Perfect Secure Computation in Two Rounds

Benny Applebaum
Tel Aviv University

Joint work with Zvika Brakerski and Rotem Tsabary

Theory and Practice of Multiparty Computation Workshop, Aarhus, 2018

The MPC Zoo

Online/offline MPC for dynamic functionalities

Semi-honest
Computational
2-party OT-channels

UC-secure
Arithmetic MPC
with CRS

Covert MPC

Maliciously-secure MPC
Client-Server Model

Perfectly-secure MPC
with Correlated randomness

Adaptively-secure
MPC with fairness

Today: Simplest Model (BGW,CCD)

- N parties
- point-to-point private channels
- passive adversary
- honest majority
- perfect security
  - unbounded adversary
  - unconditional

“the simplest nervous system is in certain jellyfish”

“Should have been discovered by the ancient Greeks…”

Ronald Cramer (3 days ago)

Today: Simplest Model (BGW,CCD)

- N parties
- point-to-point private channels
- passive adversary
- honest majority
- perfect security
  - unbounded adversary
  - unconditional

"the simplest nervous system is in certain jellyfish"

Simple BUT:

- Useful starting point for more realistic adversaries
- Still quite a few **open problems**

Completeness Results

1988:

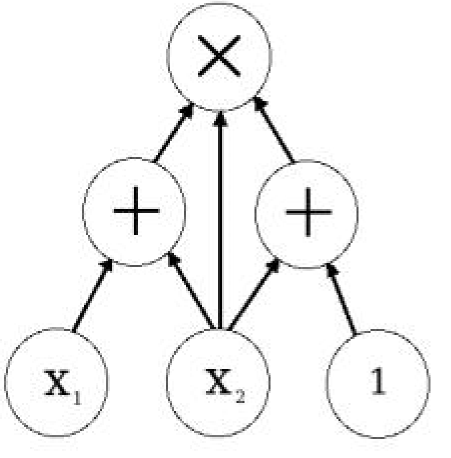
[Ben-Or, Goldwasser, Wigderson, Chaum, Crépeau, Damgård]

Thm. At the presence of honest majority, **every** function f can be perfectly computed

Tight: Honest Majority is necessary

Complexity: poly(circuit-size(f))

Rounds: Multiplicative depth of f



Constant Round Complexity?

[Bar-Ilan-Beaver-1989]: expected O(1) round for all f

- efficient protocol for NC1
- worst-case O(1) round with statistical security

90’s: restricted interaction patterns &
efficiency slightly beyond NC1

- [FKN ‘94,IK’ 97, CD ‘00]

Constant Round Complexity?

2000-02: [Ishai-Kushilevitz]

Thm. With honest majority,

3-round **perfect** protocol for all functions

- Randomizing Polynomials
  - Every function reduces to degree 3 computation

Constant Round Complexity?

2000-02: [Ishai-Kushilevitz]

Thm. With honest majority,

3-round **perfect** protocol for all functions

- Efficient for NC1 and log-space
- Computational variant for poly-size circuits [AIK05]

Constant Round Complexity?

2000-02: [Ishai-Kushilevitz]

Thm. With honest majority,

3-round **perfect** protocol for all functions

- 1 round is impossible
- Yields 2 rounds if privacy threshold <n/3

Open: With honest majority,

2-round perfect protocol for all functions?

- [IK00] cannot be achieved with degree-2 randomizing polynomials

Ishai-Kushilevitz 2000:

"An open question of a somewhat different flavor is that of finding the exact number of rounds required for privately evaluating an arbitrary (i.e., a worst-case) function f with an optimal privacy threshold.

Using randomizing polynomials, an upper bound of 3 was obtained. If this bound is tight (i.e., 2 rounds are not enough) then, in a very crude sense, the randomizing polynomials approach is non-restrictive."

Our Results

Thm 1: With honest majority,
2-round perfect protocol for all functions

- Efficient for NC1 and log-space
- New paradigm: Multiparty Randomized Encoding
  - Relaxes Randomized Encoding
  - Abstracts Garbled Protocols [Garg-Srinivasan-2017]

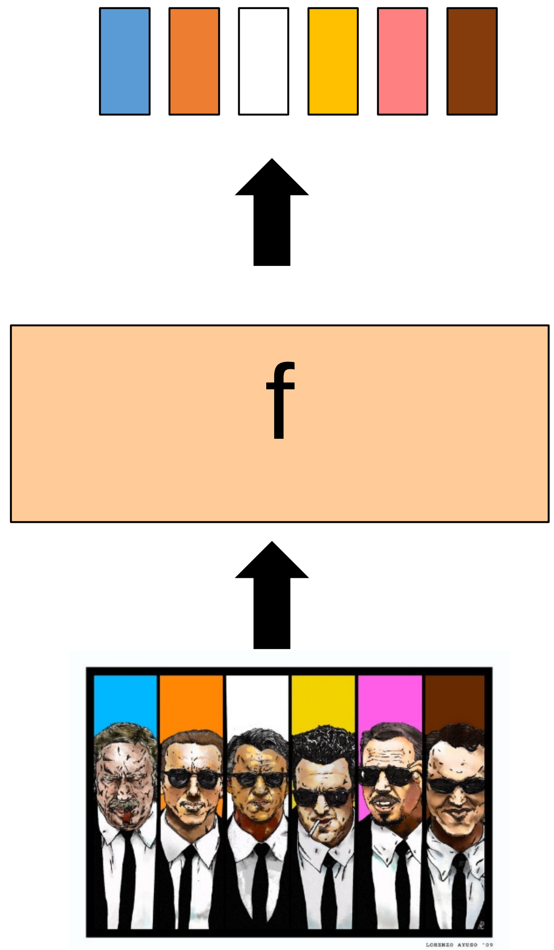
Thm 2: Assume OWF and honest majority.
eff. **2-round** comp. protocol for poly-size circuits

- Parties make only BB calls to OWF.
- Incomparable to [Garg-Srinivasan'18], [Benhamouda-Lin'18]
- We don't need OT but require honest majority

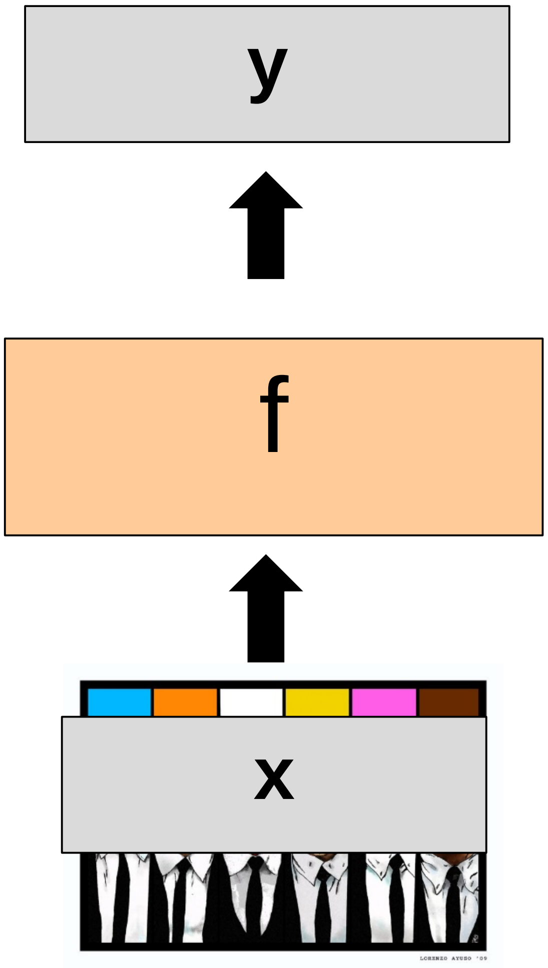
The rest of the talk

- Randomizing Polynomials
- Multiparty Randomized Encoding (MPRE)
- About the proof: MPRE with degree* 2
- Conclusion

Randomizing Polynomials [IK00]

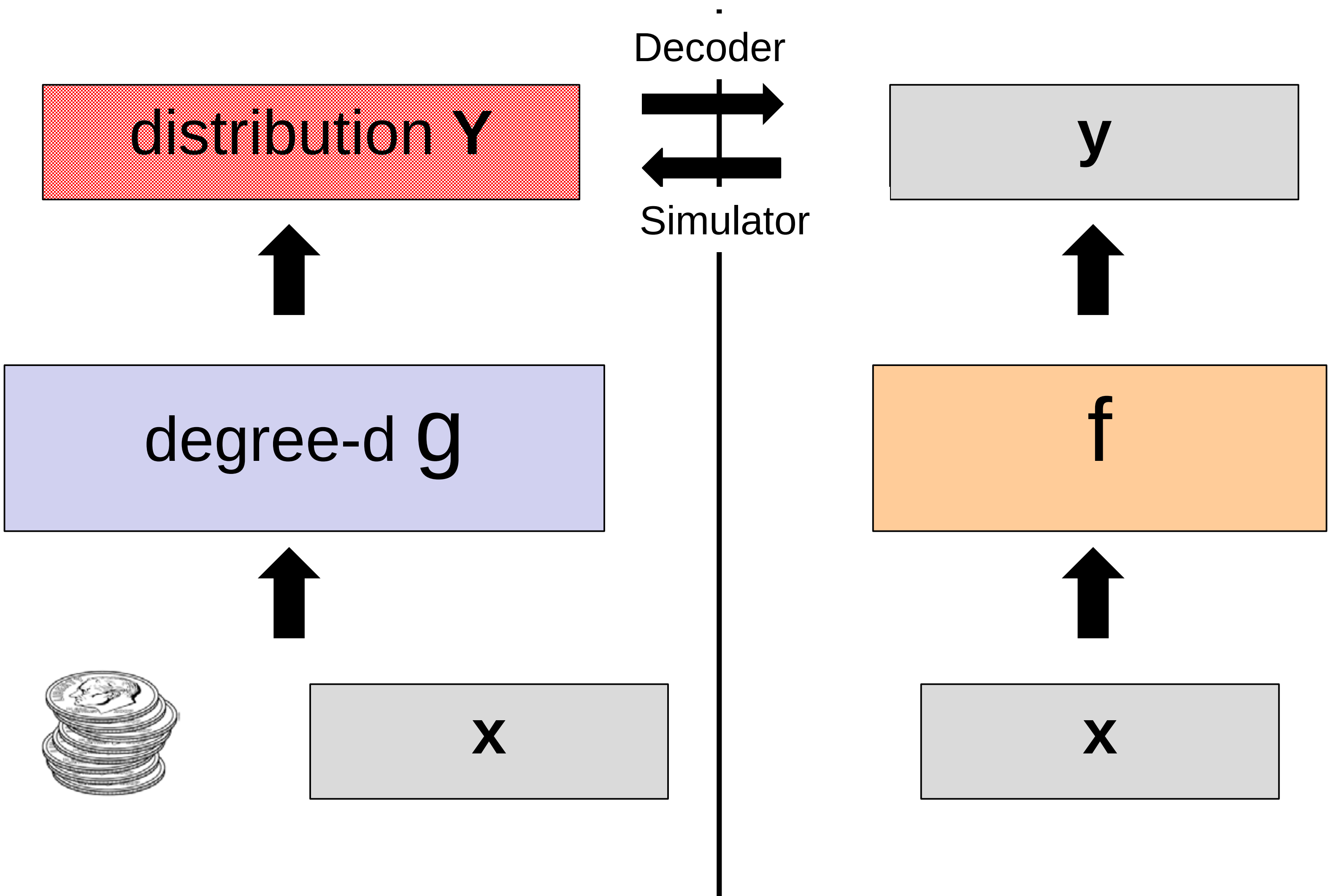


Randomizing Polynomials [IK00]



