Randomized Encoding of Functions
&
MPC with Low Round Complexity

Benny Applebaum

Tel Aviv University

Back to Yuval's Talk

Alibaba Cloud box Dropbox iCloud Google Drive OneDrive Yandex.Disk

x_1	x_2	x_3	x_4	x_5	x_6	x_7

Completeness Theorems [1988]

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

Every function f can be perfectly computed

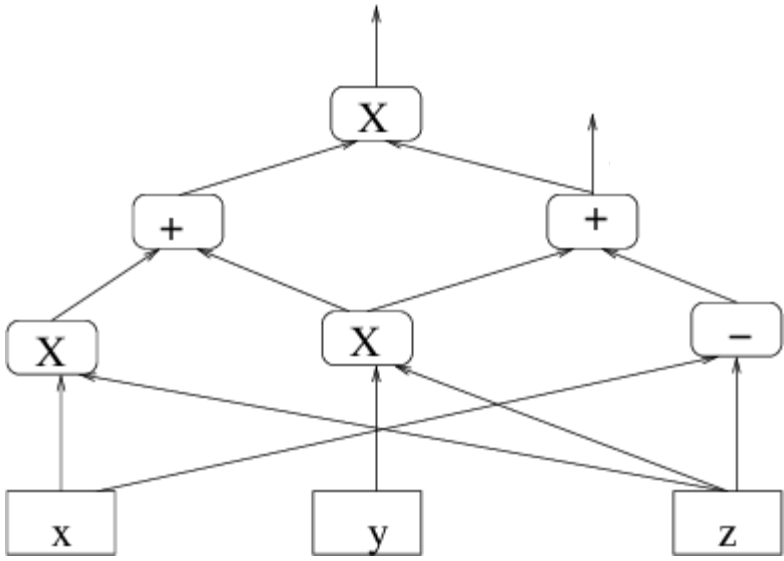
- Passive adversary T<N/2 (honest majority)
- Active adversary T<N/3 (strong majority)

Tight: Bounds on T are optimal

Complexity: poly(circuit-size(f))

Rounds: Multiplicative depth of f

~ log degree+1



Interaction is Expensive

Can we get constant-round protocol?

What is the best achievable round complexity R_{MPC}?

- $R_{MPC}>1$ even for weakest security notion

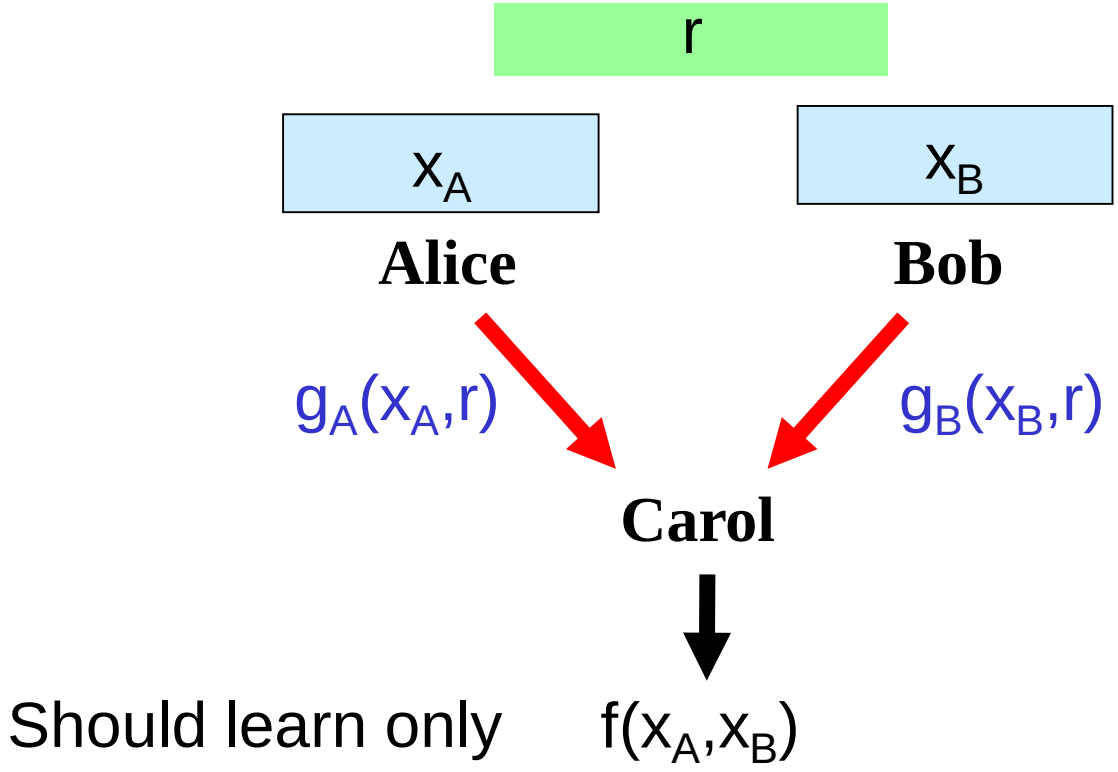
Non-Interactive IT MPC in some model ?

Questions valid even with LARGE communication

Non-Interactive MPC

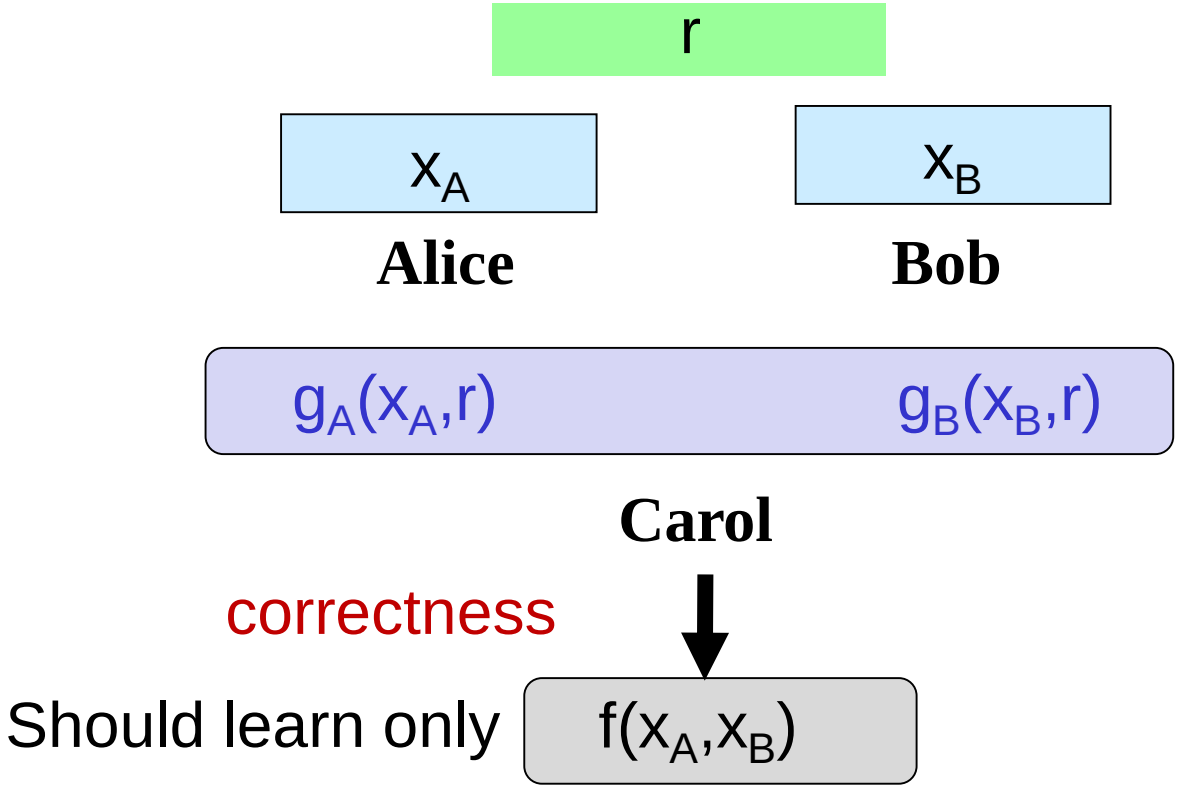
Private Simultaneous Message Protocols

"minimal model for secure computation" [Fei-Kil-Nao94, Ish-Kus98]



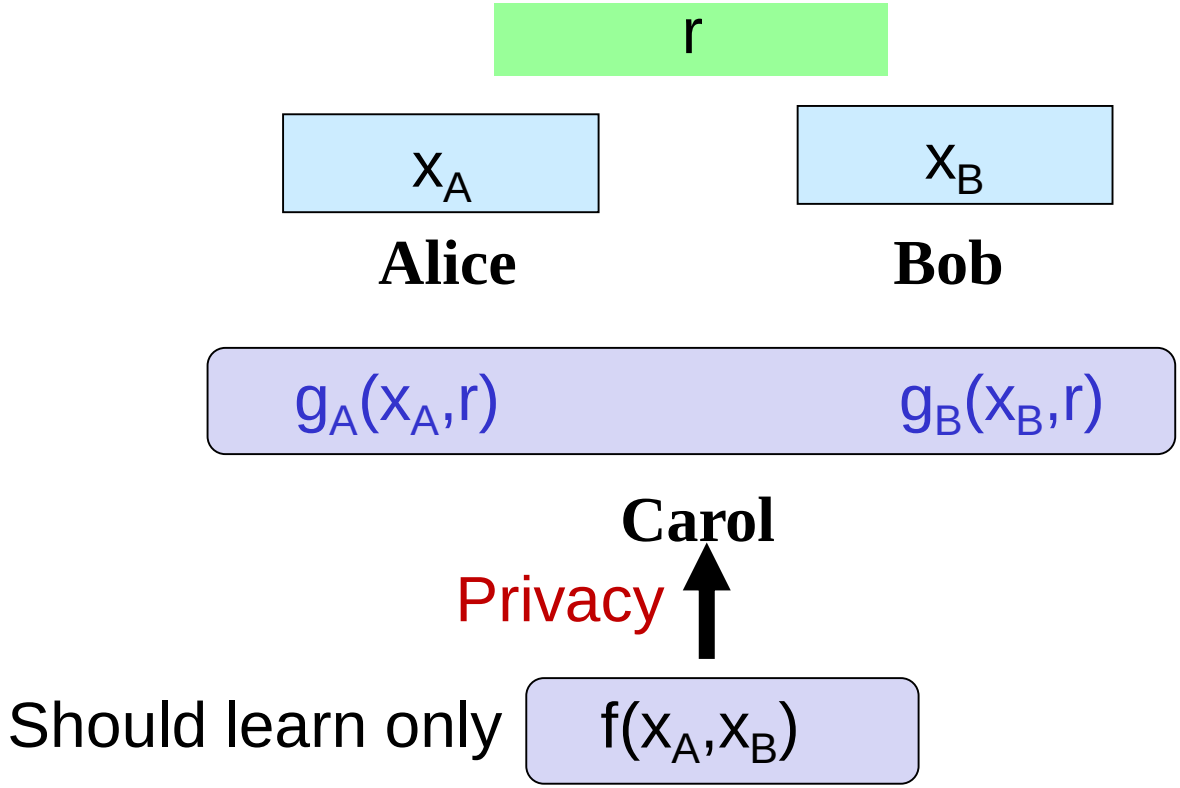
Private Simultaneous Message Protocols

"minimal model for secure computation" [Fei-Kil-Nao94, Ish-Kus98]



Private Simultaneous Message Protocols

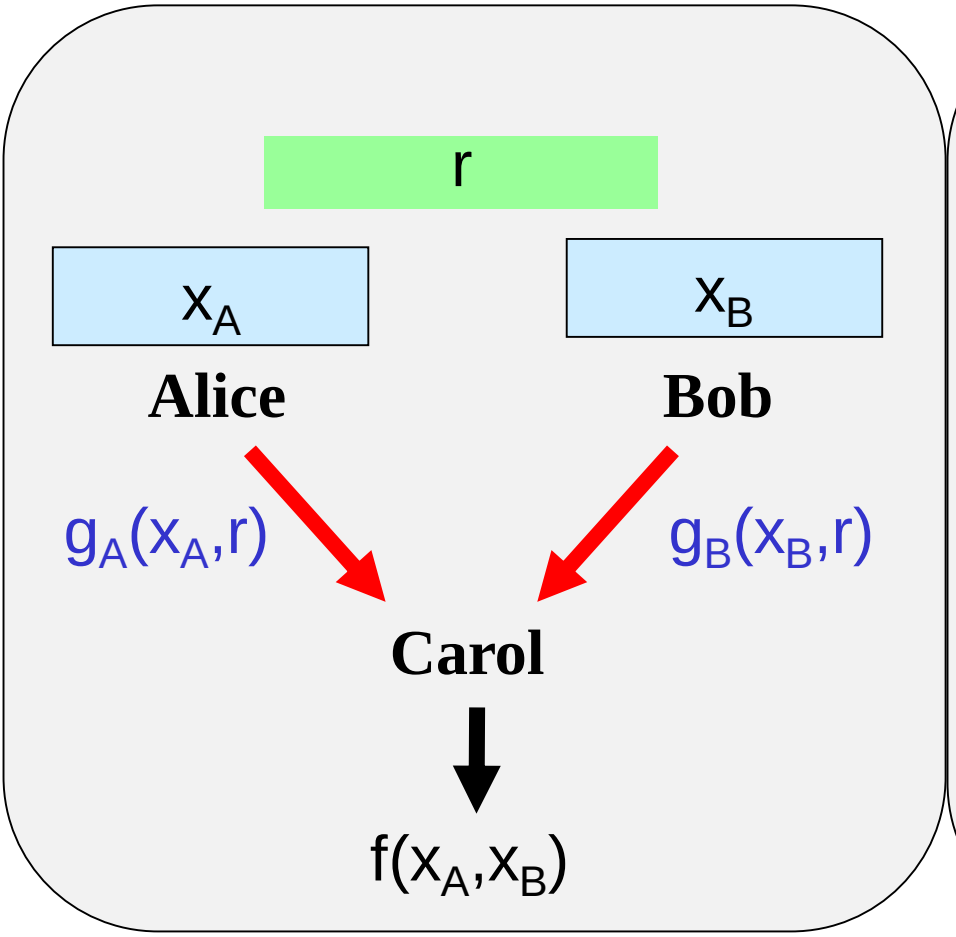
"minimal model for secure computation" [Fei-Kil-Nao94, Ish-Kus98]

