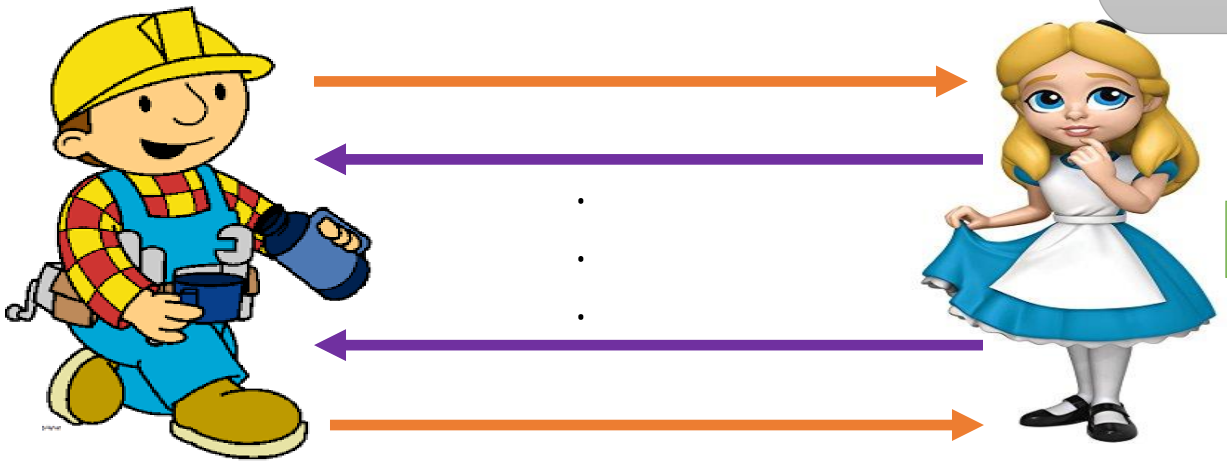
m_2

.

.

.