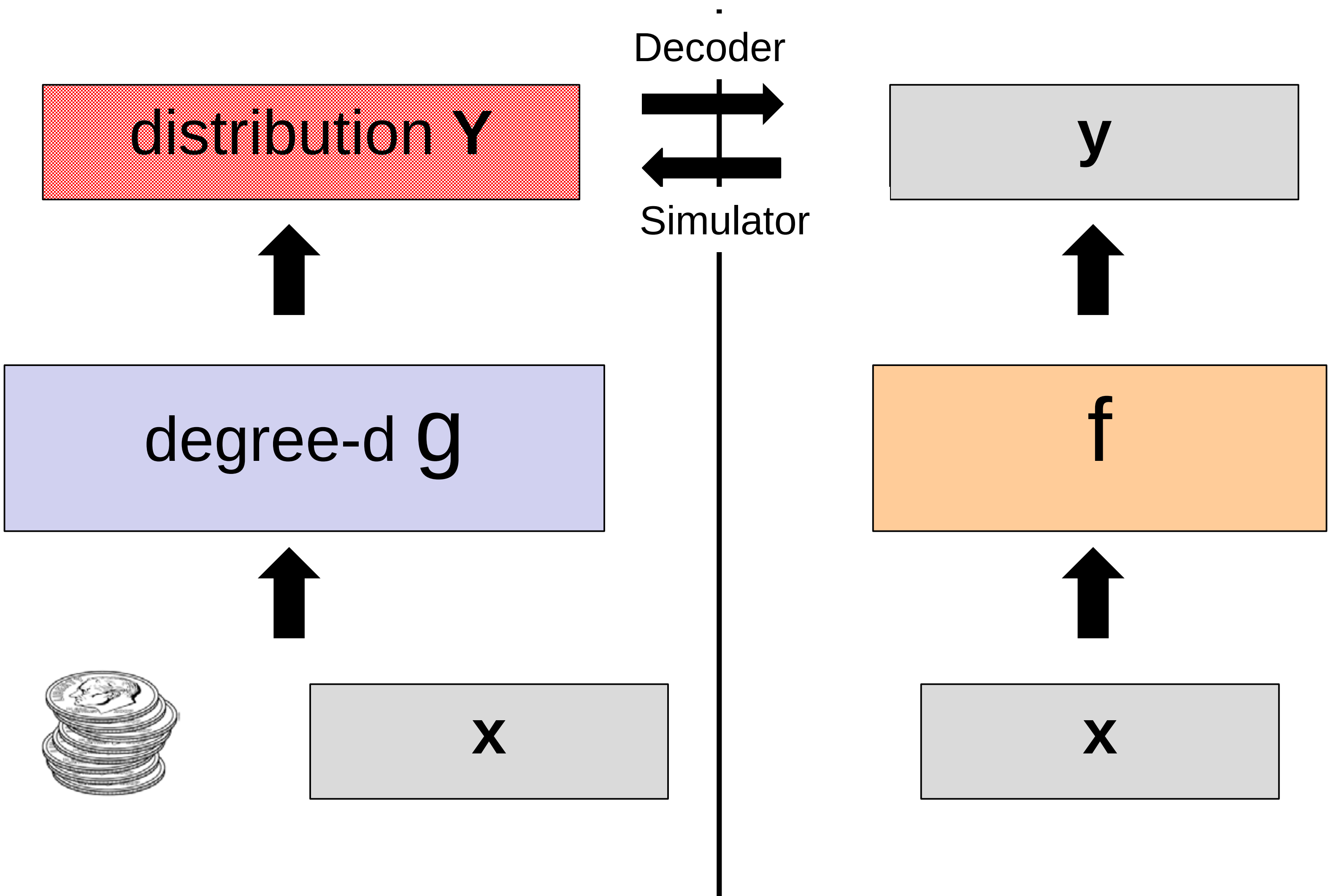
Randomizing Polynomials [IK00]

MPC for g $\Rightarrow$ MPC for f



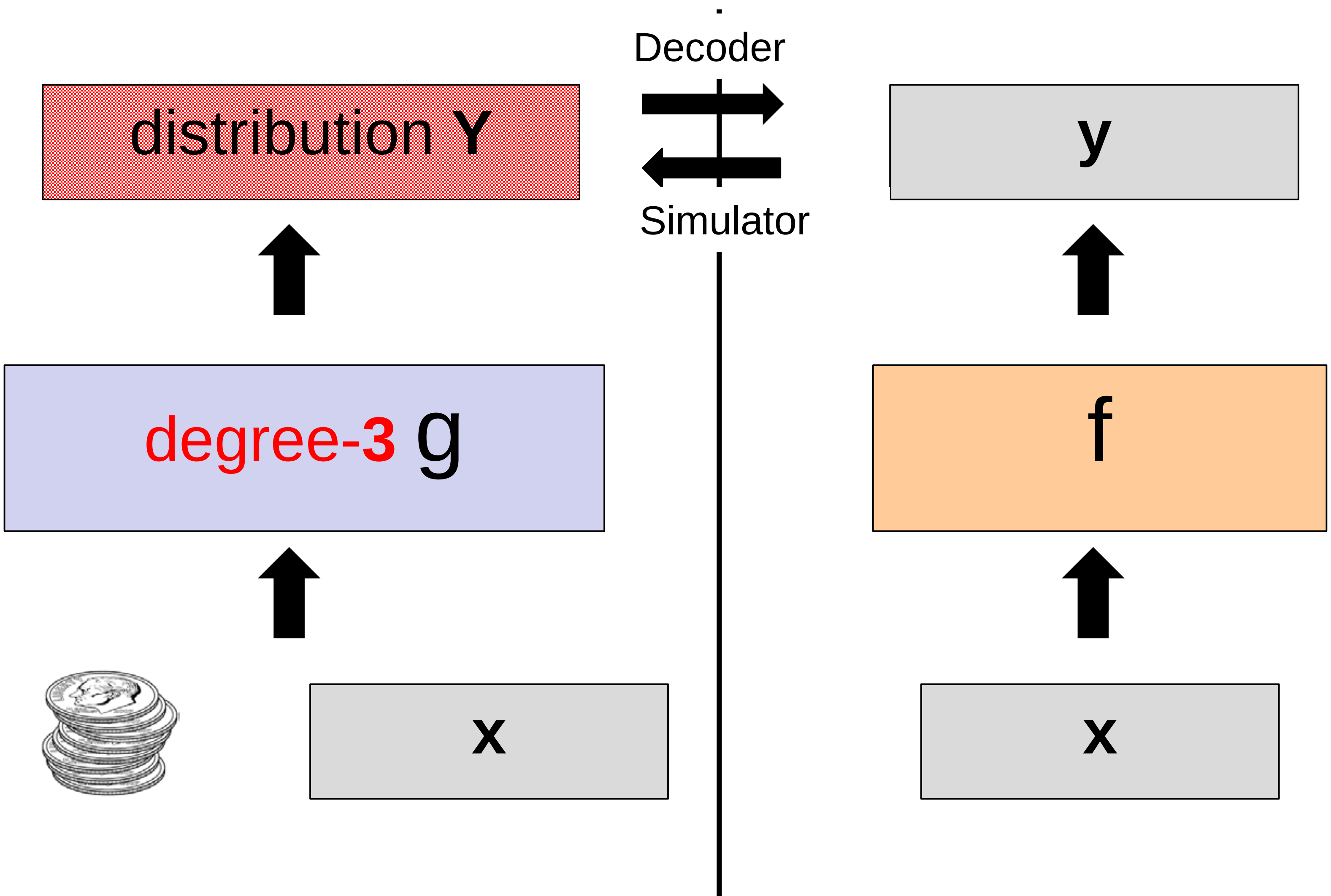
Randomizing Polynomials [IK00]

g has d-round protocol ⇒ f has d-round protocol !

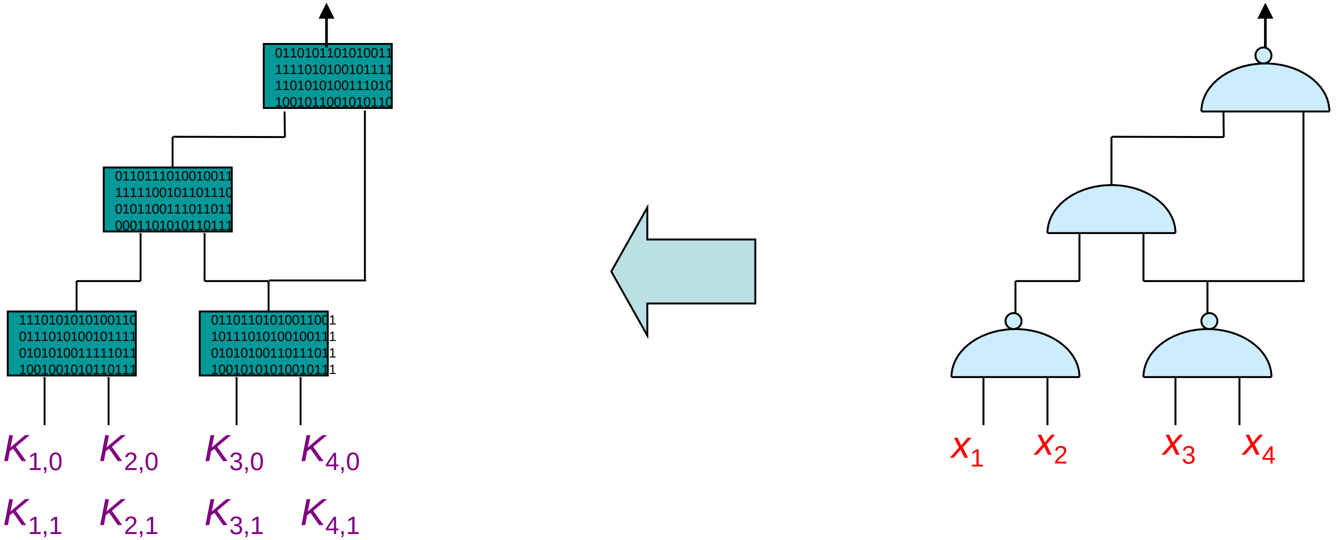


Randomizing Polynomials [IK00]

Thm [IK02] Every f has perfect RP of degree 3

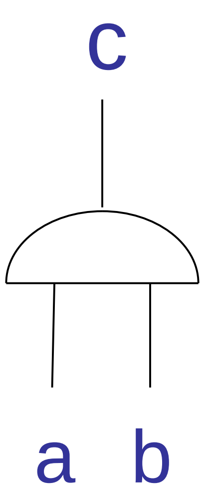


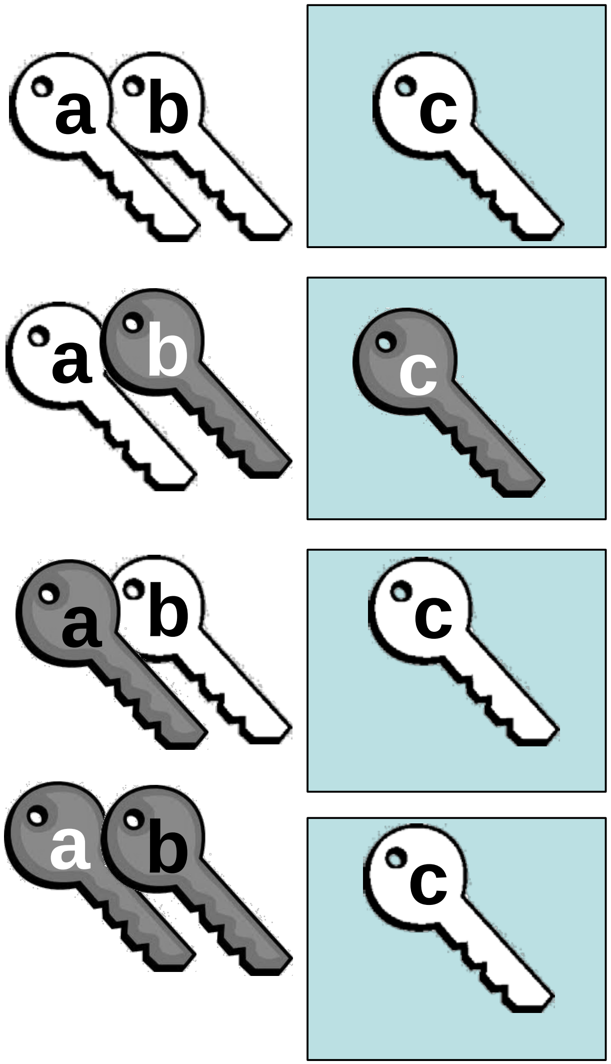
Degree-3 RP from Information-Theoretic Garbled Circuit [IK02]