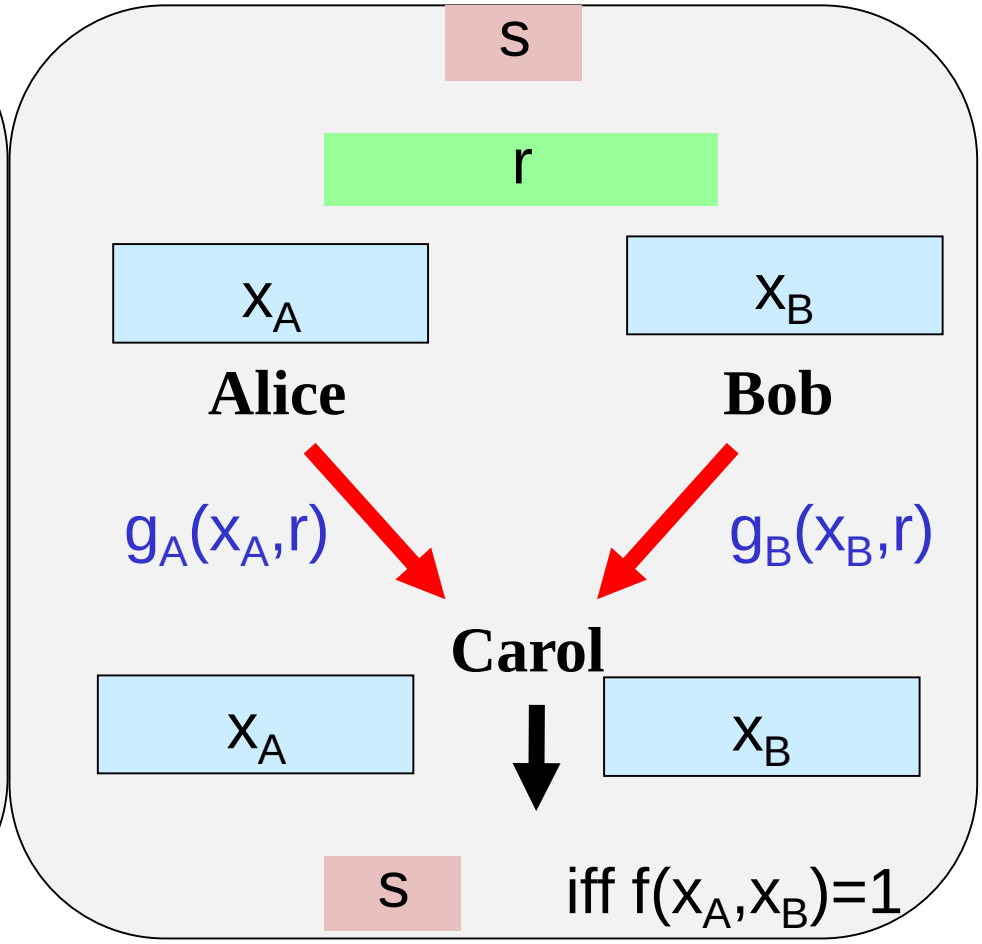


PSM vs CDS

CDS(f)≤ PSM (g)

where g(x,(y,s))= s iff f(x,y)=1



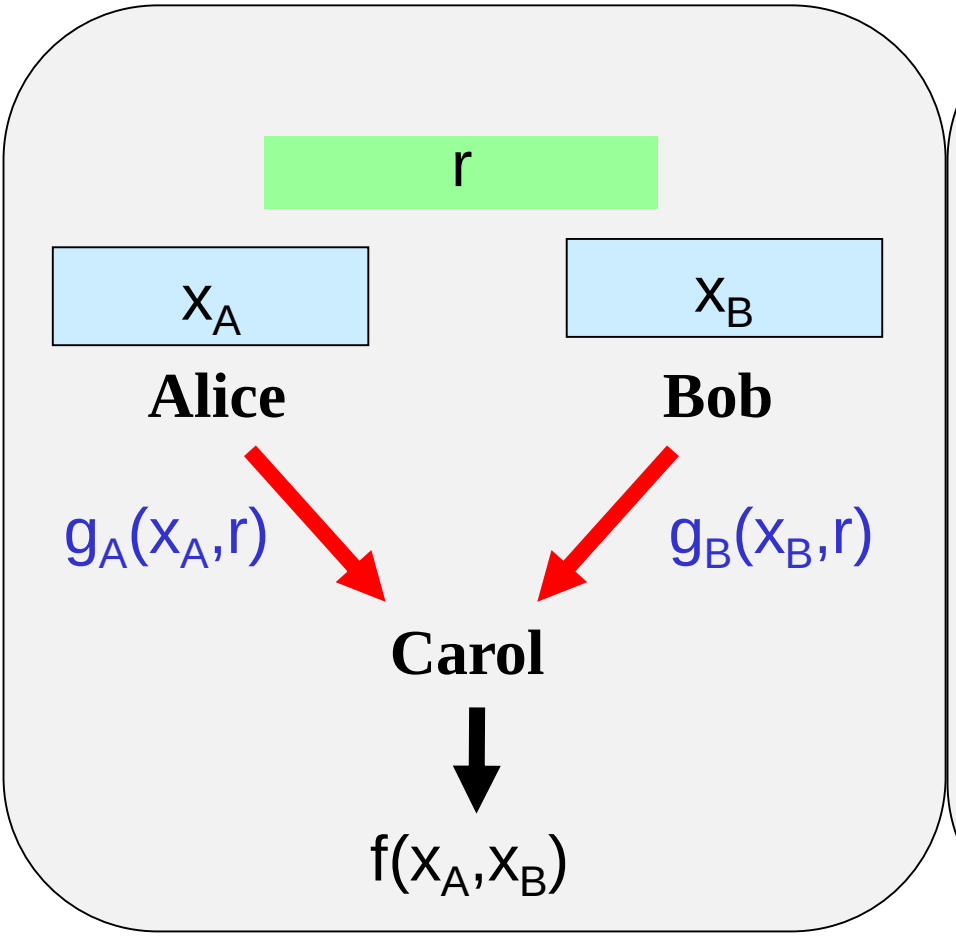


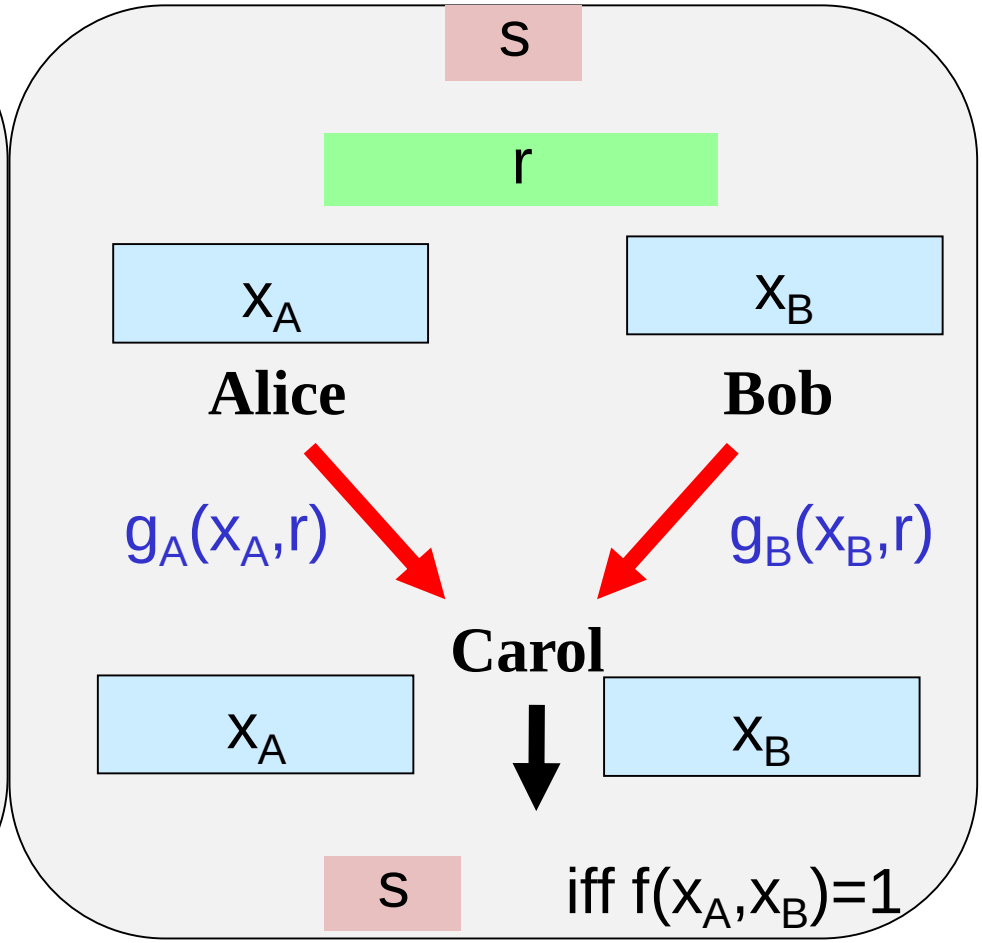
PSM vs CDS

$3n \leq \mathrm{PSM}(f) \leq 2^{n/2}$

[FKN94,A-HMS17] [BIKK14]

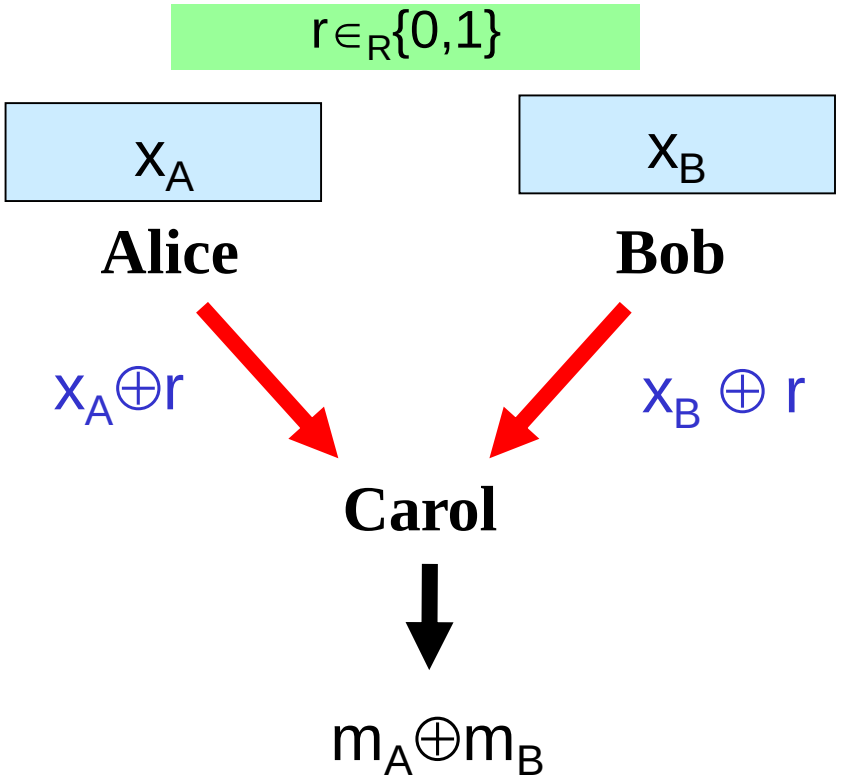
$n \leq \mathrm{CDS}(f) \leq 2^{\tilde{O}(\sqrt{n})}$