m_n



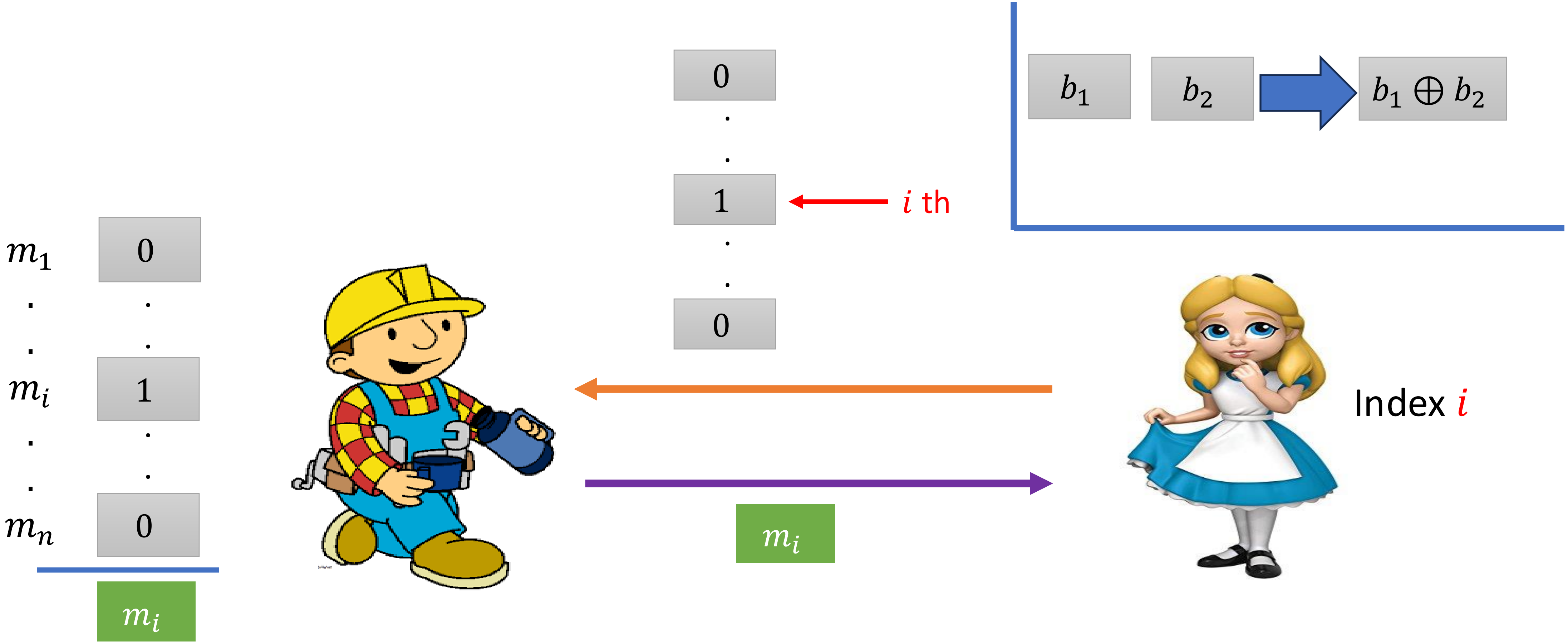
m_i

Index i

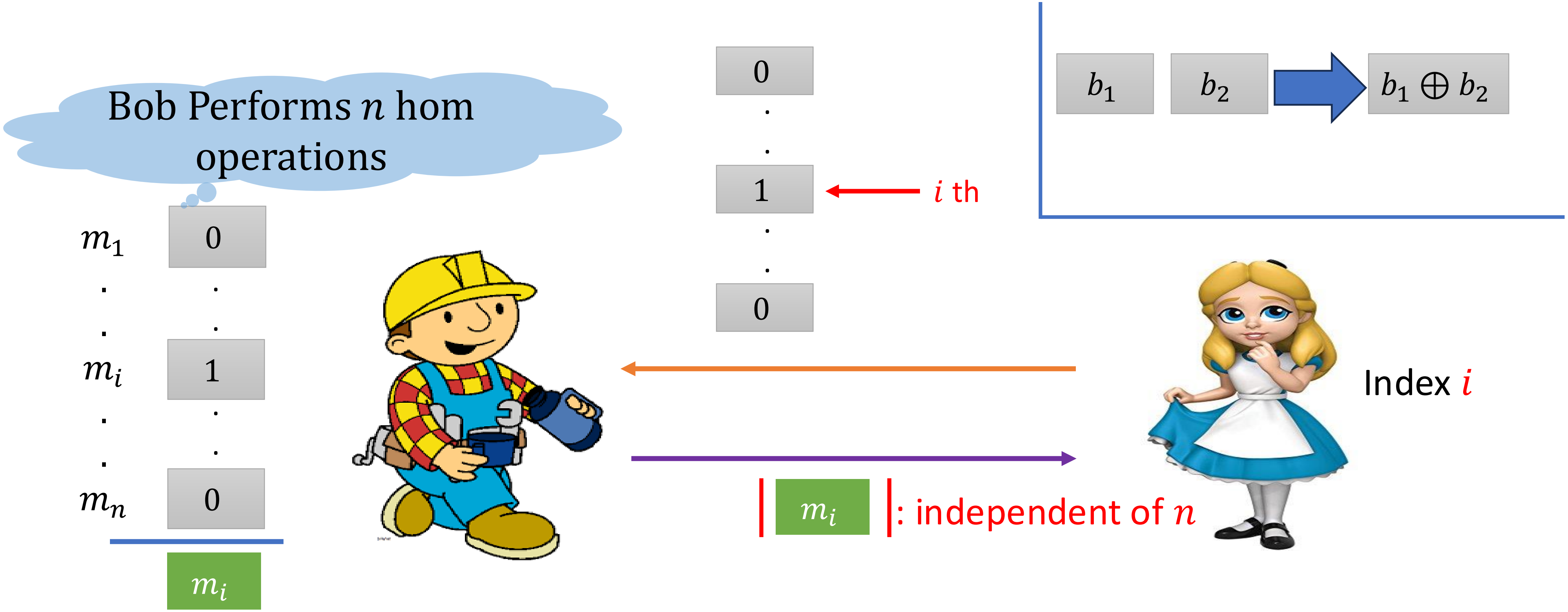
No Preprocessing!

Non-Trivial PIR: Satisfy (1) and (2) and perfect correctness

PIR from Additively Homomorphic Encryption



PIR from Homomorphic Encryption



PIR from Homomorphic Encryption

Inherent?