$g(x,(k_{i,b},r)) = ((k_{i,x_i})_{i=1..n}$, garbled tables)

GC-based Randomized Encoding





Randomness per wire:

- random mask bit
- 2 keys

Release:

per input wire release:

- corresponding key

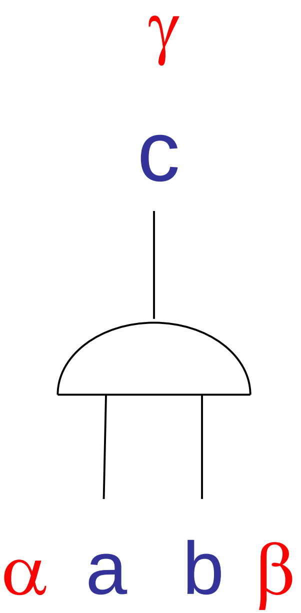
per gate:

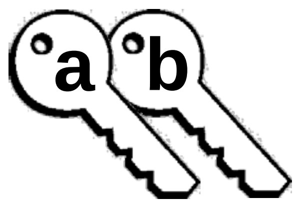
- 4 ciphertexts
- deg-3=deg(gate)+1

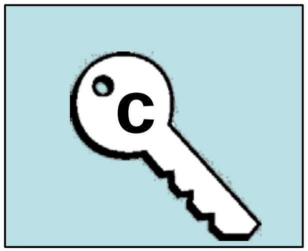
per output wire:

- release mask bit

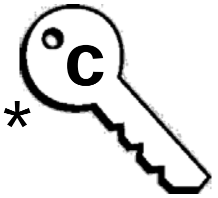
GC-based Randomized Encoding







$G(\alpha,\beta)*$



$+[1-G(\alpha,\beta)]*$

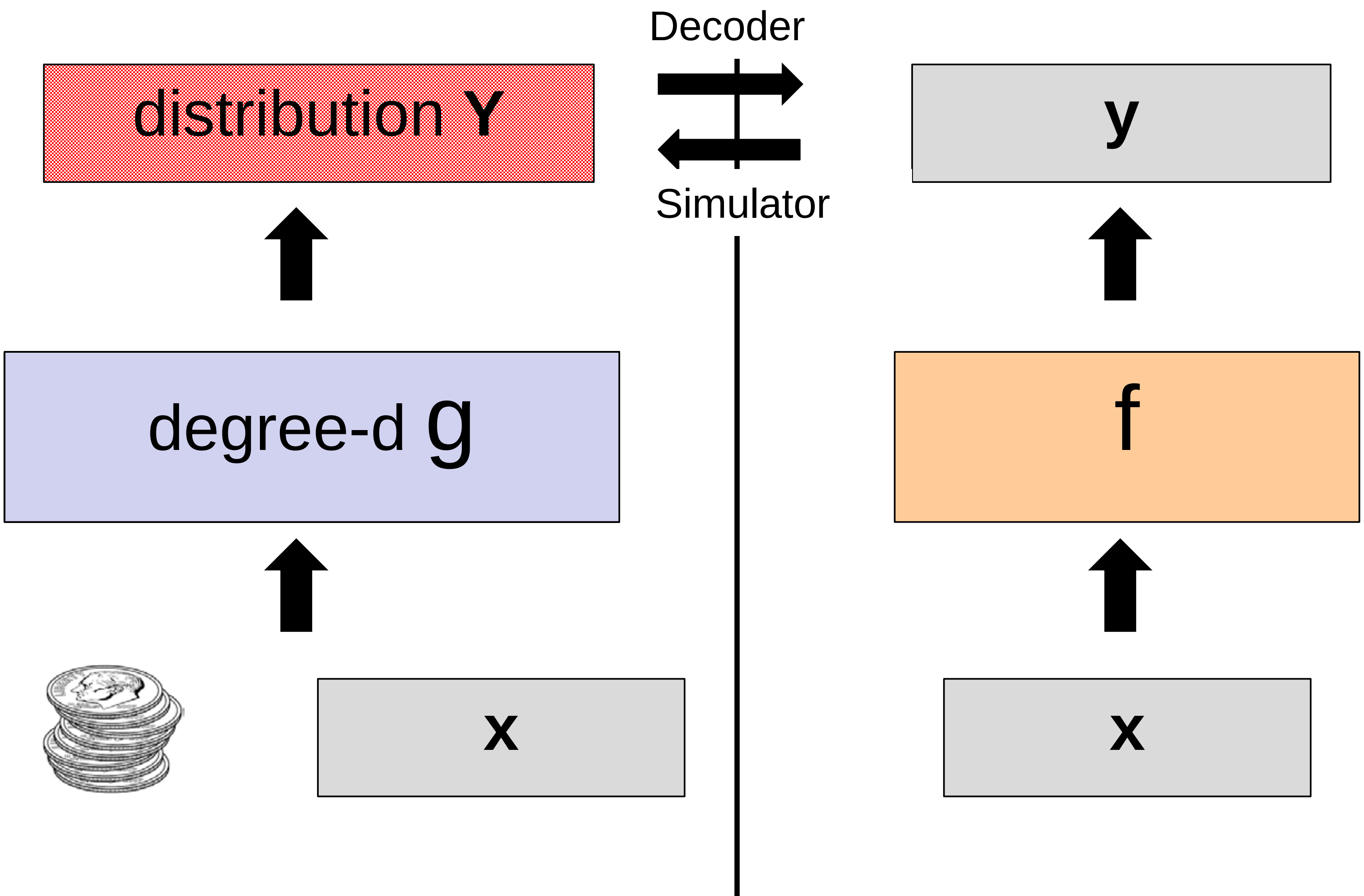
per gate:

- 4 ciphertexts
- deg-3=deg(gate)+1

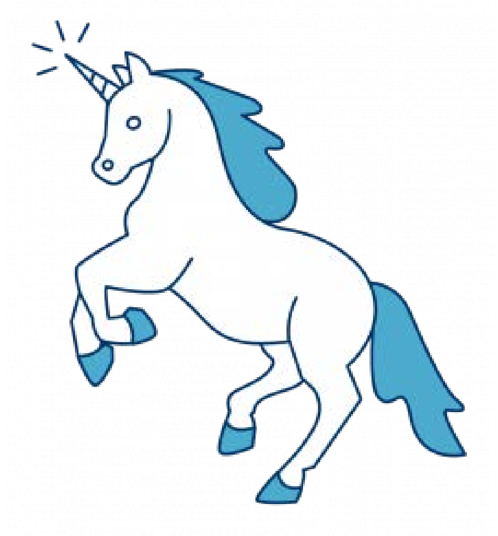
Randomizing Polynomials

Many other applications (e.g., parallel crypto)

- decouple simplicity from semantics !



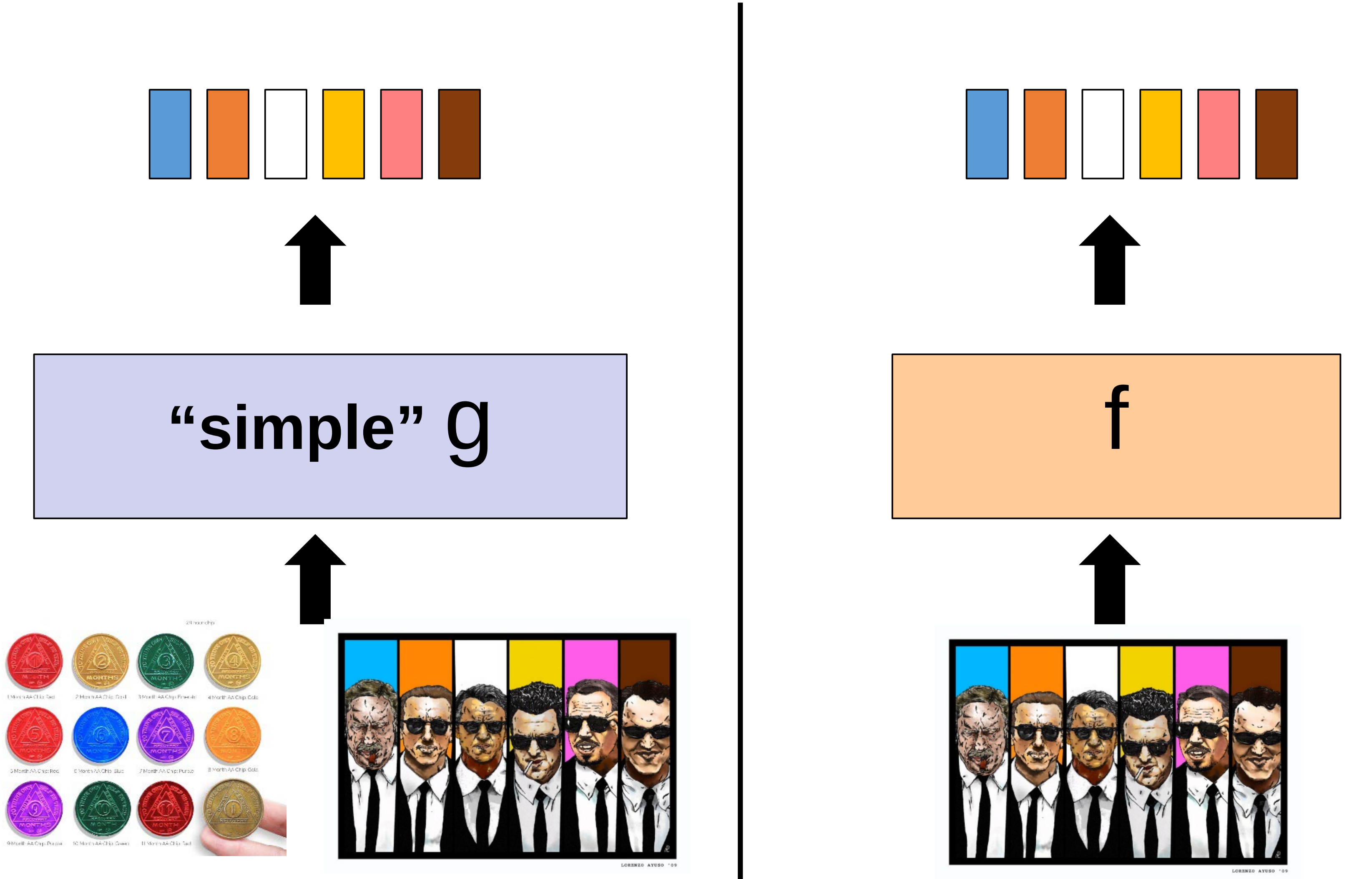
Problem:
For most functions,
NO degree-2 perfect RE's