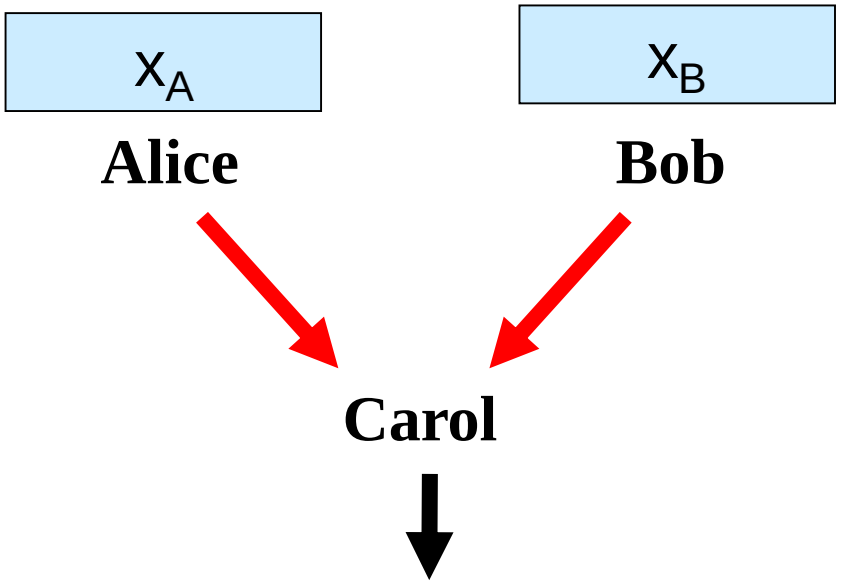
Example: XOR

- $f(x_A,x_B) = x_A \oplus x_B$ $(x_A,x_B \in \{0,1\})$



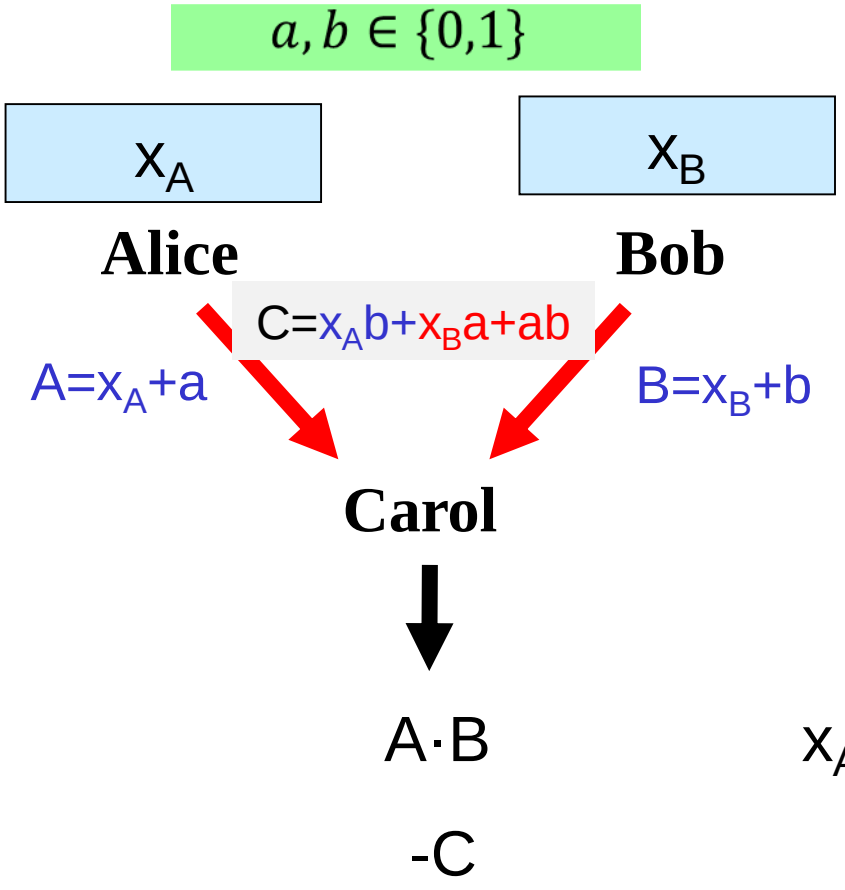
AND?

- $f(x_A,x_B) = x_A \cdot x_B$



AND: Intermediate protocol

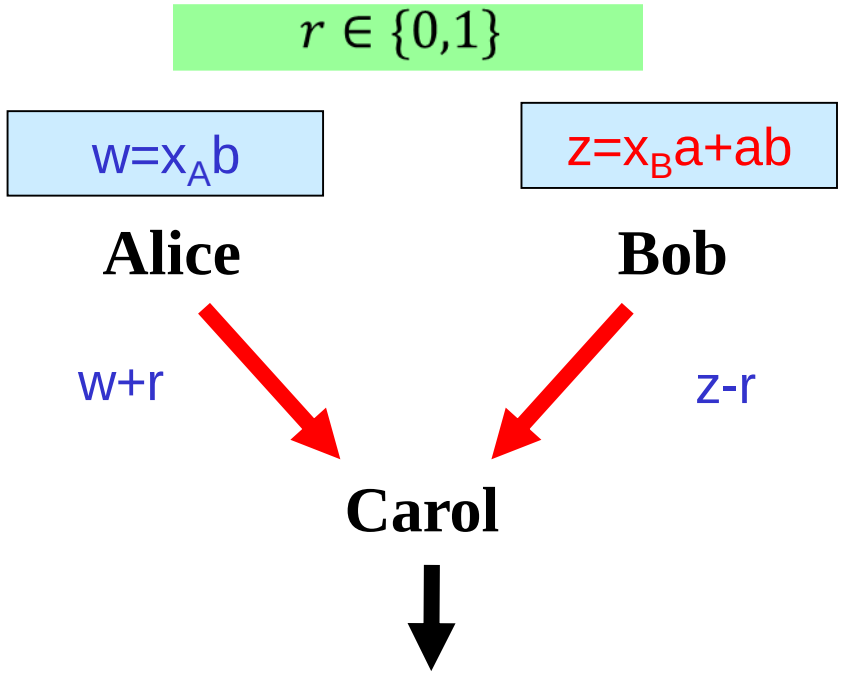
- $f(x_A,x_B) = x_A \cdot x_B$



$x_Ax_B+x_Ab+x_Ba+ab$

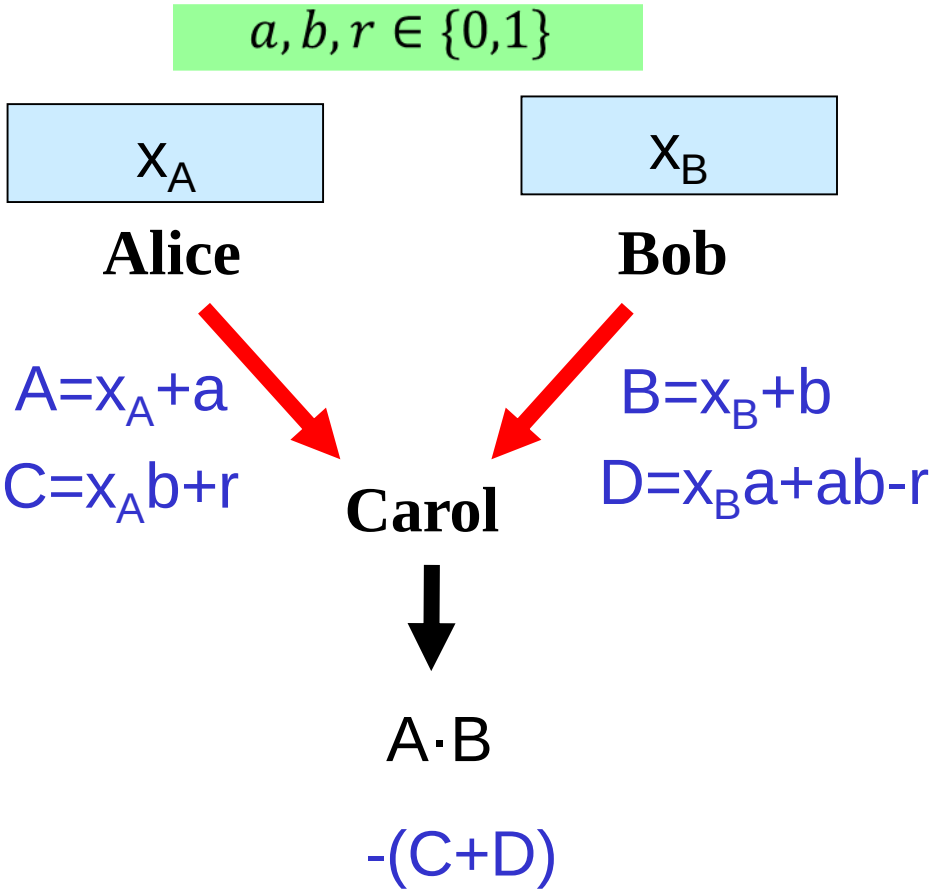
AND: Second Step

- $f(x_A,x_B) = x_A \cdot x_B$



AND!

- $f(x_A,x_B) = x_A \cdot x_B$



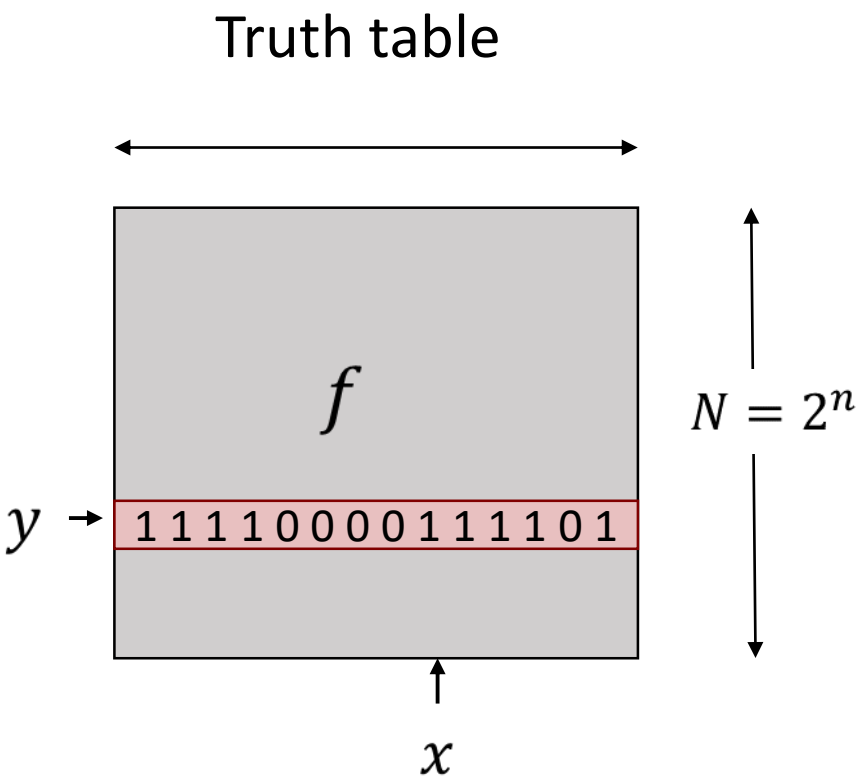
Introspection

- Gradually constructed the protocol
- Intermediate construction didn’t satisfy syntax
  but preserved information
- Used Simple Maneuvers

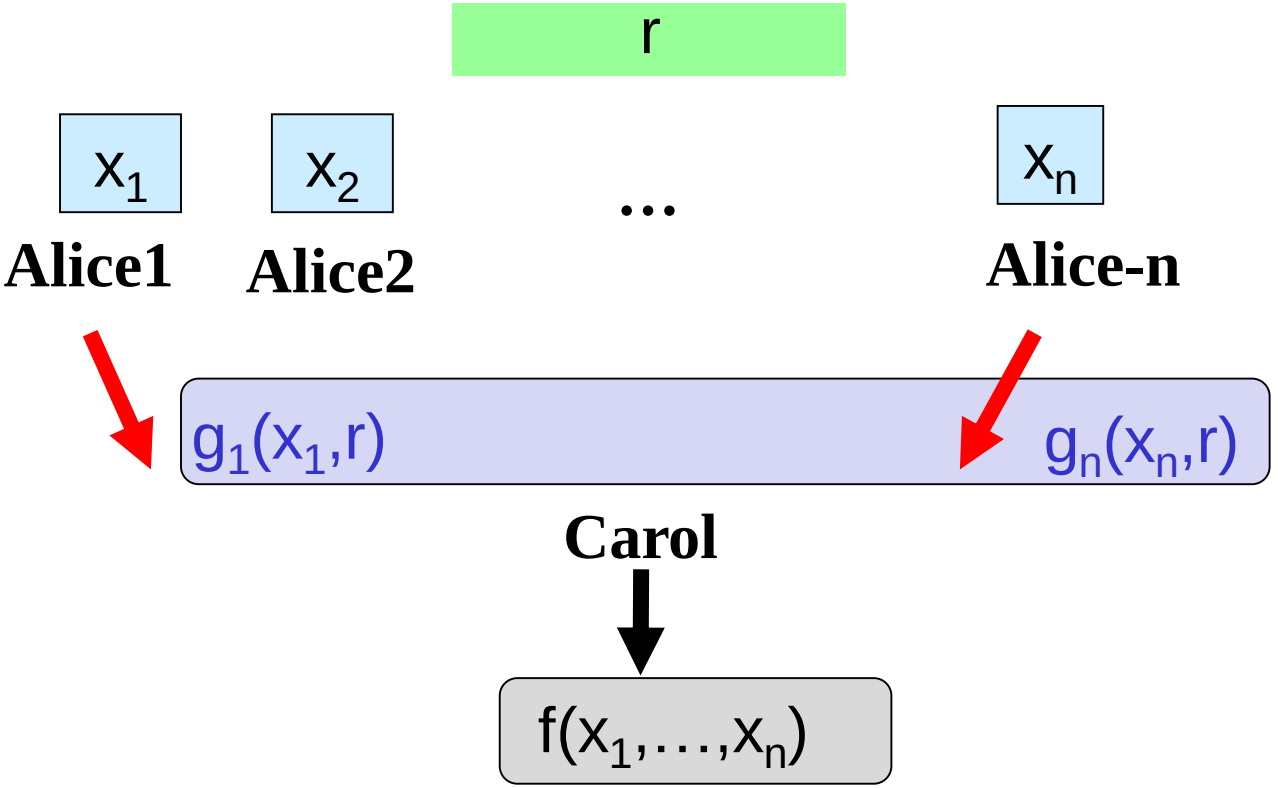
Test your intuition:

- Q:Combine PSM(f), PSM(g) to PSM(f AND g)?