Bob Performs n hom operations

m_1	0
.	.
.	.
m_i	1
.	.
.	.
m_n	0
	m_i

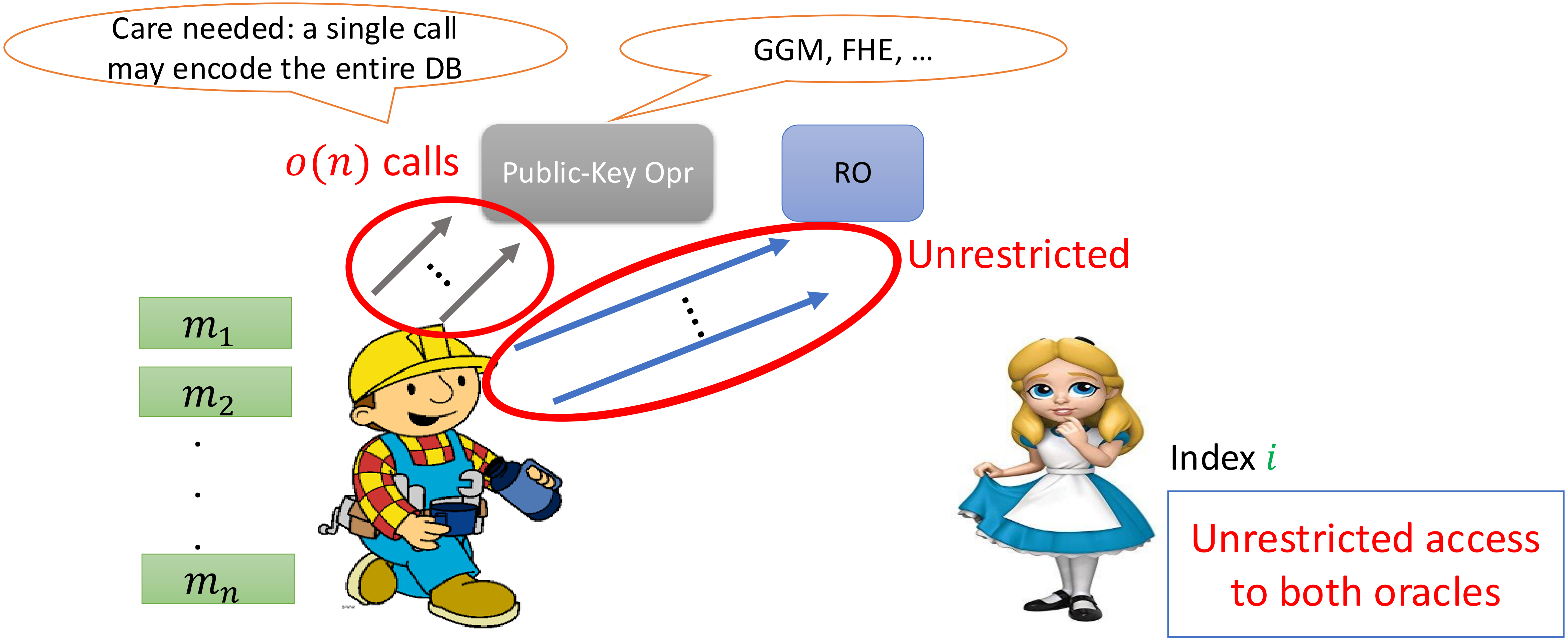
0
.
.
1 ← i th
.
.
0

b_1 b_2 → $b_1 \oplus b_2$

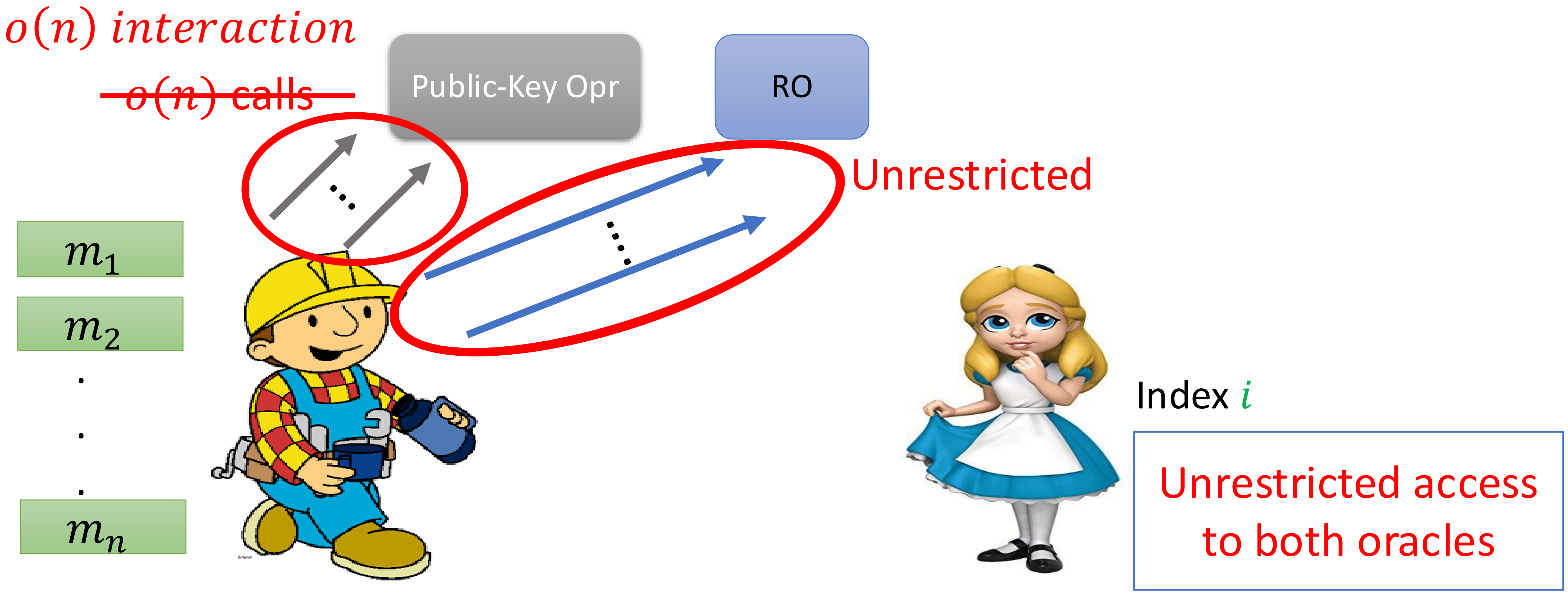
$|m_i|$: independent of n

Index i

Problem Formulation (Rough)



Motivating Question



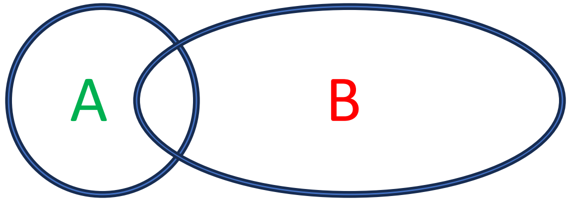
PIR: Computational Efficiency

- Server's running time must be at least n [CGKS95,BIM00]
  - If server's running time < n, the sever isn't probing at least one entry, leaking info about index i.

- PIR implies oblivious transfer and hence requires public-key assumptions [CMO00,IR88].
- Curious Fact: All Non-Trivial PIRs based on Generic Groups/Hom Enc employ at least a linear number of public-key operations, irrespective of # symmetric-key operations. [IP07,DGIMMO19,GHO20,CGHLM21,…]
- Is n public-key operations inherent?
  - Can we have a PIR protocol where the number of server's public-key operations $\ll n$ but the number of symmetric-key operations is arbitrarily large?

- Let’s Look at another scenario where we are left with a linear number of public-key operations.

Asymmetric Private-Set Intersection (PSI)



- Computing set intersection when $|A| \ll |B|$.
- Goal: semi-honest privacy+ Communication complexity not growing with $|B|$.
- Solutions based on trapdoor hash, rate-1 OT, etc [IshPas07, Dottling et al 19, GarHajOst 20, Alamati et al 21, Chase et al 21,Brakerski et al 22, …]
- But # public-key operations (e.g., group operations) grows at least linearly with $|B|$.
- We can have PSI protocols Π where $Comm(\Pi)$ grows with $|B|$ but the number of public key operations don't (e.g., based on OT Extension [PinSchTkaYan19,ChaseMiao21,…]).
- Is one or the other inherent?

Commonality between PIR and Asymmetric PSI

Both a special case of asymmetric MPC: computing $f(x, y)$, where $|x| \ll |y|$, Alice holding x and Bob holding y.