Sol: Compromise!
Aim for a weaker notion

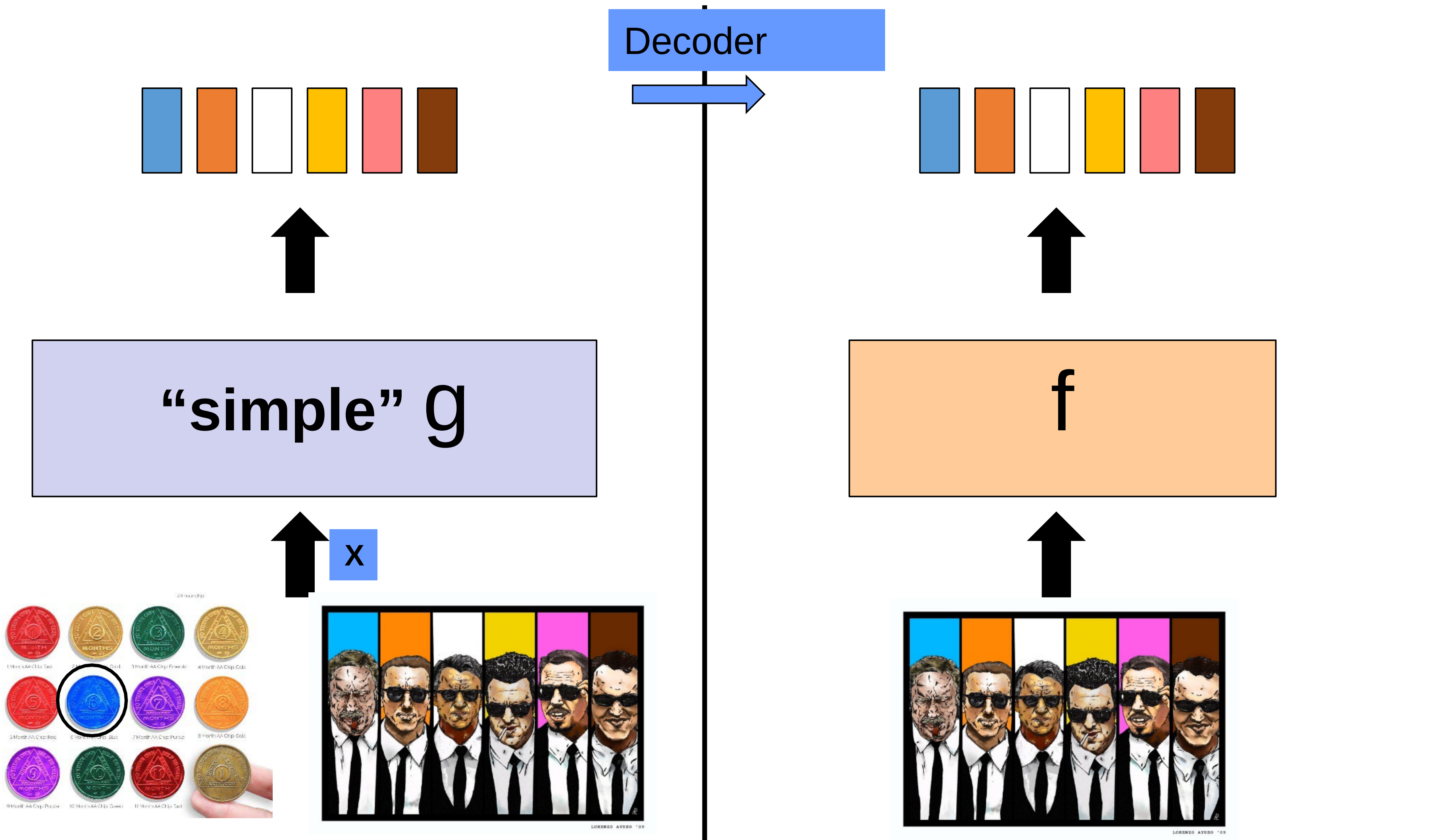
Multiparty Randomized Encoding (MPRE)

Relaxed correctness: Each party has a decoder



Multiparty Randomized Encoding (MPRE)

Relaxed correctness: Each party has a decoder



Multiparty Randomized Encoding (MPRE)

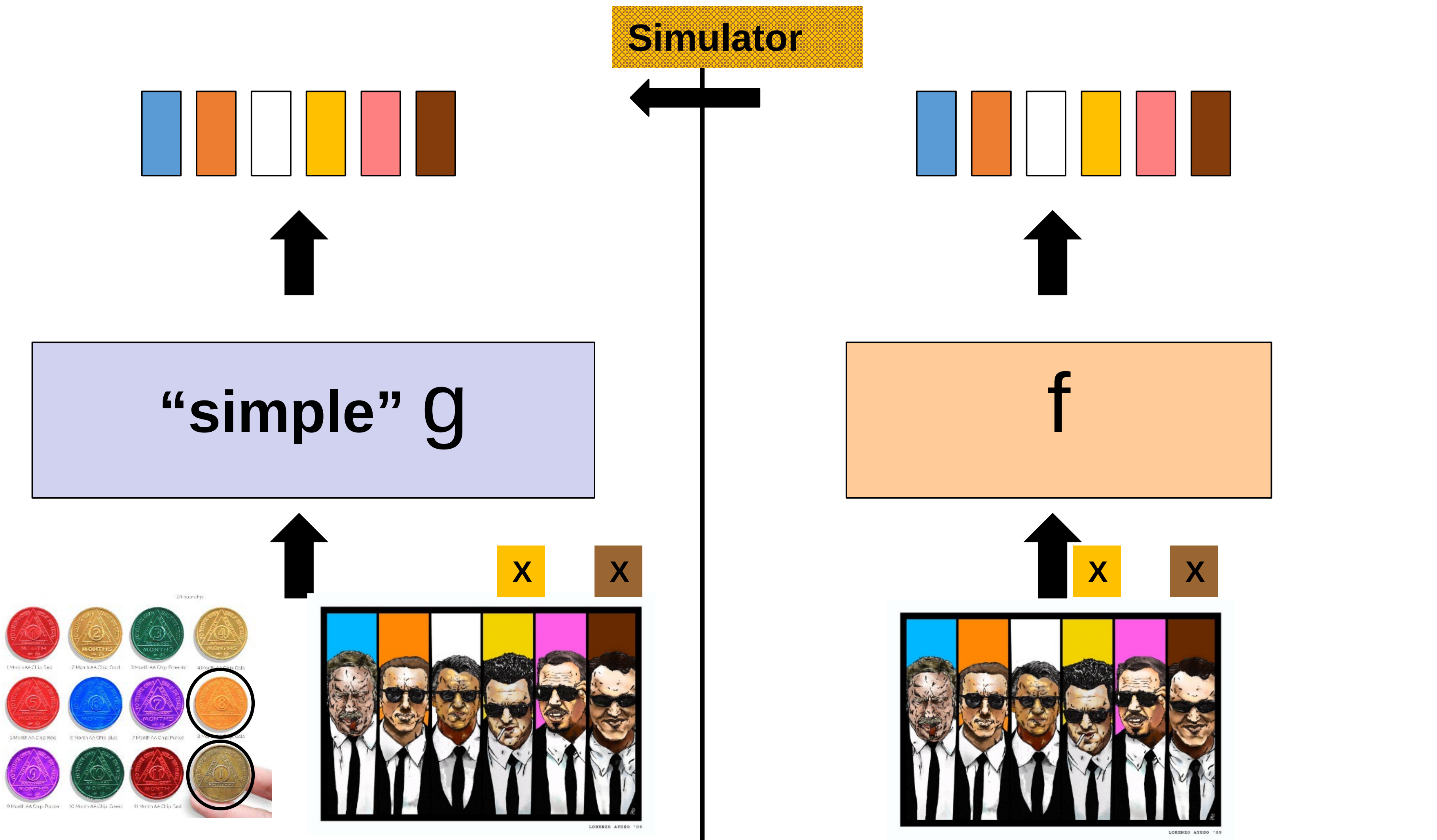
Relaxed privacy: Every minority has a simulator

“simple” g

f

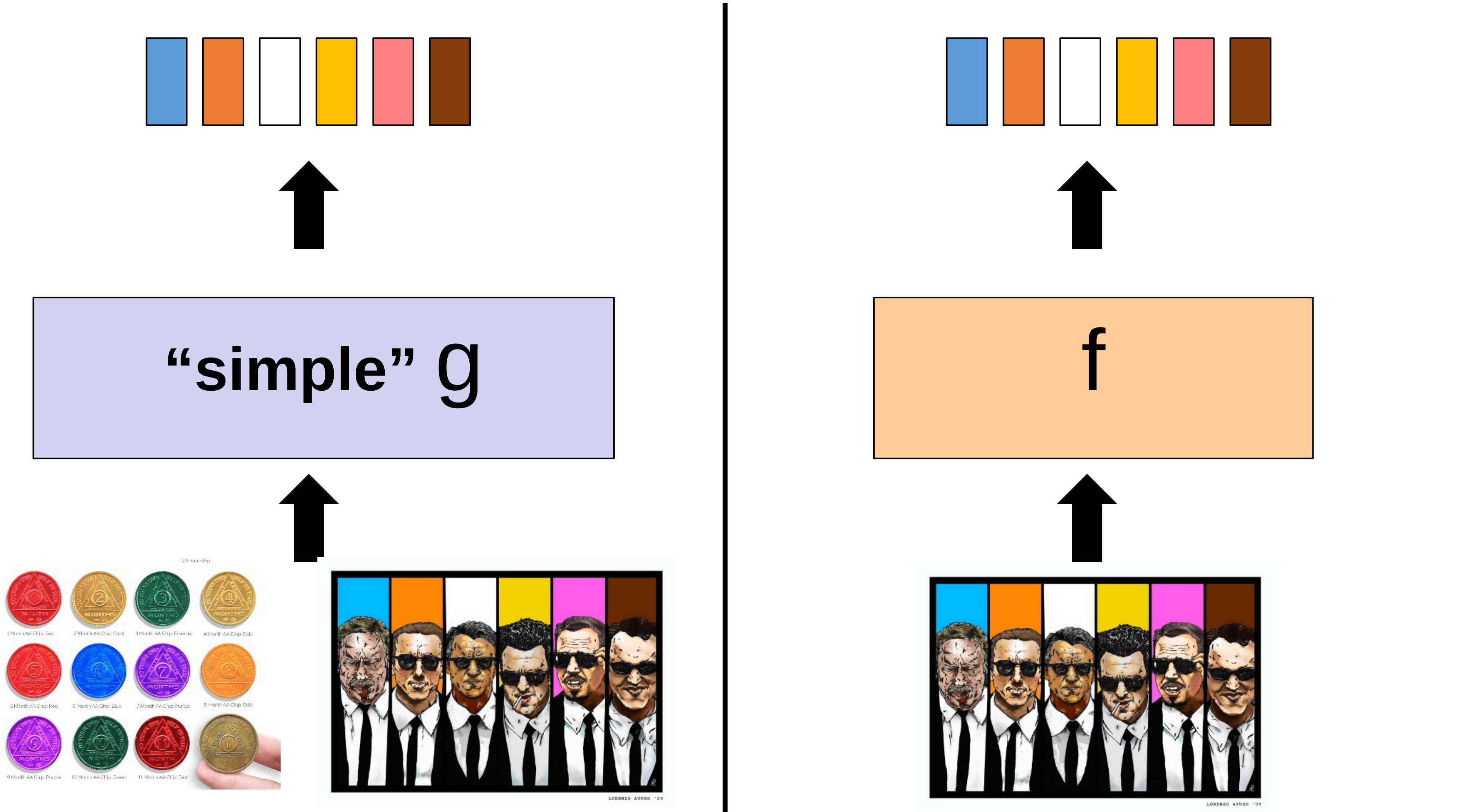
Multiparty Randomized Encoding (MPRE)