- Ex: PSM(f) based on truth-table randomization

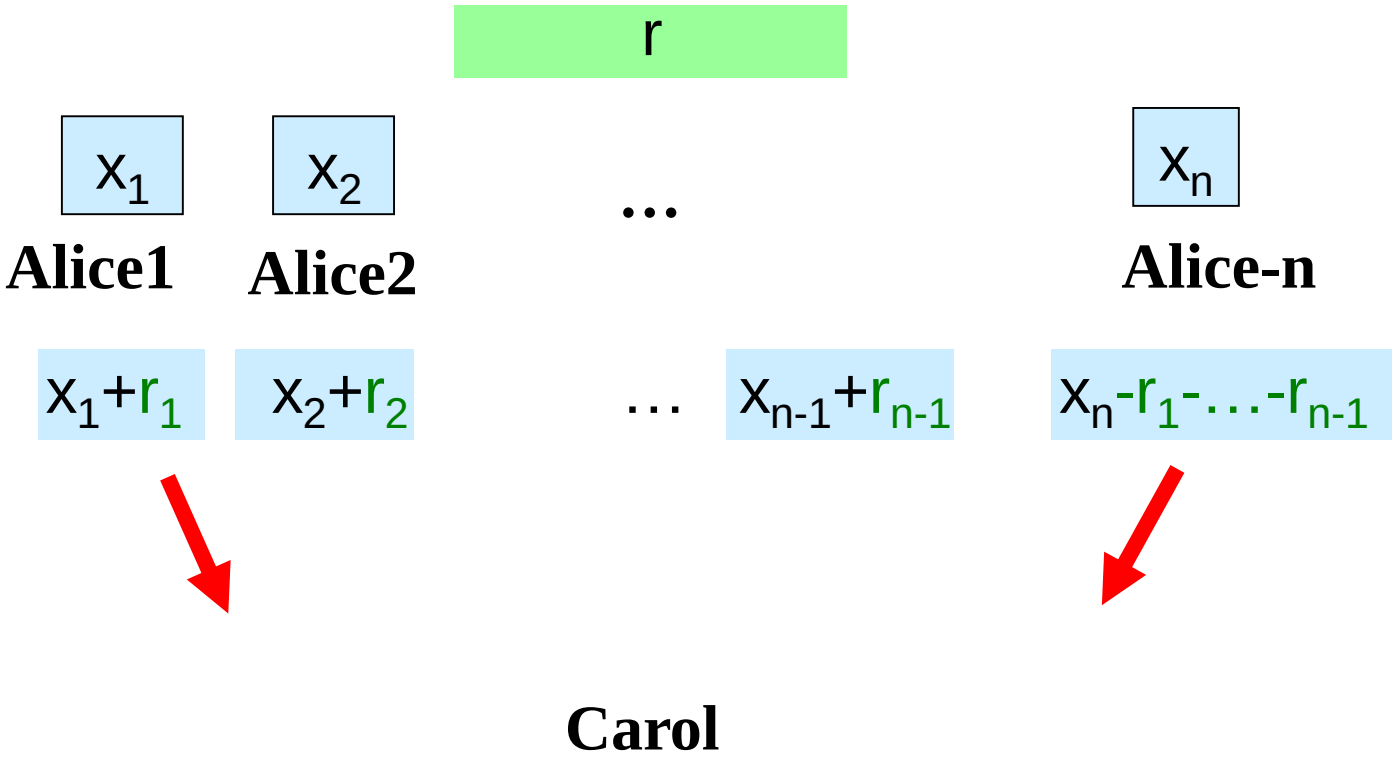


Multiparty Version [IK98]



Example: iterated group product

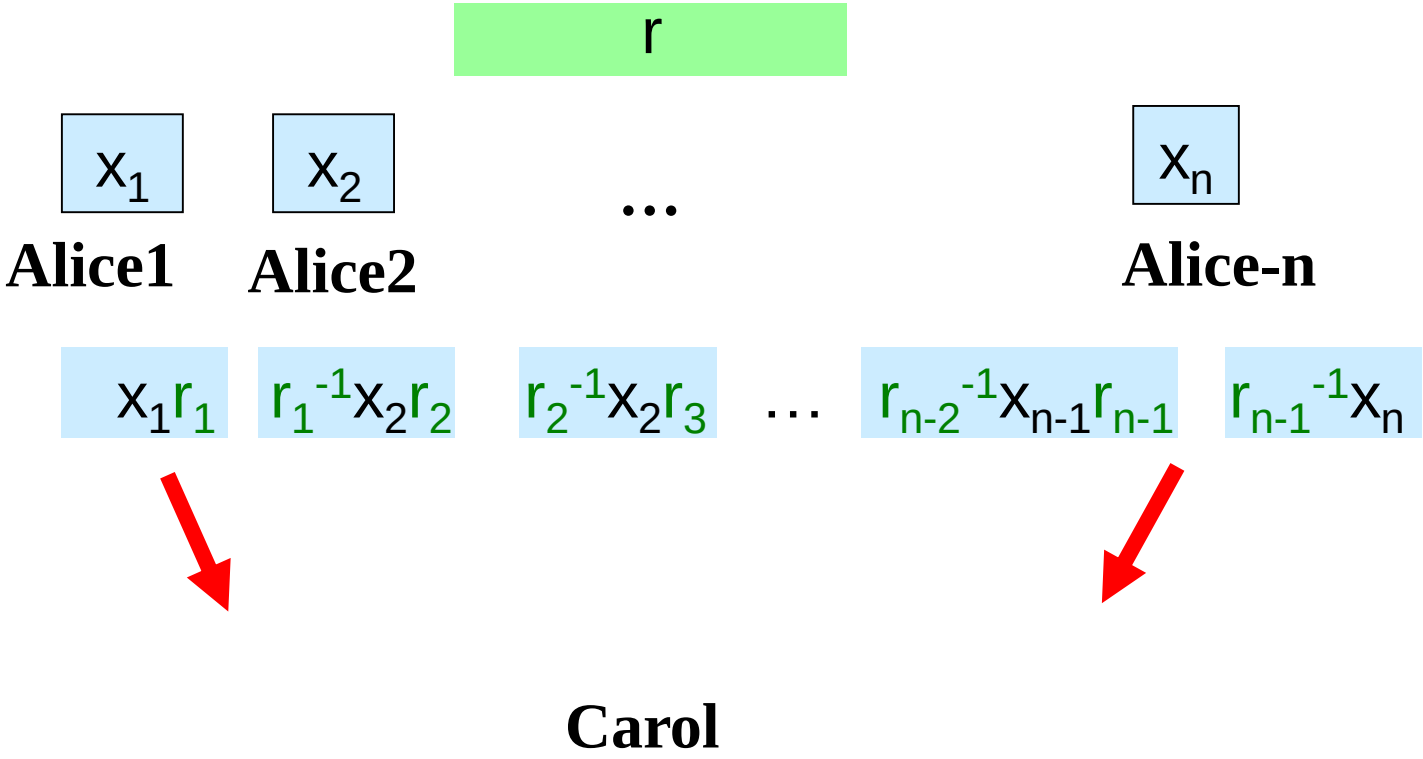
Abelian Group: $f(x_1,\ldots,x_n)=x_1+x_2+\ldots+x_n$



Example: iterated group product

Non-abelian Group: $f(x_1,\ldots,x_n)=x_1x_2\ldots x_n$

[Kilian 88]



Handling General Functions

Theorem [Barrington 86]

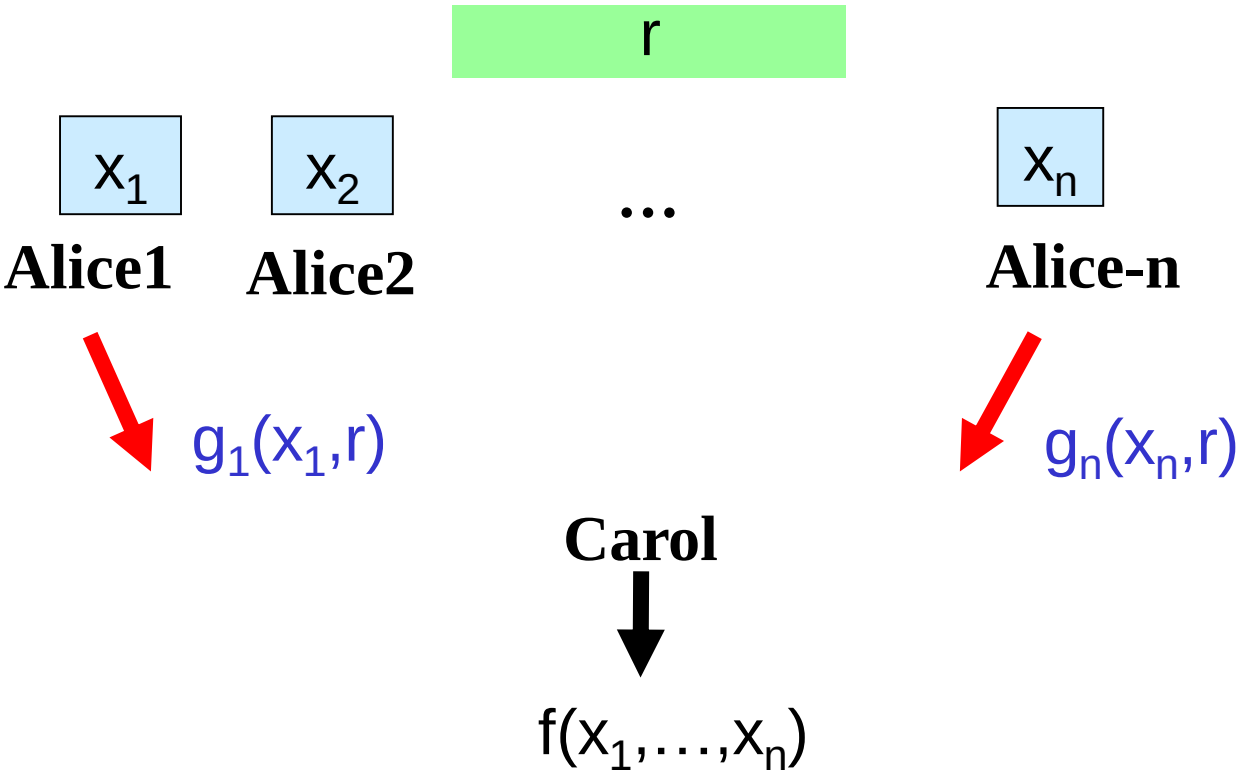
Every boolean $f \in NC^1$ can be written as iterated group product

Corollary

Every $f \in NC^1$ has multiparty PSM with poly-communication

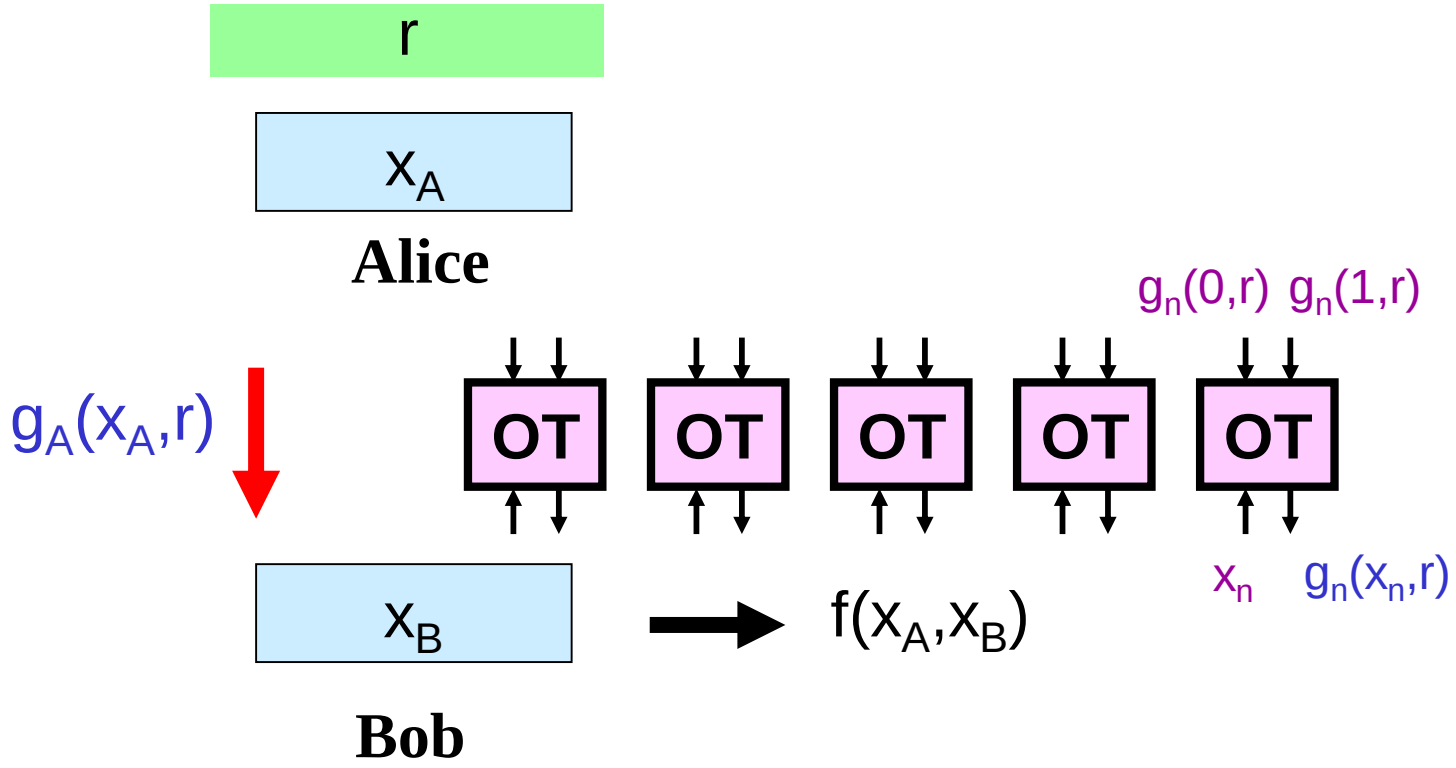
From Multiparty PSM to OT-based MPC

Application: Basing SFE on OT [Yao,Kilian 88, ...]