1. We only require semi-honest security for Alice.
2. Bob-to-Alice Communication: $< |y|$.

Our lower-bounds on the number of public-key operations will also apply to this general setting.

Possible Approach for Minimizing Public-Key Operations?

Why not use OT extension?

OT Extension [Beaver96,IKNP03]

Symmetric-Key Operations

A few OTs → Large number of OTs

OT Extension [Beaver96,IKNP03]

RO

OT

(m_0^1, m_1^1)
(m_0^2, m_1^2)
$\vdots$
(m_0^ℓ, m_1^ℓ)

b_1
b_2
$\vdots$
b_ℓ

Should learn $m_{b_i}^i$ for all i

Number of OT calls: $Poly(\lambda)$ for a fixed $Poly$, independent of ℓ

- [Beaver 96]: OT extension via non-black-box use of PRGs.
- [IKNP03]: OT extension via black-box use of RO.

OT Extension Implications

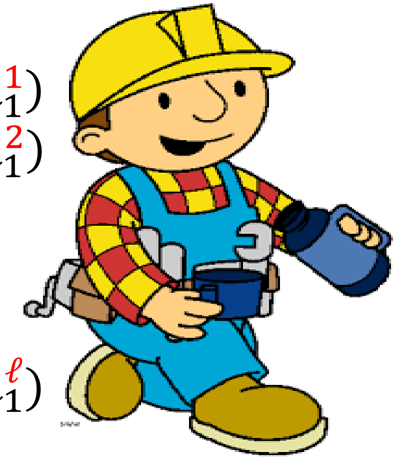
- We can realize MPC for any function f by making a number of public-key operation calls independent of $|f|$.
- Why doesn't OT extension solve computationally-efficient PIR?
  - OT extension isn't communication efficient for chosen-message OTs!
- Let's take a closer look!

Communication of OT Extension [Beaver96,IKNP03]

RO

OT

(m_0^1, m_1^1)
(m_0^2, m_1^2)
.
.
.
(m_0^ℓ, m_1^ℓ)



b_1
b_2
.
.
.
b_ℓ

Should learn $m_{b_i}^i$ for all i

Number of OT calls: $Poly(\lambda)$ for a fixed $Poly$, independent of ℓ

Sender (Bob) communication of [IKNP03] at least $2\ \ell$ bits.

Needed for PIR: sender rate of 1, defined as $\frac{\ell}{|Bob\ communication|}$

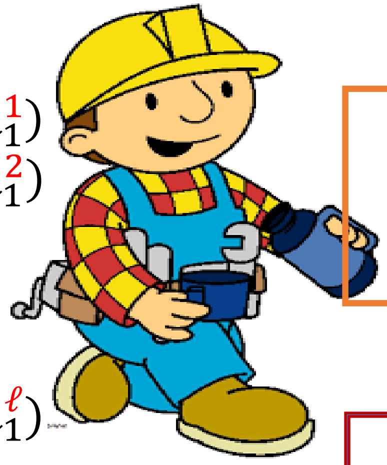
What Blows up OT-Extension Comm?

RO

OT

(r_0^1, r_1^1)
(r_0^2, r_1^2)
$\vdots$
(r_0^ℓ, r_1^ℓ)

(m_0^1, m_1^1)
(m_0^2, m_1^2)
$\vdots$
(m_0^ℓ, m_1^ℓ)



OT Correlations

$\{(m_0^i \oplus r_0^i, m_1^i \oplus r_1^i)\}$

Communication Blow-Up

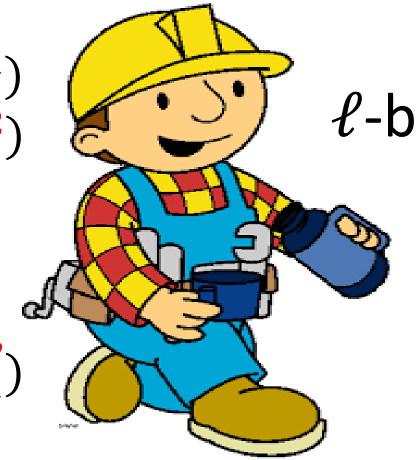
b_1 $r_{b_1}^1$
b_2 $r_{b_2}^2$
$\vdots$
b_ℓ $r_{b_\ell}^\ell$