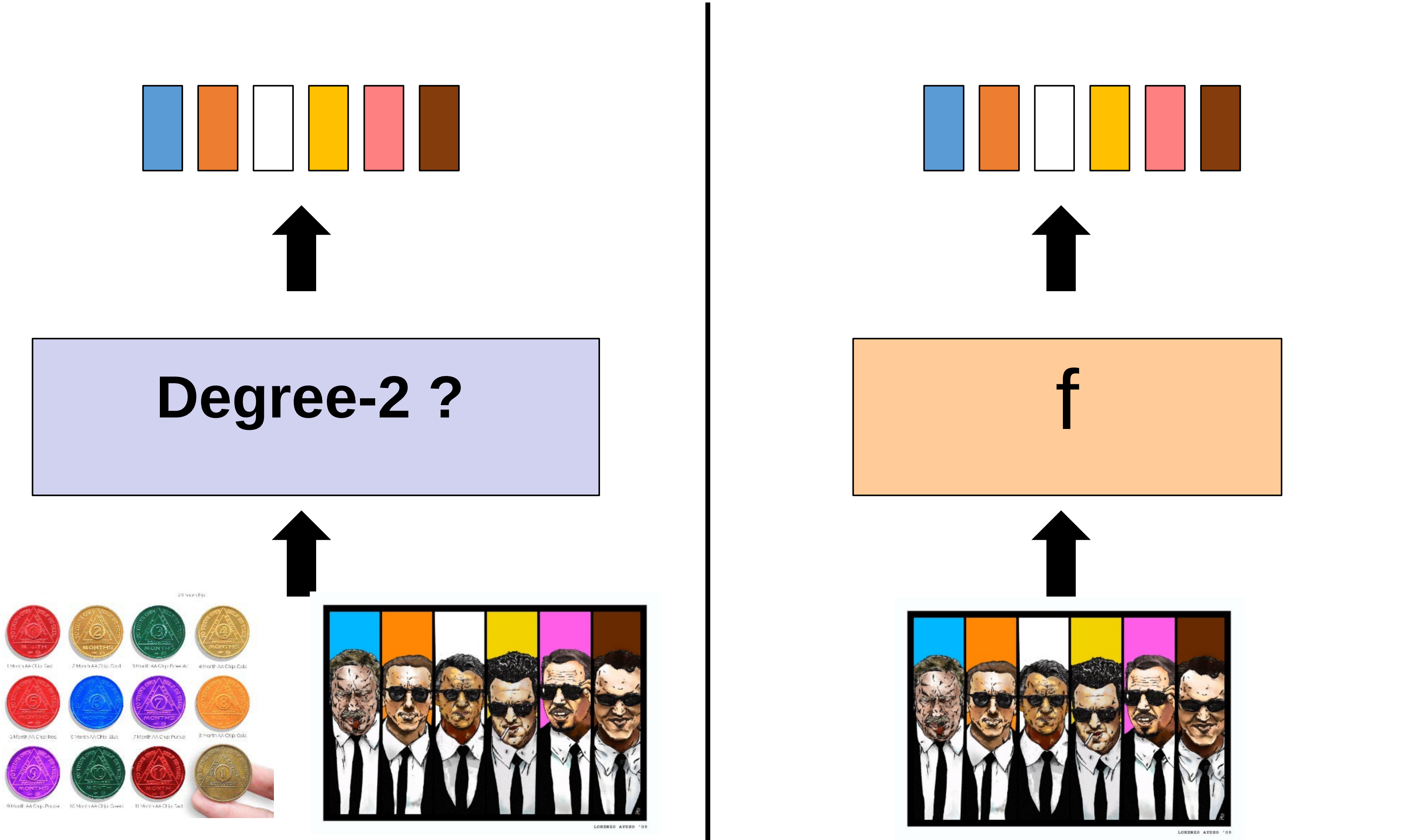
Relaxed privacy: Every minority has a simulator



MPRE **relaxes** Randomized Encoding

- Encodes functionality
- RE is a special case of MPRE
- Protocol for g $\Rightarrow$ Protocol for f