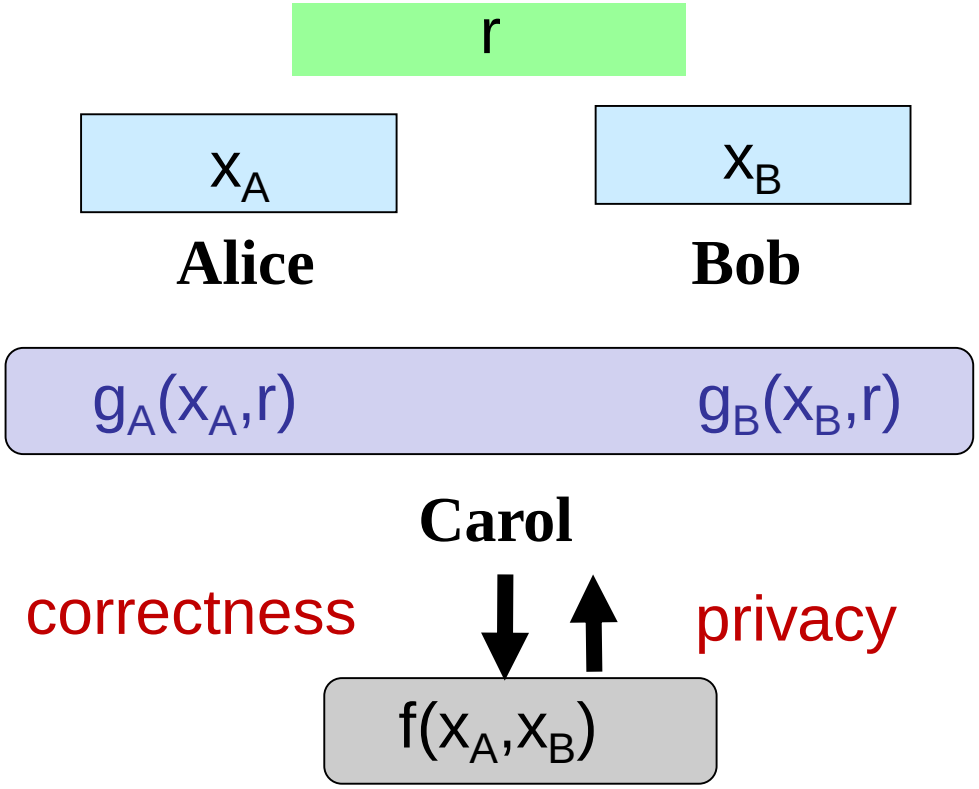


From Multiparty PSM to OT-based MPC

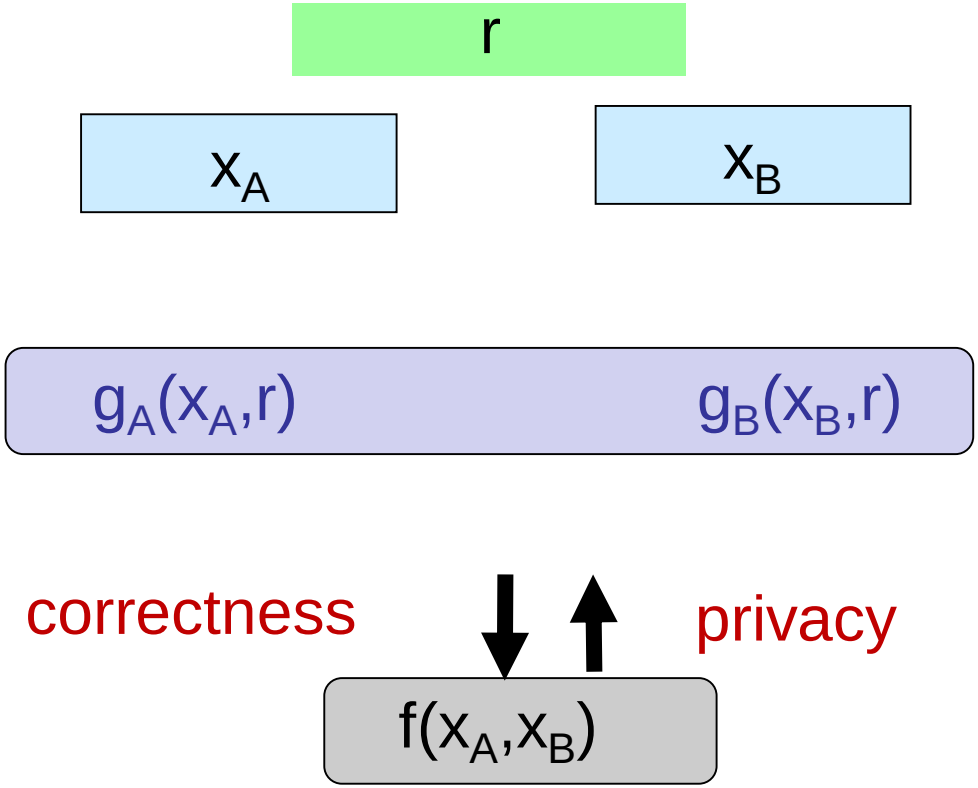
Application: Basing SFE on OT [Yao,Kilian 88, ...]



Can we leverage these ideas in more general settings?



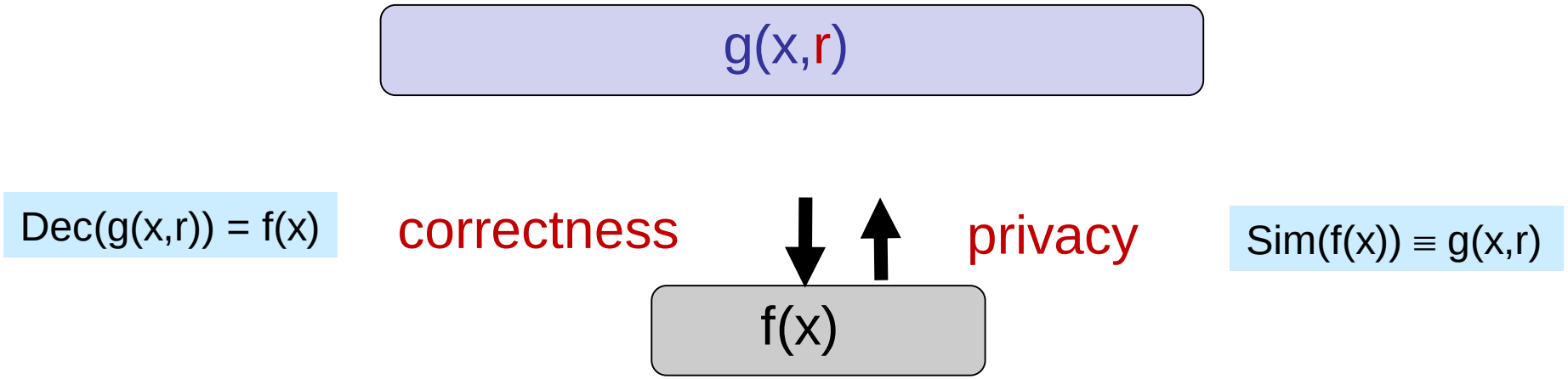
Can we leverage these ideas in more general settings?



Randomized Encoding of Functions

[Ish-Kus00, A-Ish-Kus04]

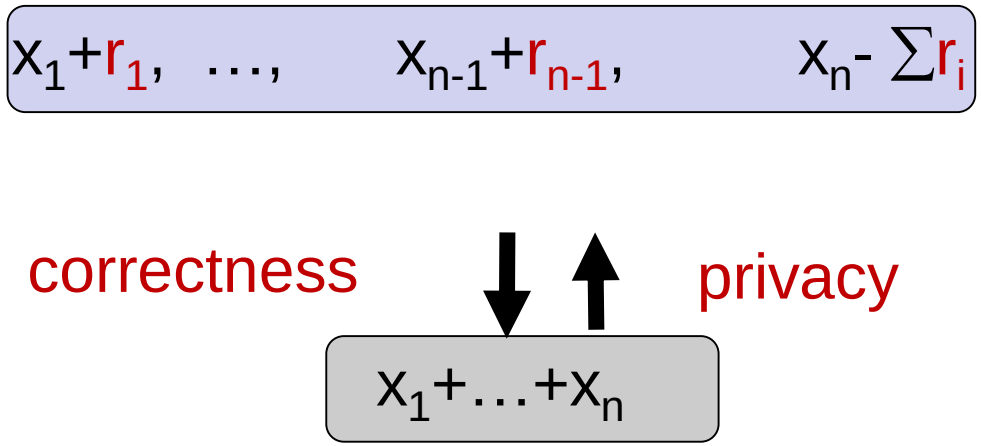
- *g* is a “randomized encoding” of *f*
  - Nontrivial relaxation of computing *f*



Randomized Encoding of Functions

[Ish-Kus00, A-Ish-Kus04]

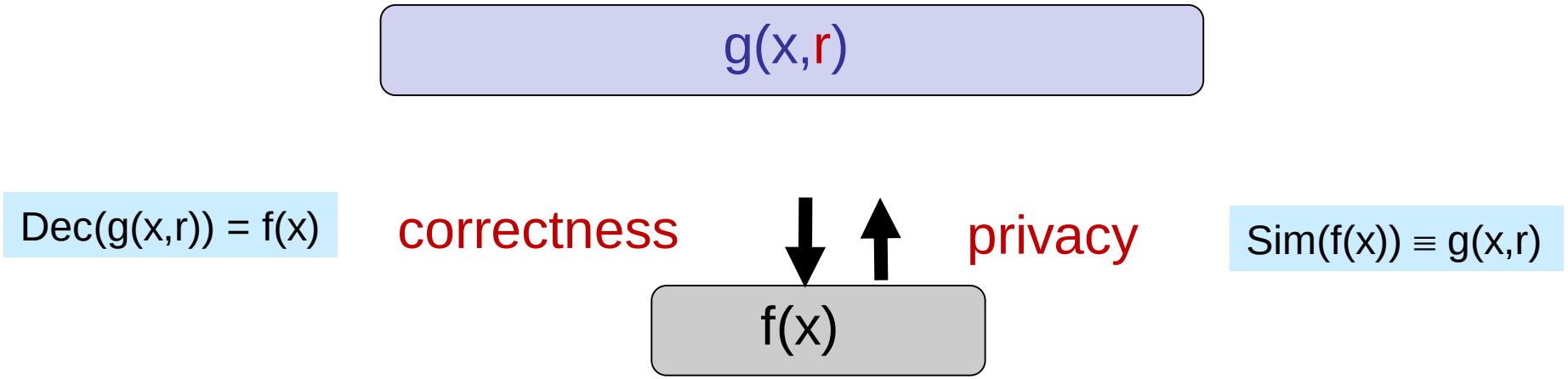
- *g* is a “randomized encoding” of *f*
  - Nontrivial relaxation of computing *f*



Randomized Encoding of Functions

[Ish-Kus00, A-Ish-Kus04]

- Securely computing g => securely computing f
- If g is realizable in MPC-model so is f



Randomized Encoding of Functions

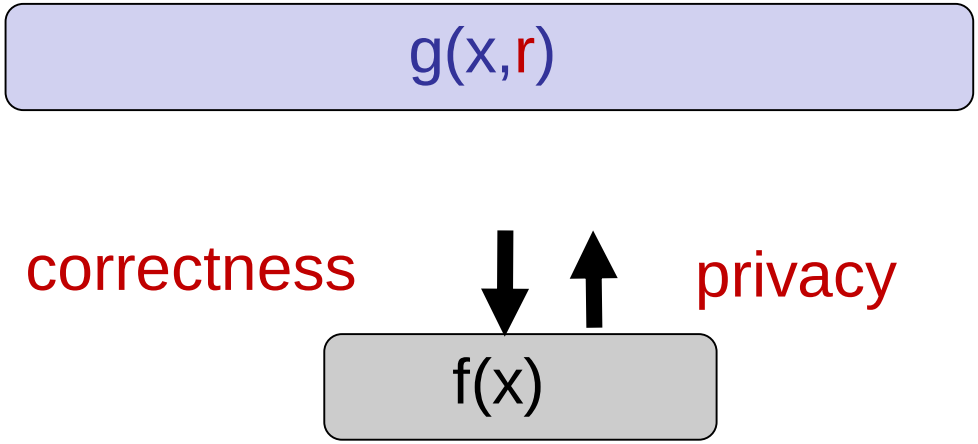
[Ish-Kus00, A-Ish-Kus04]

General paradigm:

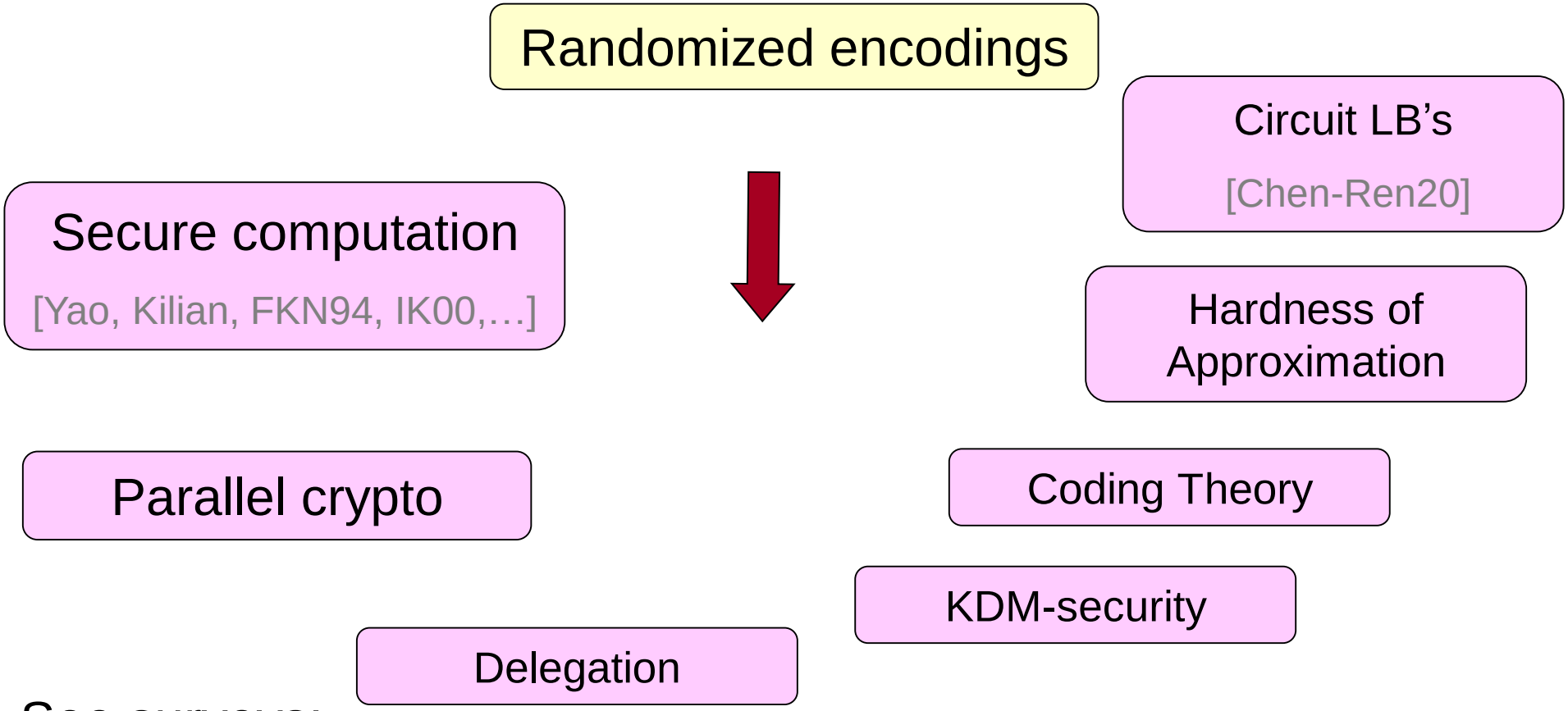
- *g* should be “simpler” than *f*

  (meaning of “simpler” determined by application)