Batch OT: Communication vs Computation

Dream Version:

1. #Public-Key Op: independent of ℓ
2. Sender (Bob) communication: $\ell + \lambda$
   (i.e., rate 1 for Bob $\frac{\ell}{\ell+\lambda}$)

(m_0^1, m_1^1)
(m_0^2, m_1^2)
$\vdots$
(m_0^ℓ, m_1^ℓ)



ℓ-batch OT

b_1
b_2
$\vdots$
b_ℓ

Communication-efficient protocols

- Achieving sender rate-1 but with large # public-key operations [IP07, DGIMMO19, GHO20,CGHLM21,BBDP22]

Computation-efficient protocols

- OT extension [IKNP03,KK13]: Small # public-key operations but large communication ($\leq 1/2$ sender rate)

Million $ question: Can we get the best of both?

No! Beating ½ rate is impossible via $O(\lambda)$ public-key operations

Main Result

- The sender of any PIR protocol should make close to linear public-key operations.

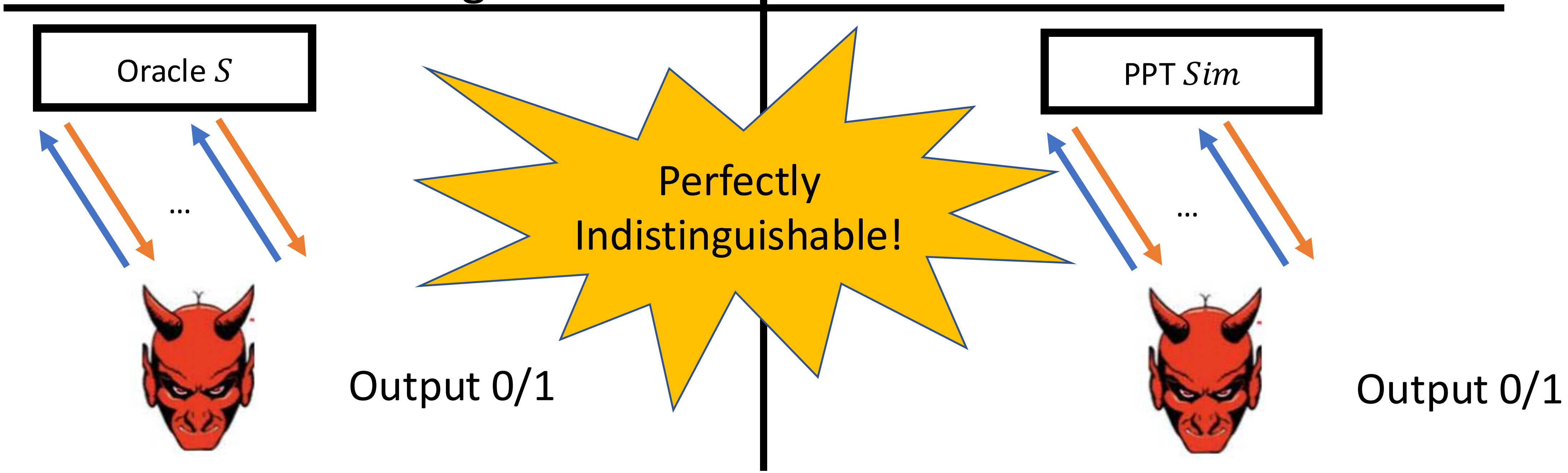
- But what is a public-key operation? Let’s define it.

Public-Key Operation

- A primitive that implies PKE and can be captured via simulatable oracles.

Simulatable Oracles

- Simulatable oracle: An oracle that can be efficiently sampled on the fly --- aka amenable to lazy sampling.
- There exists an efficient lazy sampler Sim, where an adversary cannot tell if he is interacting with a true oracle S or with Sim.



Examples of Simulatable Oracles

- Trivial Examples: a random oracle is simulatable, since it can be sampled on the fly.
  - The lazy sampler will simply sample a random output on a new input.

- With more work you can show GGM, FHE, iO are all simulatable.