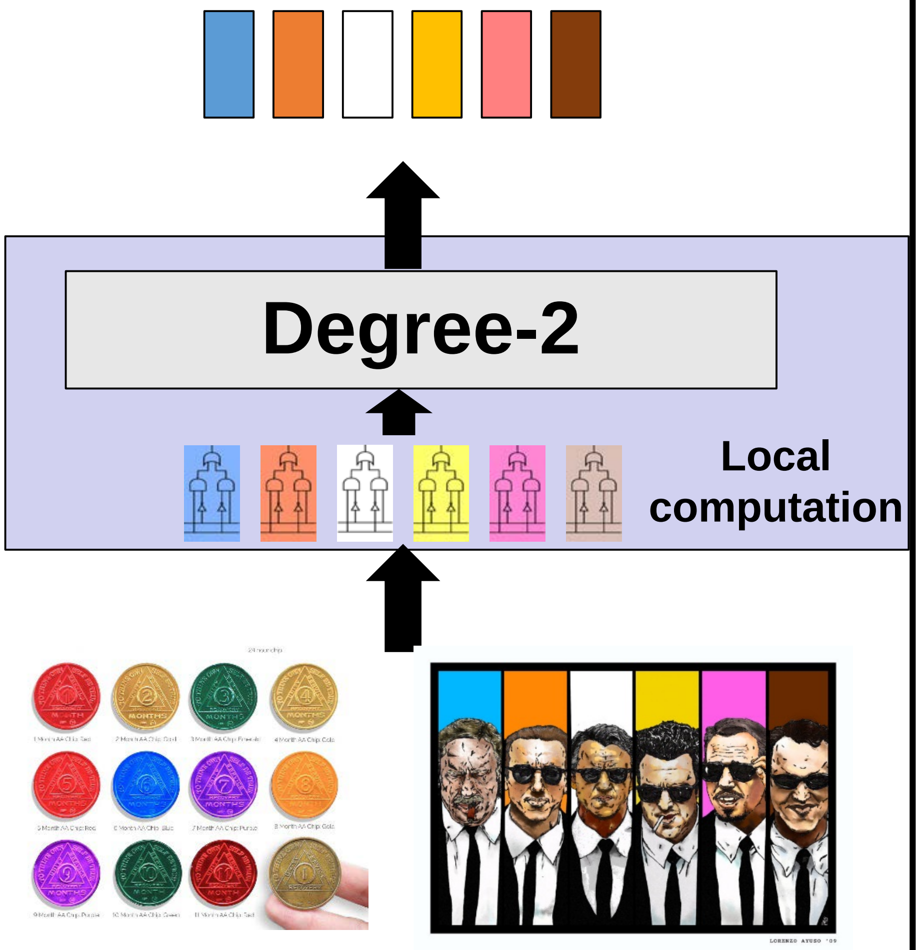


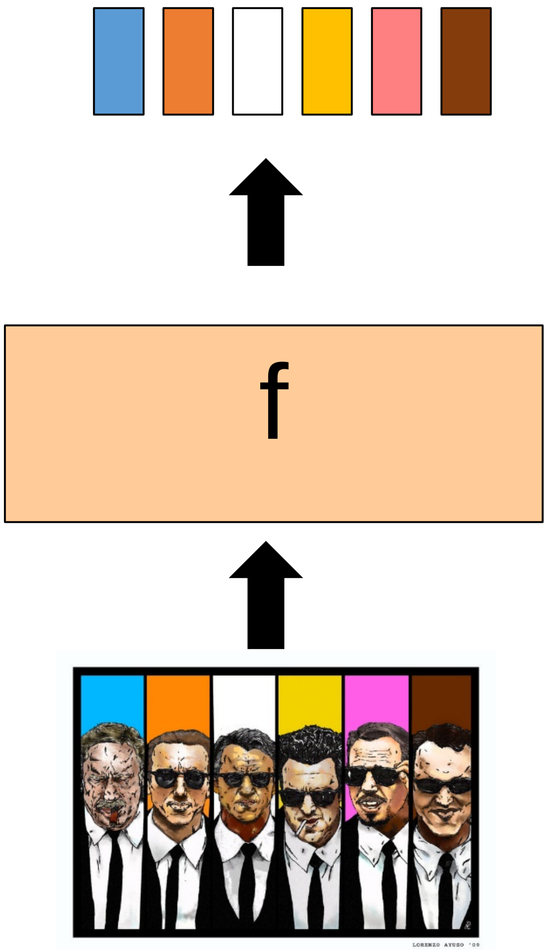
Degree-2 ?
f

Thm: every functionality has MPRE of “effective” deg-2

- Efficient for NC1
- Efficient computational-MPRE for general circuits

⇒ 2-round honest-majority protocol



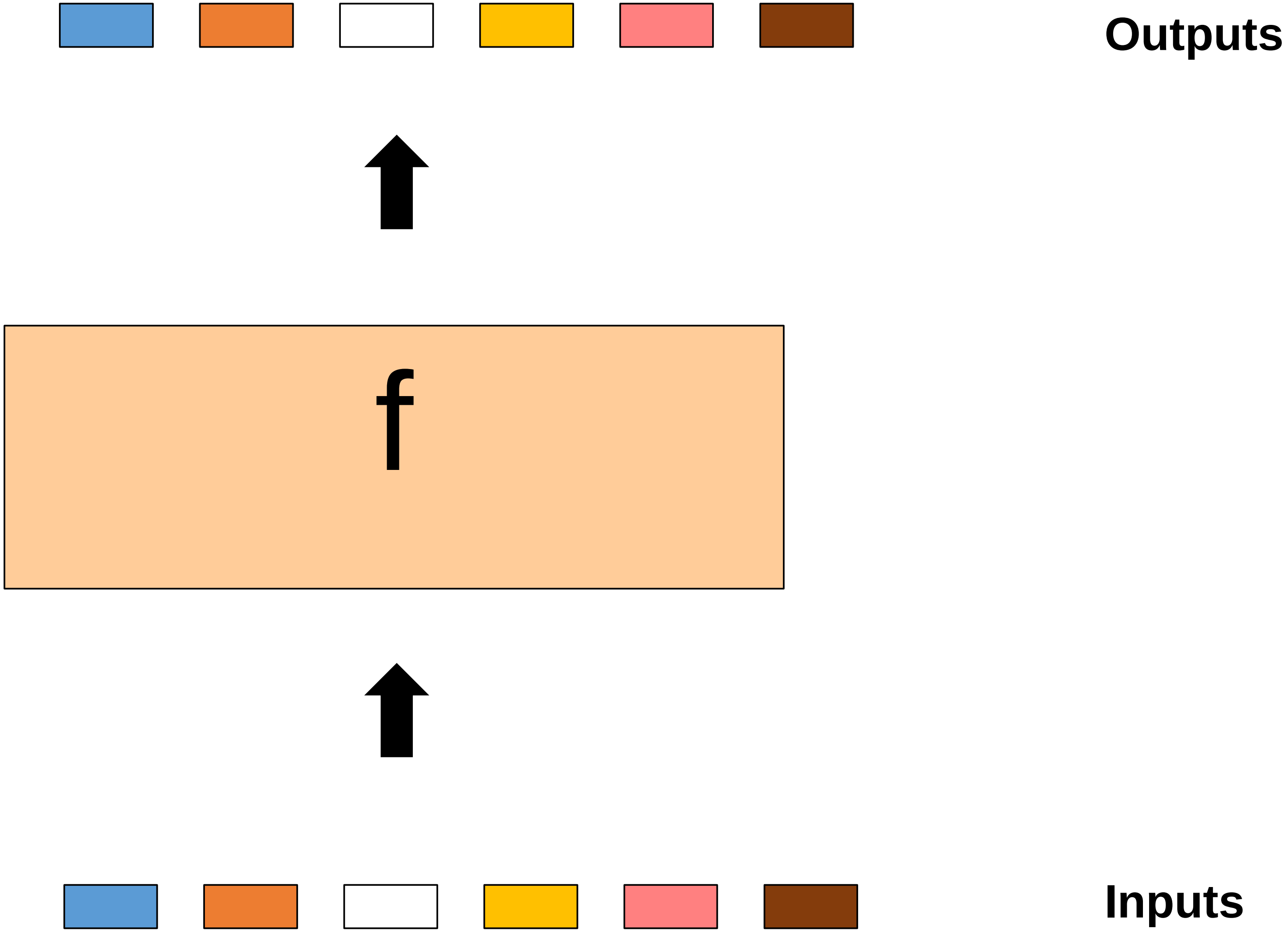


Proof Idea

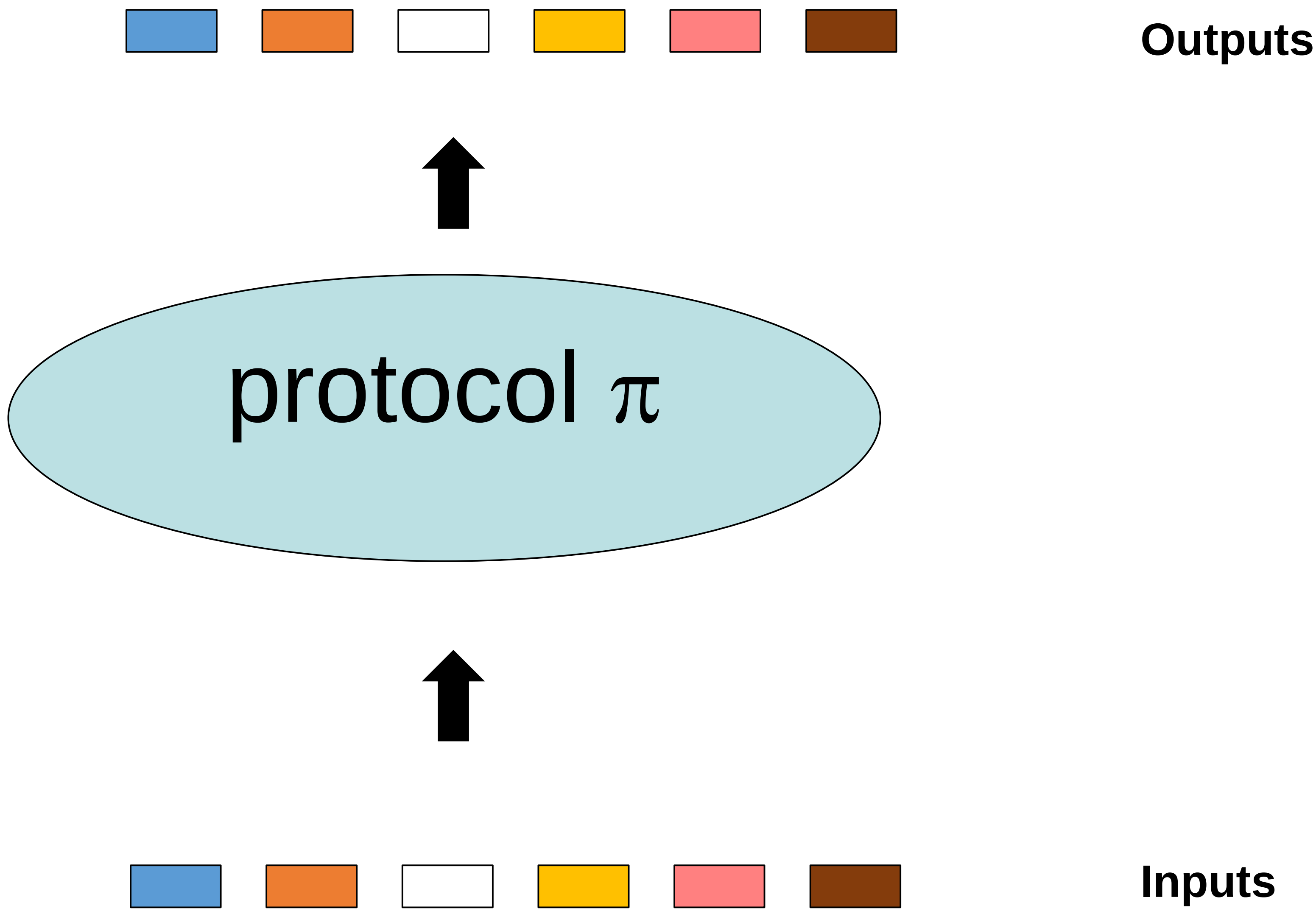
1. From protocols to “nice”-MPRE

2. From “nice”-MPRE to deg-2 MPRE

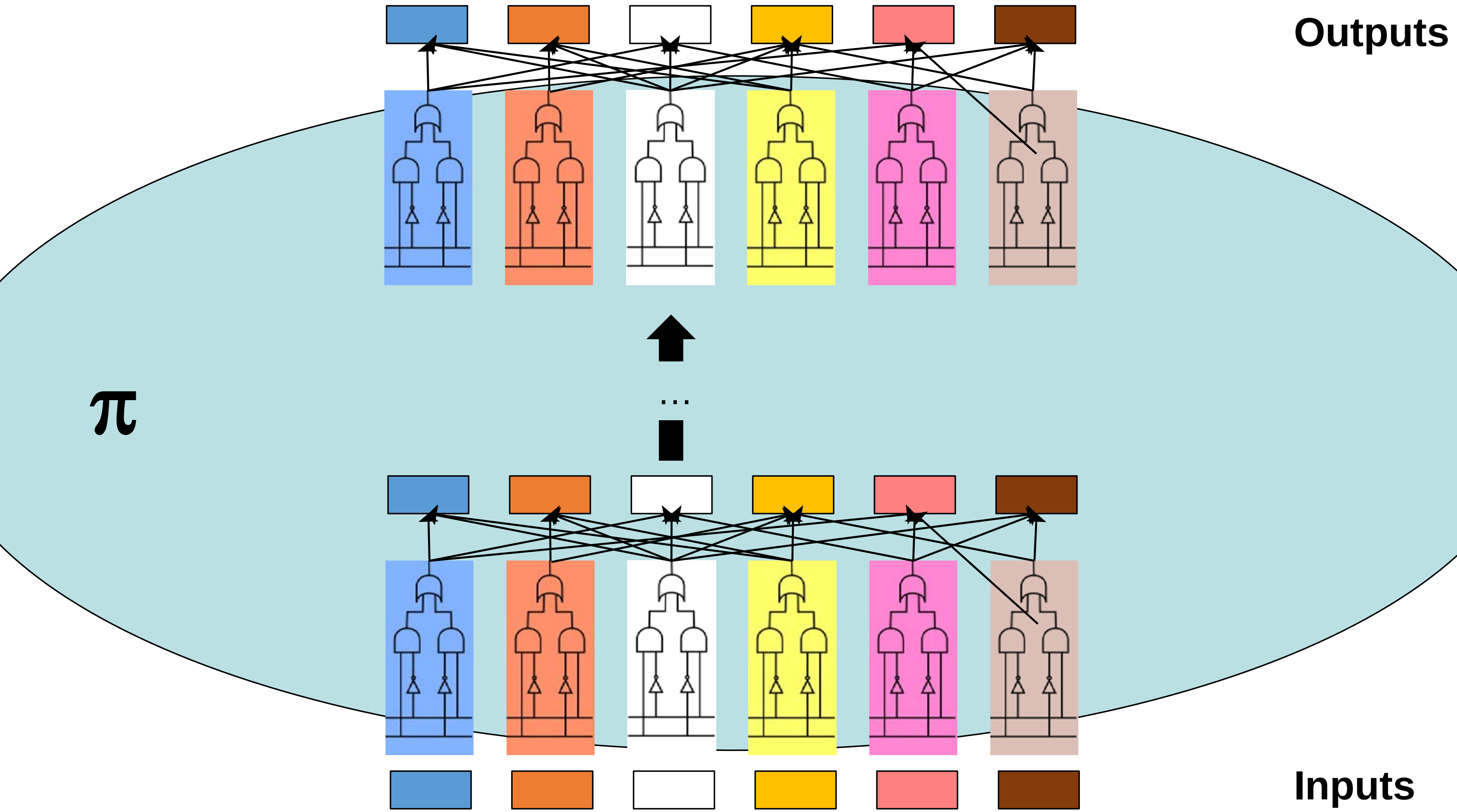
Step 1: Protocol Induced MPRE



Step 1: Protocol Induced MPRE

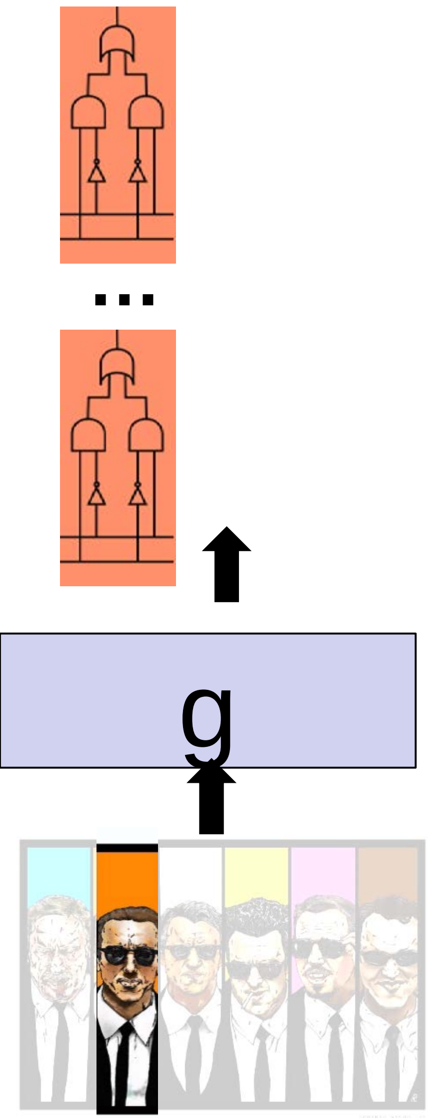


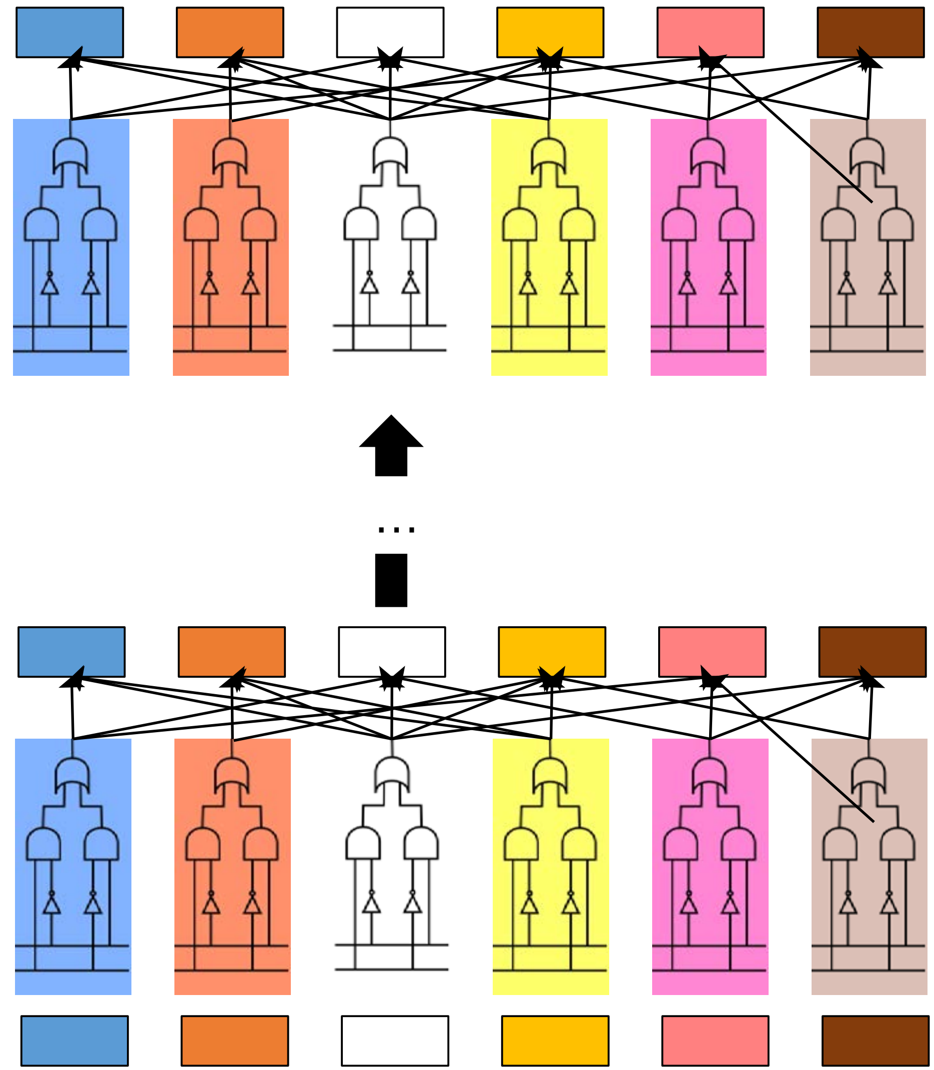
Step 1: Protocol Induced MPRE



Step 1: Protocol Induced MPRE