- *g* can be used as a substitute for *f*



Applications at a Glance

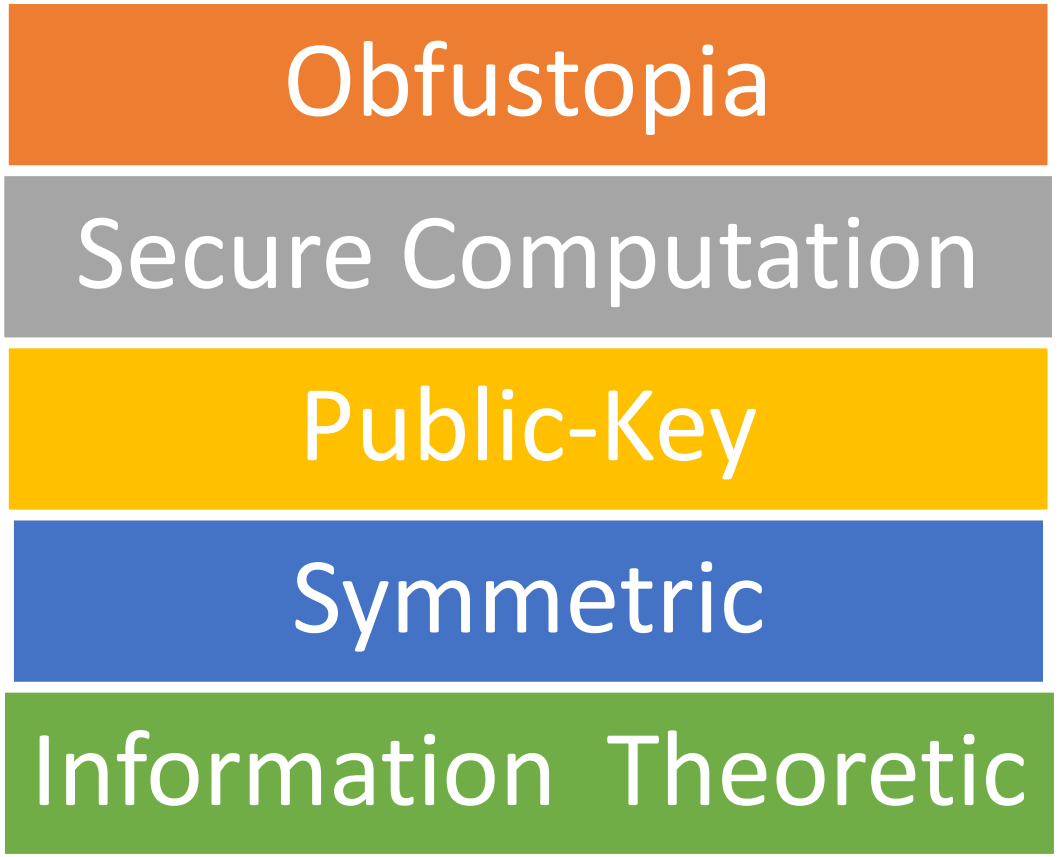


See surveys:

Ishai: Randomization Techniques for Secure Computation

Applebaum: Garbled Circuits as Randomized Encodings of Functions: a Primer

Randomized Encoding in Cryptography



Compact Functional Encryption/ Obfuscation

Reusable Garbled Circuit

Functional Encryption

Garbled Circuits

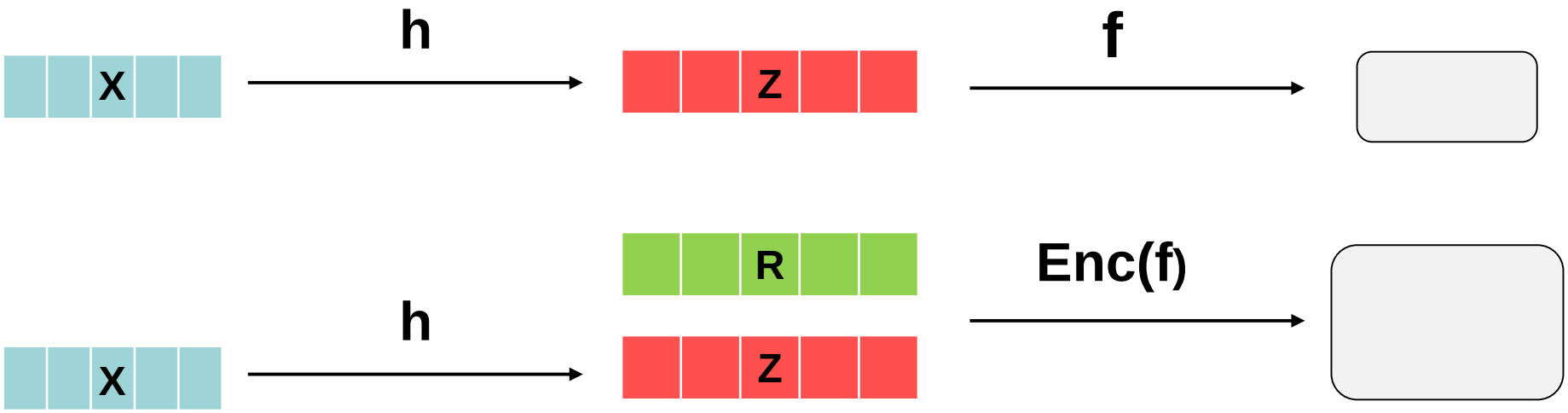
PSM

Useful Properties

Composition: Enc(Enc(f)) is an encoding of f.

Concatenation: (Enc(f_1), Enc(f_2)) is an encoding of f=(f_1,f_2).

Substitution: Enc(f)°h encodes f°h



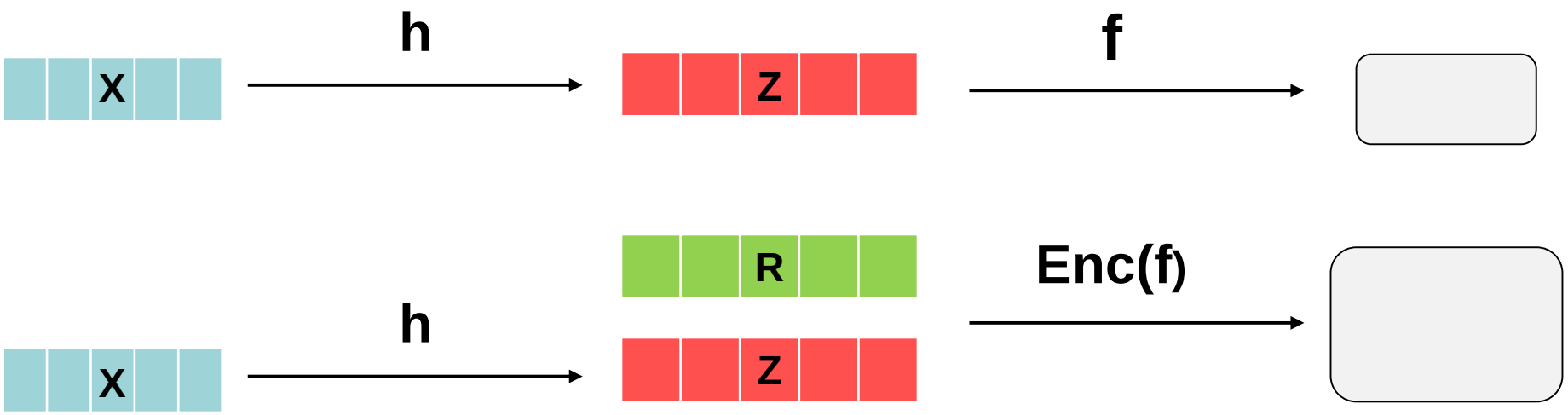
Glue REs of simple Functions to General Formulas (or even to Circuits assuming PRG)

[AIK11] framework

Composition: Enc(Enc(f)) is an encoding of f.

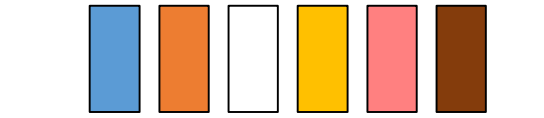
Concatenation: (Enc(f_1), Enc(f_2)) is an encoding of $f=(f_1,f_2)$.

Substitution: Enc(f)$\circ$h encodes f$\circ$h



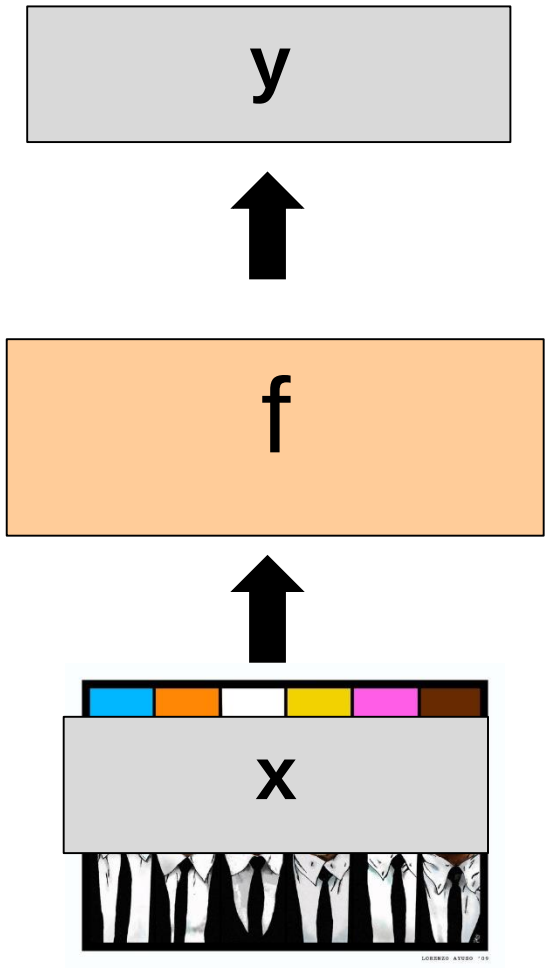
Back to Constant-Round MPC

Randomizing Polynomials [IK00]



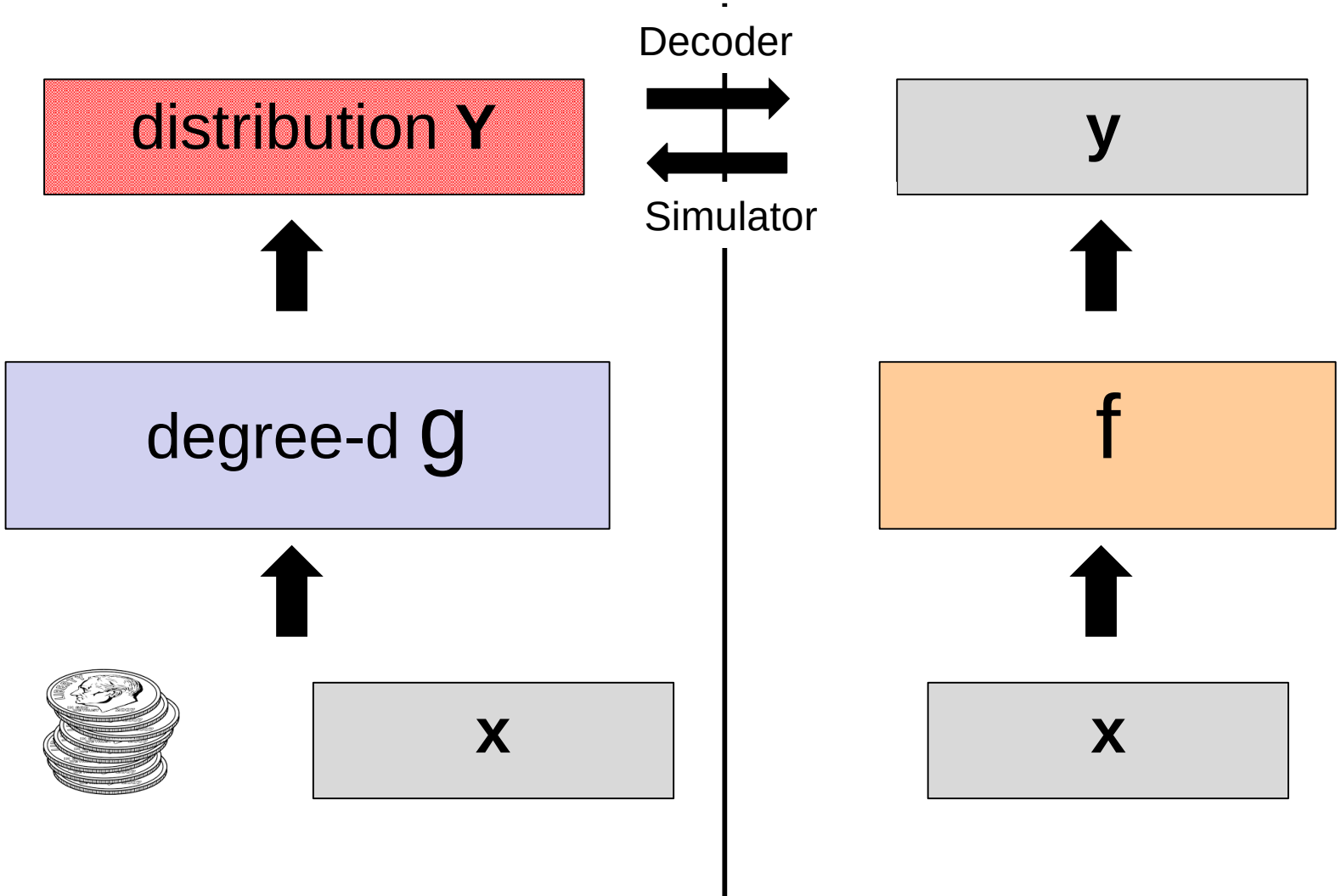
f

Randomizing Polynomials [IK00]