Main Result (More Formal)

- Let S be a simulatable oracle for a public-key primitive.
- PIR^S => $\overline{PIR}$ where $\overline{PIR}$ makes no calls to S!
- Receiver privacy remains intact!
- Sender-Comm($\overline{PIR}$) = Sender-Com(PIR^S)+O(#calls to S by PIR sender)

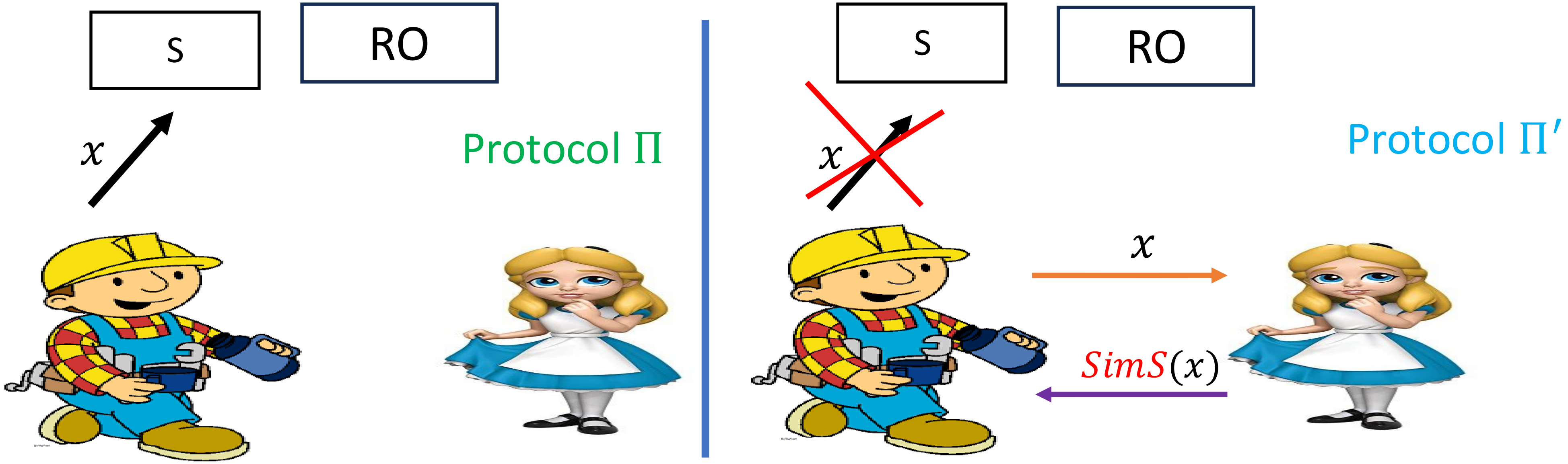
Consequence

- Any $PIR^{S,RO}$ protocol that makes a small # calls to S can be compiled into a non-trivial PIR protocol $\overline{PIR}^{RO}$ that makes no calls to S!
- But we know PIRs cannot be realized relative to ROs [ImpRud88, CreMalOst2000]!

Main Idea: Compilation

- Compilation: let the PIR receiver act as a lazy sampler, and answer queries to the Simulatable oracle S for both herself and the sender!
- More detail: when the sender is to make an S query, he forwards the query to the receiver and the receiver simulates the response!

Compilation (Cont'd)



- Bob-$Comm(\Pi') =$ Bob$-Comm(\Pi) + \#\, S\ oracle\ calls \cdot (query\ size)$
- Impossibility:$c < 1, \mathrm{Comm}(\Pi)$=$n^c$ $\#\, S\ oracle\ calls = \mathrm{n}^{\mathrm{c}}$ $query\ size = \lambda$

Consequences

- We can prove similar computation-communication tradeoffs for any asymmetric 2PC (e.g., asymmetric PSI)
- The communication complexity of IKNP is close to optimal (see the paper for that)

Follow-Up Work: Doubly-Efficient PIR

- Recent result Eurocrypt 2025

Black Box Crypto is Useless for Doubly Efficient PIR

Wei-Kai Lin-Ethan Mook-Daniel Wichs

Generalizes our techniques and shows that doubly-efficient PIR is impossible with respect to any assumption that can be captured as black-box oracles.

Open Problems

- Lower-bound on # Public-Key Operations in other settings?
  - For example, we don't have a hybrid encryption paradigm for functional encryption. Prove that a linear number of public-key operations for certain FE primitives (e.g., Inner-Product FE) is inherent.
- Can we prove query-lower-bounds for multi-server PIR?
  - In general, multi-server PIR can be done information theoretically, but certain kinds of multi-server PIR require cryptographic assumptions .