Let g be MPRE that gives to a party its view & intermediate values



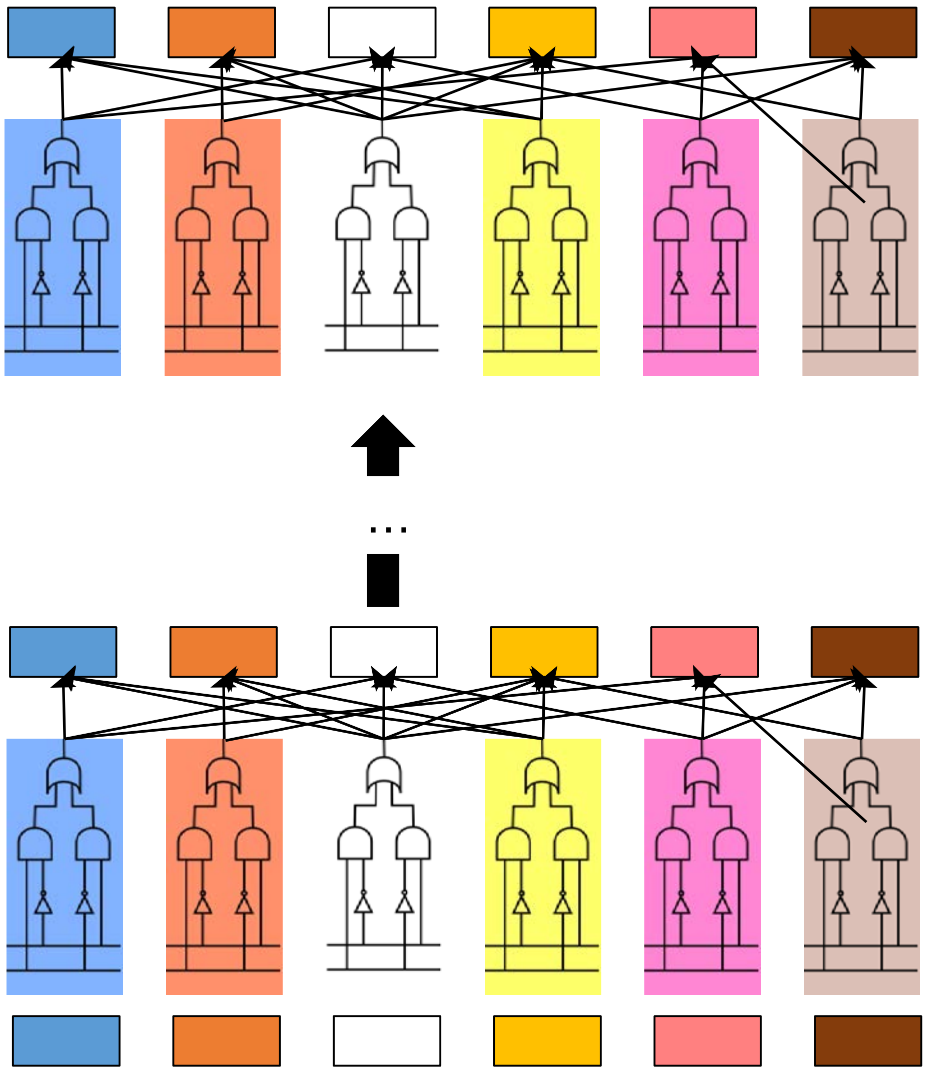


Key observation

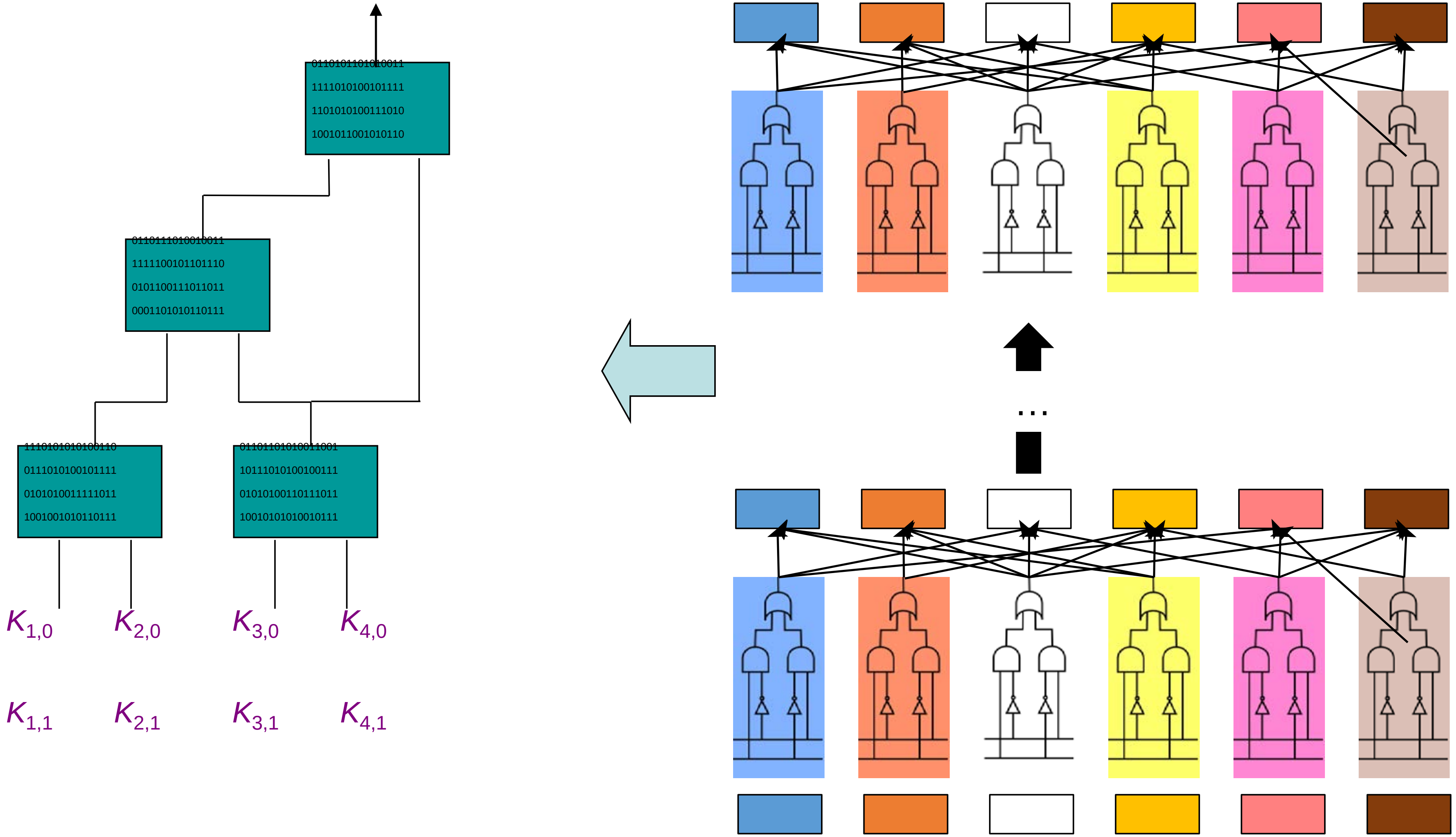
The MPRE is "simple":

Each output y is either:

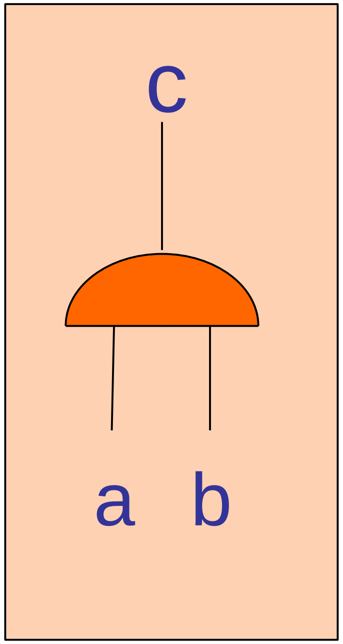
- output of local computation
- Value sent by another party



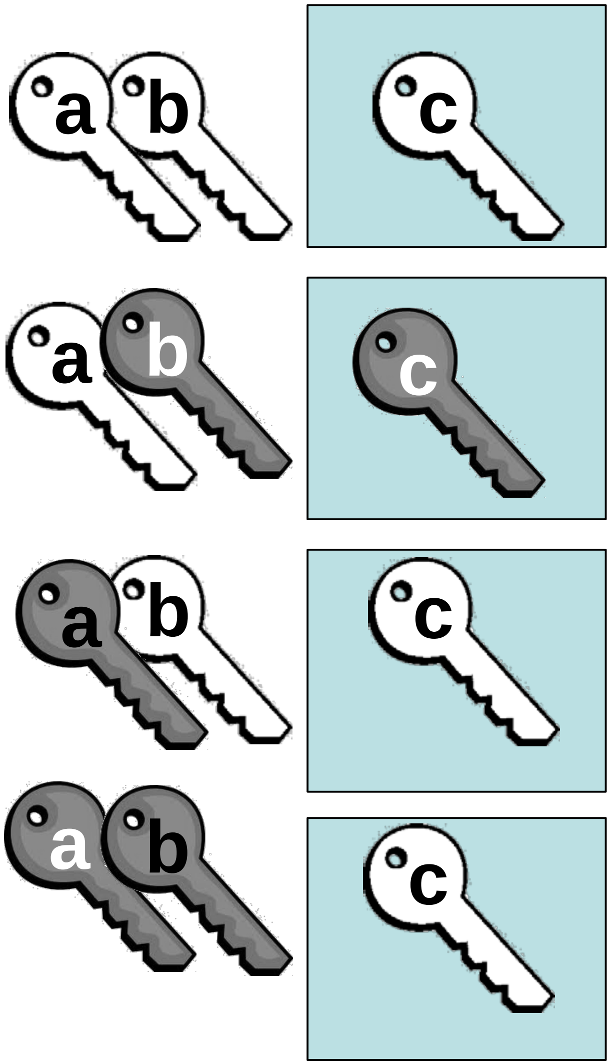
Step 2: re-encode via perfect Garbled Circuit



Case 1: Local Computation Gates



Orange party



Randomness per wire:

- mask bit

owned by orange party

- 2 keys

shared between all

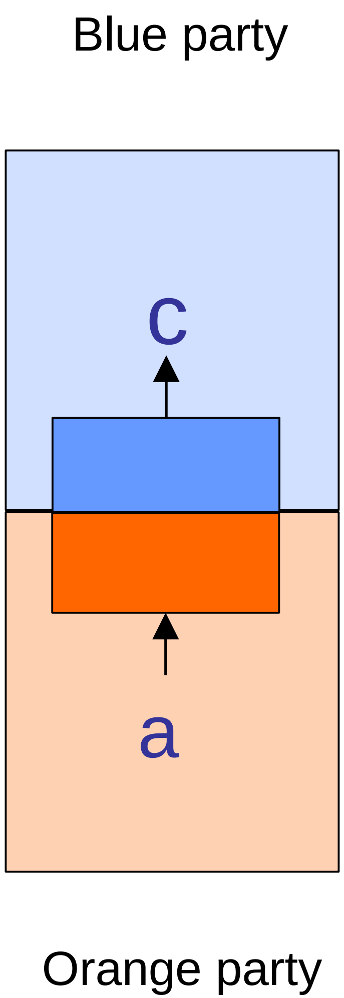
Release:
4 ciphertexts per gate
For all parties

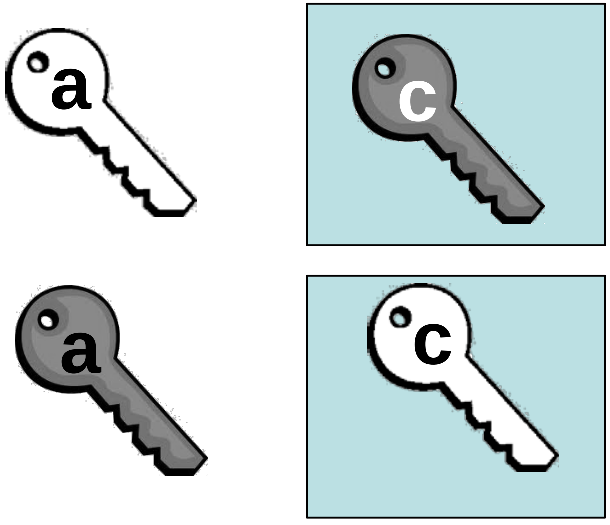
Degree-2
(after preprocess)!