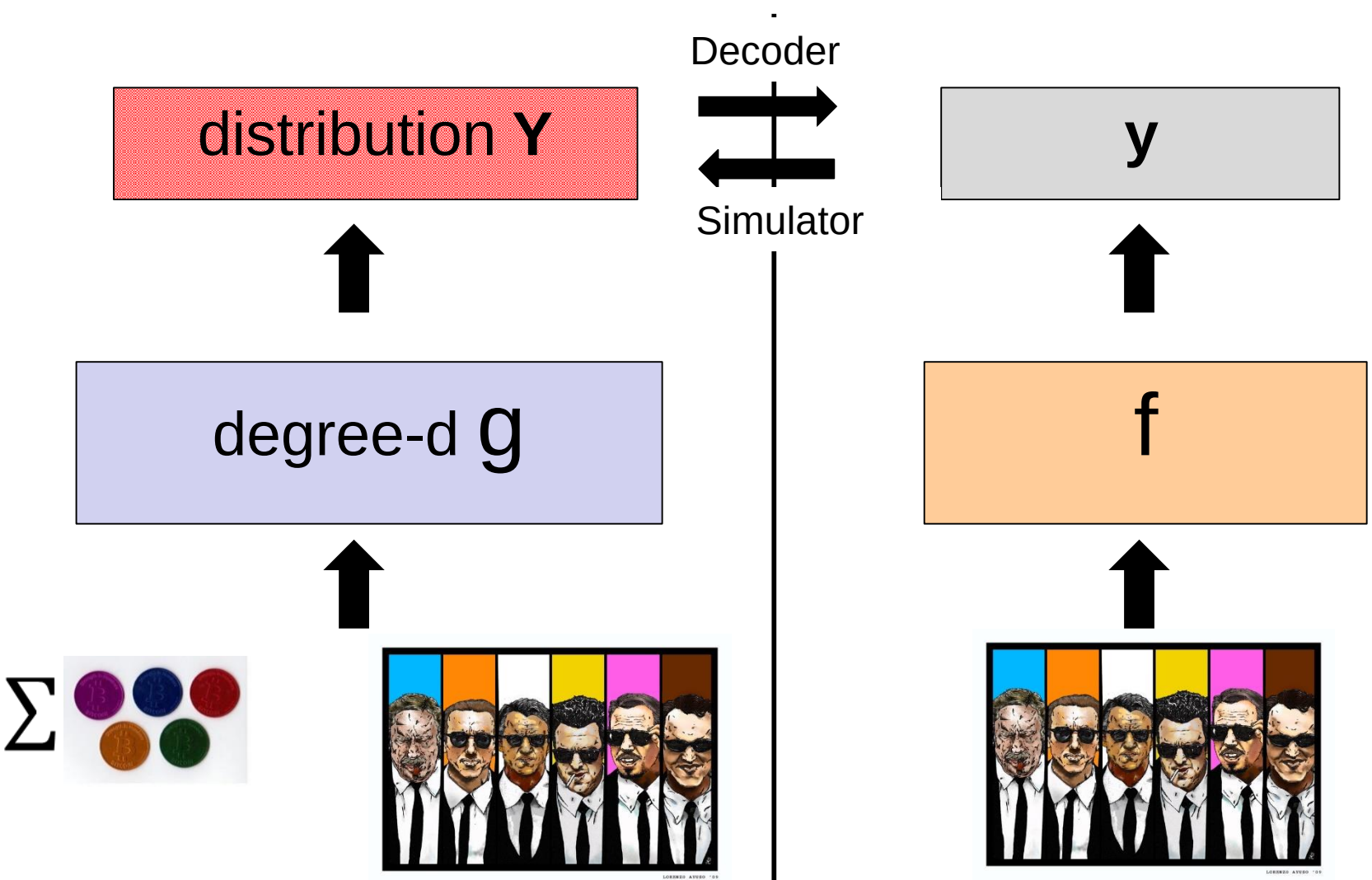
Randomizing Polynomials [IK00]

MPC for g $\Rightarrow$ MPC for f



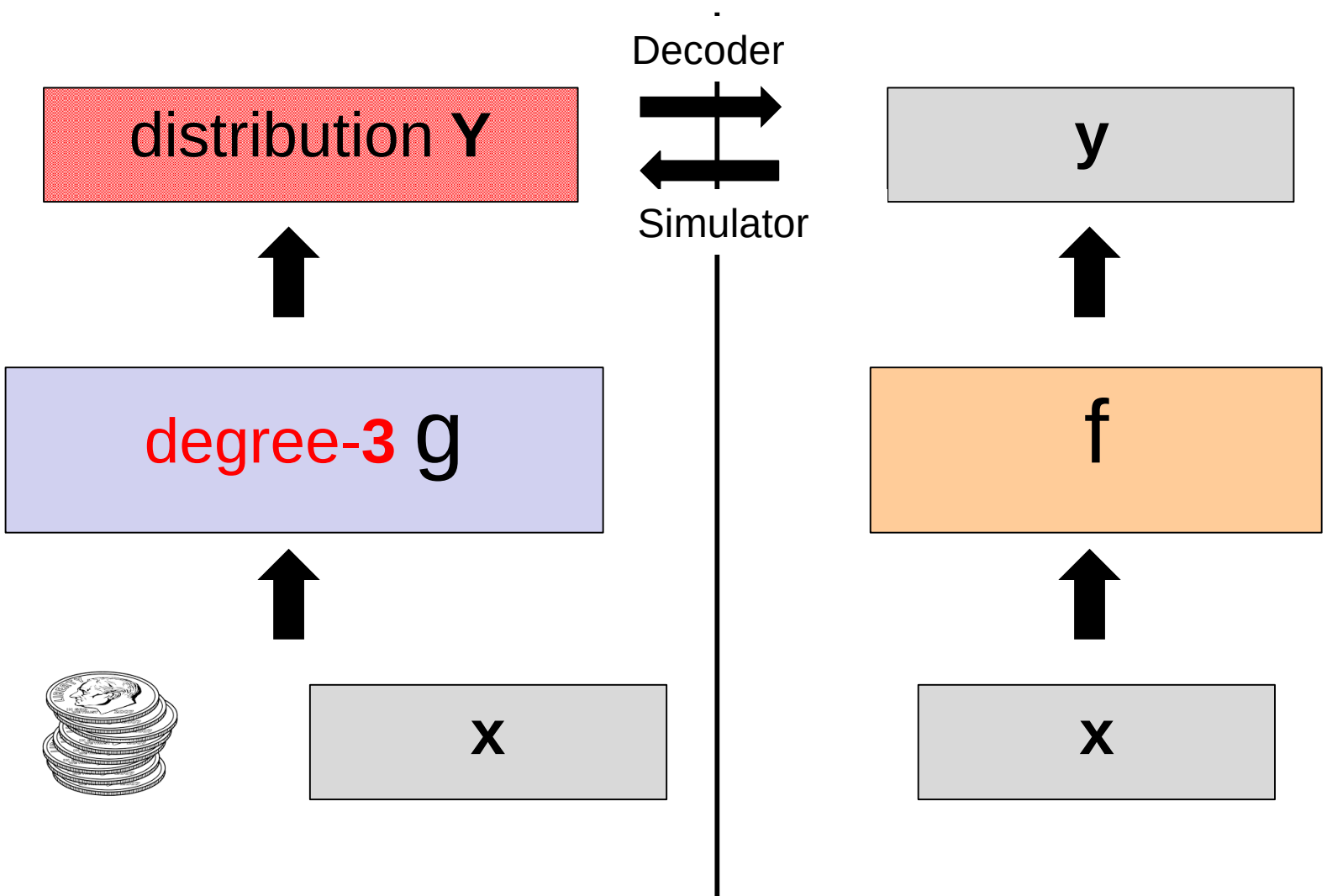
Randomizing Polynomials [IK00]

g has d-round protocol $\Rightarrow$ f has d-round protocol !



Randomizing Polynomials [IK00]

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



Perfect-MPC with Constant Round

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

- Efficient for NC1, log-space

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

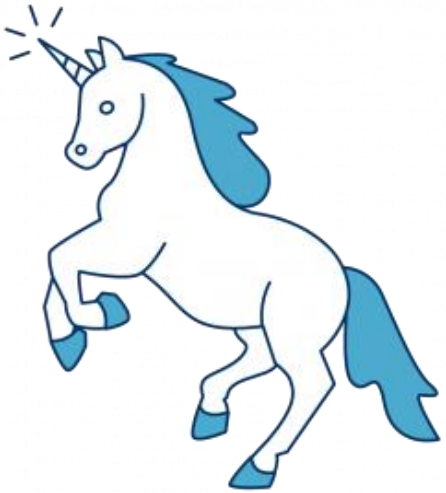
- Passive T<N/2: 3 rounds ≥?>1
- Active T<N/3: large const. ≥?> 2

Q: What's the optimal round complexity?

Problem:

For most functions,

NO degree-2 perfect RE's



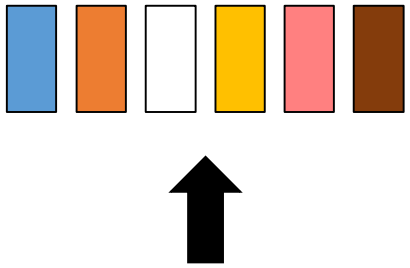
Sol: Compromise!

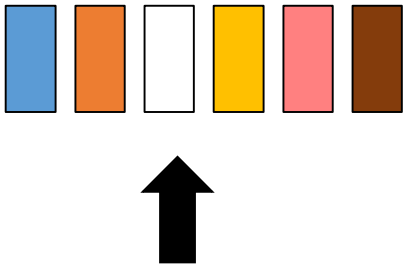
Aim for a weaker notion

Multiparty Randomized Encoding (MPRE)

[A-Bra-Tsa18]

Relaxed correctness: Each party has a decoder





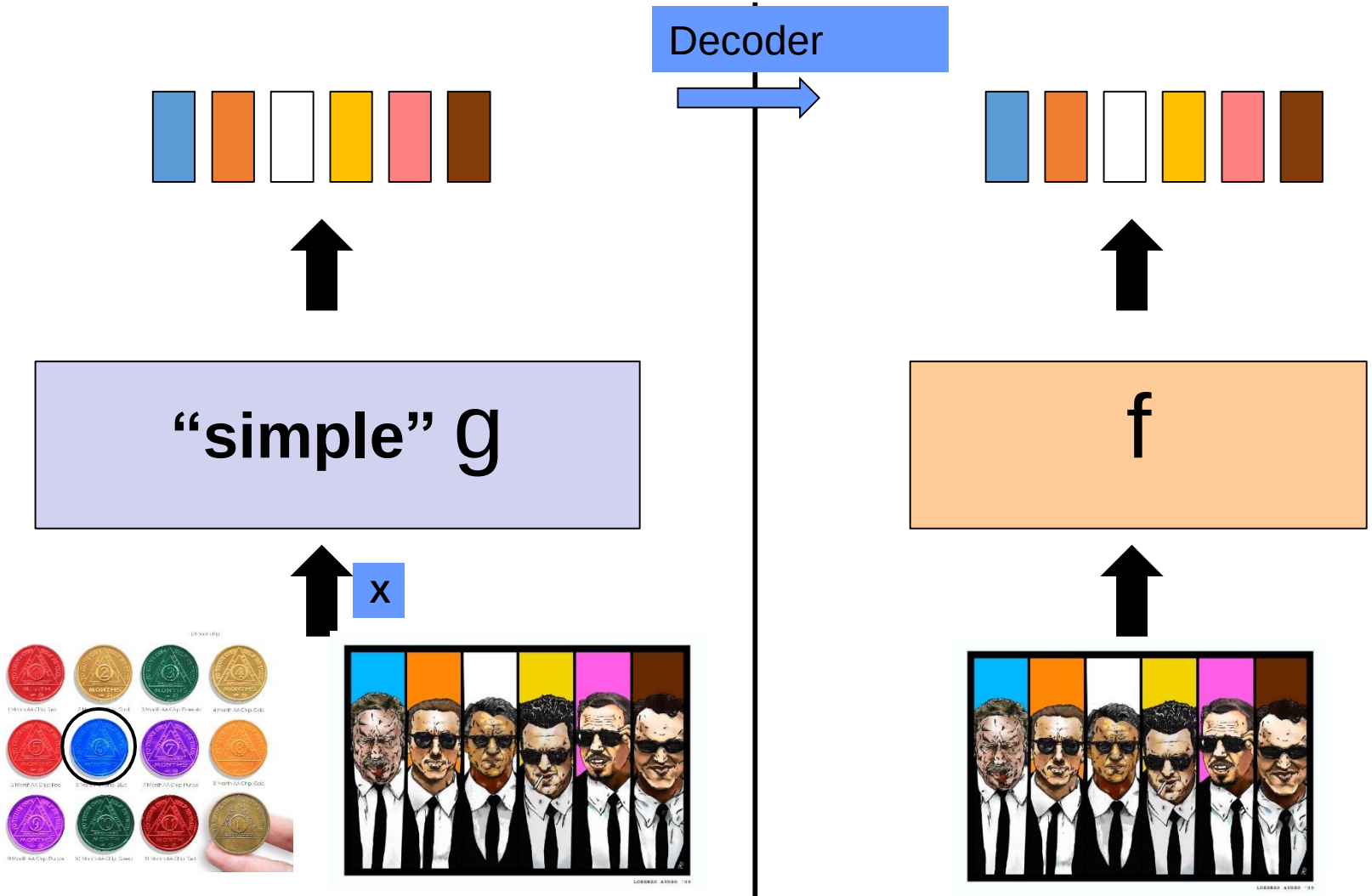
“simple” g

f

Multiparty Randomized Encoding (MPRE)