- since masks of a,b,c

are known to same party

Case 2: Transmission Gates





Randomness per wire:

- mask bit

owned by orange/blue

- 2 keys

shared between all

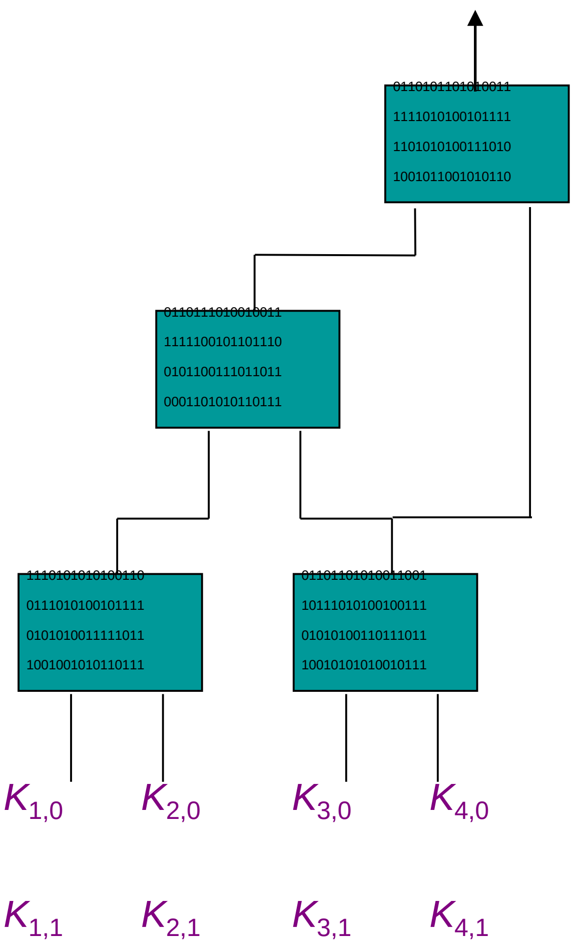
Release for all:
2 ciphertexts per gate

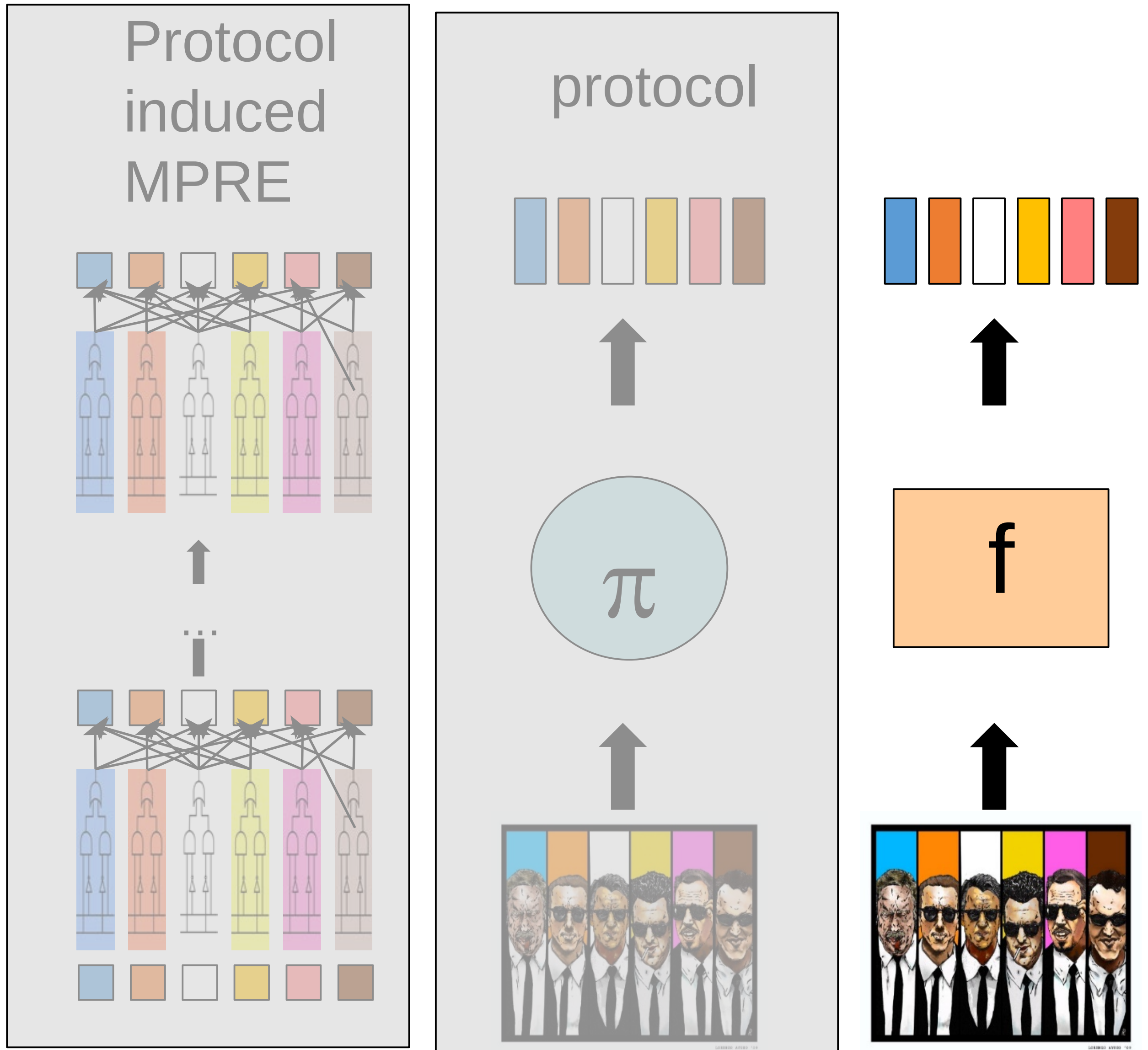
Degree-2 !

- since deg(gate)=1

Putting it all together

GC-based MPRE
Effective deg-2





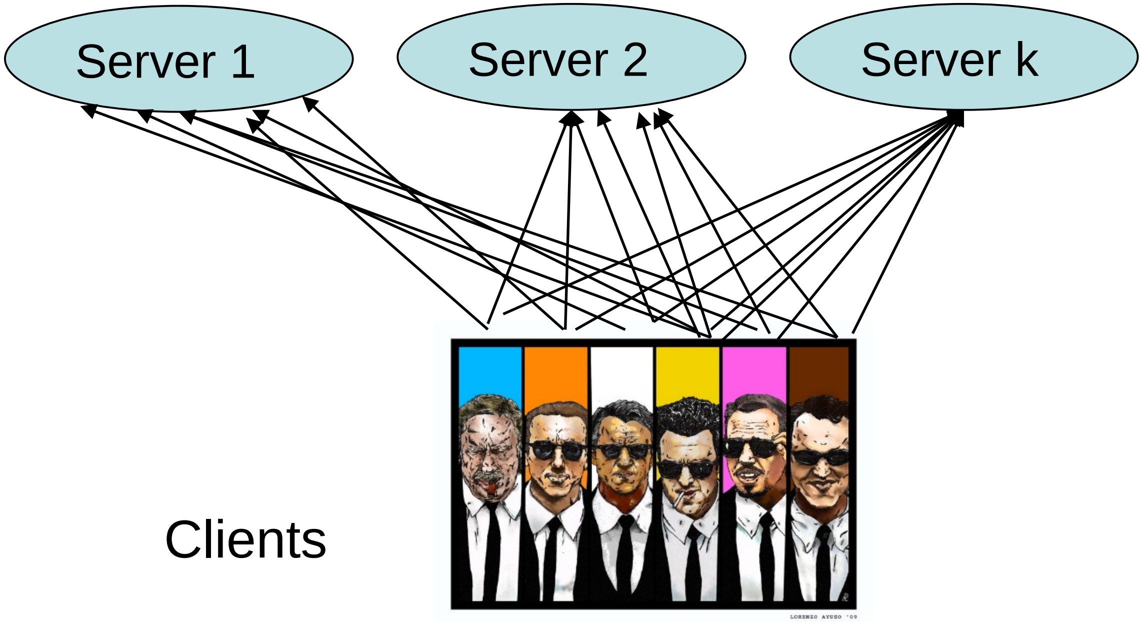
Conclusion

Assuming honest majority and passive adversary:

- Every function has perfect 2-round protocol
  - Efficient for NC1, log-space
  - Computational variant for poly-size circuits using OWFs

Conclusion

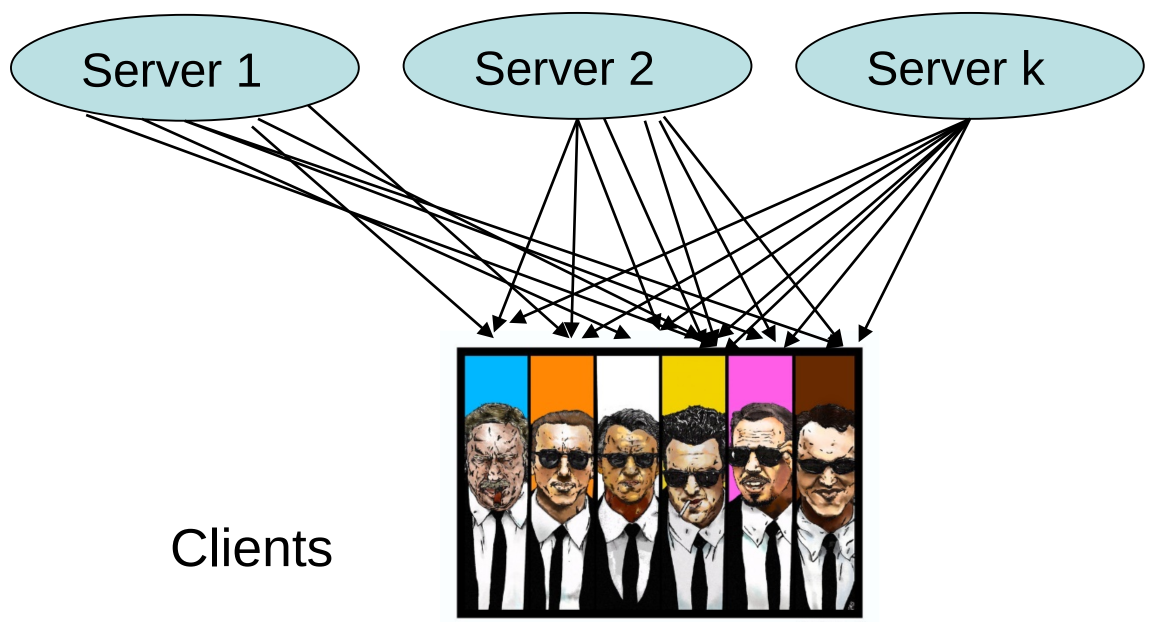
- Practical relevance?
  - 2-round protocols easily transfer to client-server model [Ishai-Damgard '05]



Conclusion

- Practical relevance?
  – 2-round protocols easily transfer to client-server model [Ishai-Damgard ‘05]

Private as long as majority of the servers & clients are honest



Conclusion

Multiparty Randomized Encoding of Functionalities

- Useful concept

- Other applications?

Thank You