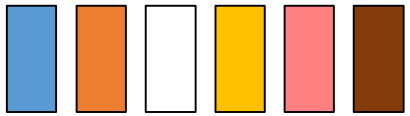
[A-Bra-Tsa18]

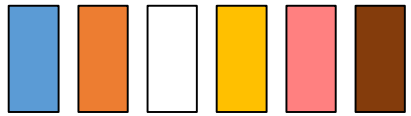
Relaxed correctness: Each party has a decoder



Multiparty Randomized Encoding (MPRE)

Relaxed privacy: Every minority has a simulator



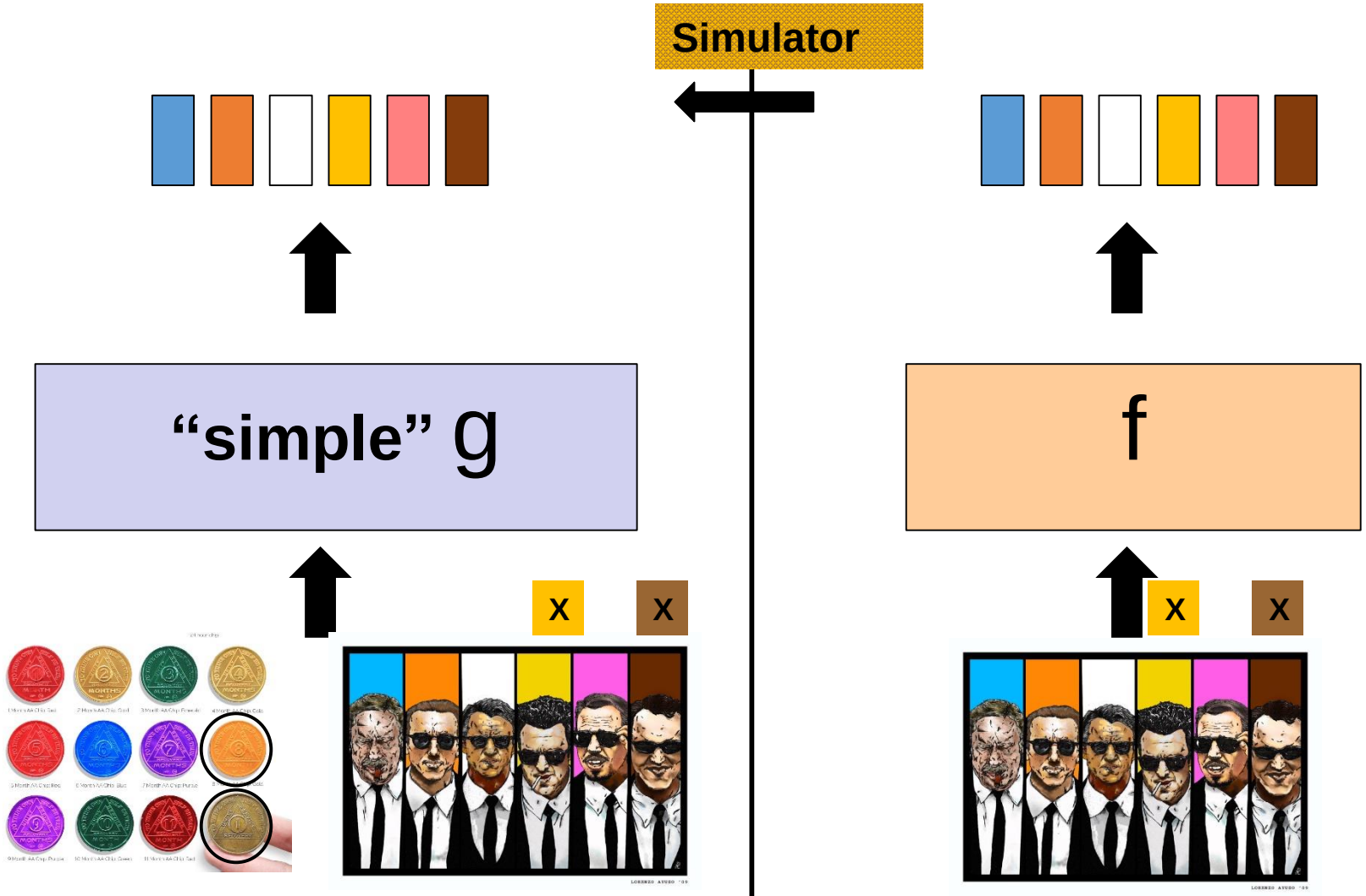


“simple” g

f

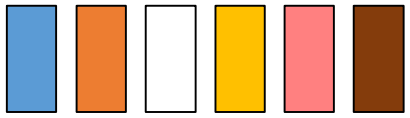
Multiparty Randomized Encoding (MPRE)

Relaxed privacy: Every minority has a simulator

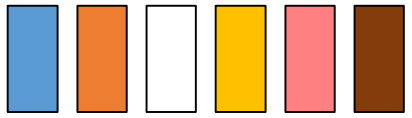


MPRE **relaxes** Randomized Encoding

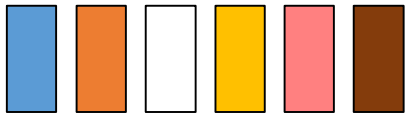
- Encodes functionality
- RE is a special case of MPRE
- Protocol for g ⇒ Protocol for f



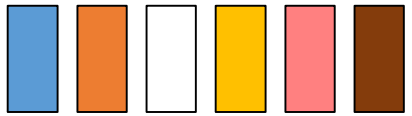
“simple” g



f



Degree-2 ?



f

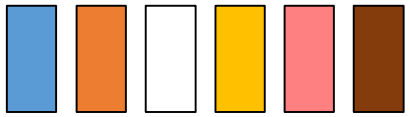
Thm [A-Bra-Tsa18]: every f has MPRE of "effective" deg-2

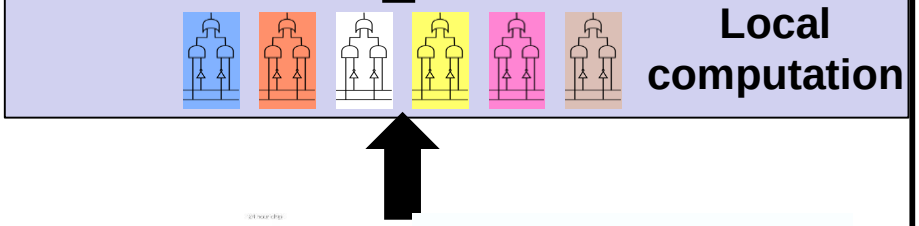
- Efficient for log-space
- Efficient computational-MPRE for general circuits

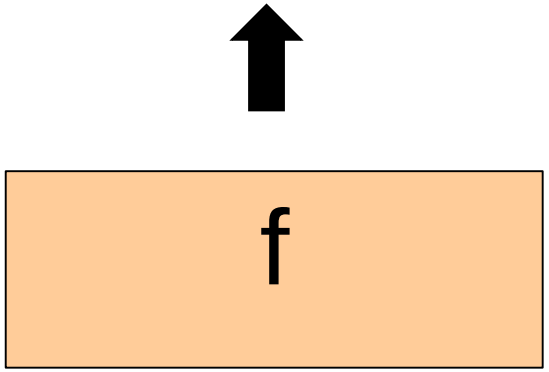
⇒ 2-round passively-secure honest-majority protocol

Also [Gar-Ish-Sri-18]

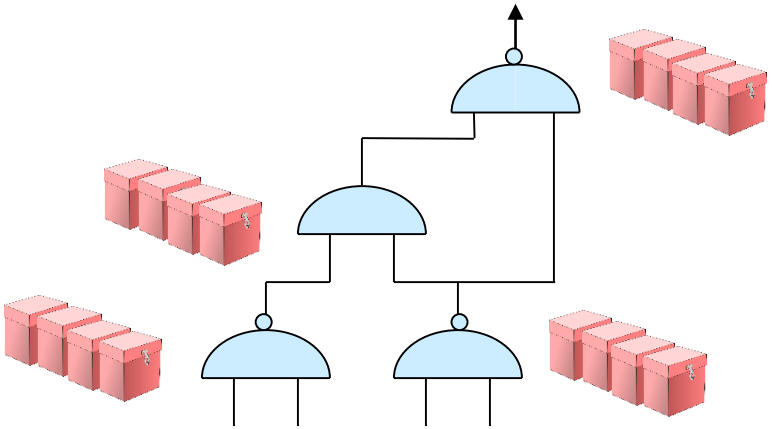
inspired by [GS-16-17]







Back to [IK00]:
Degree-3 Randomizing Polynomials from Information-Theoretic Garbled Circuits

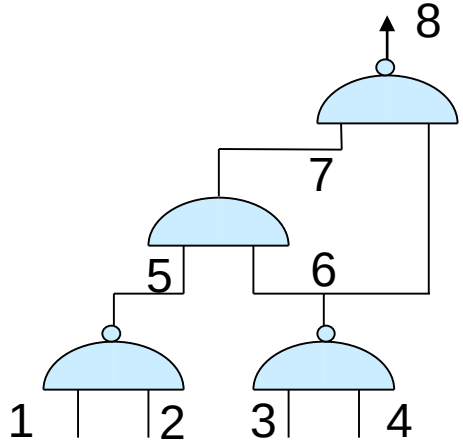


g

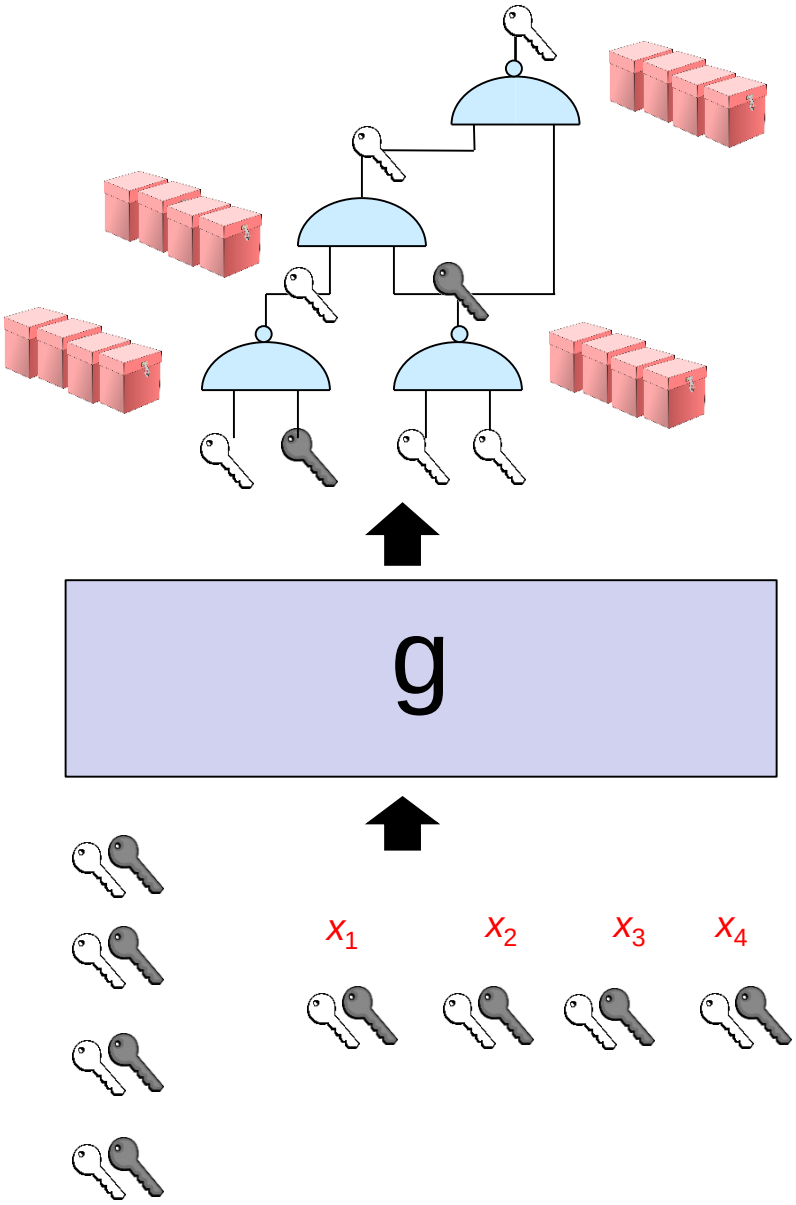
5 6 7 8

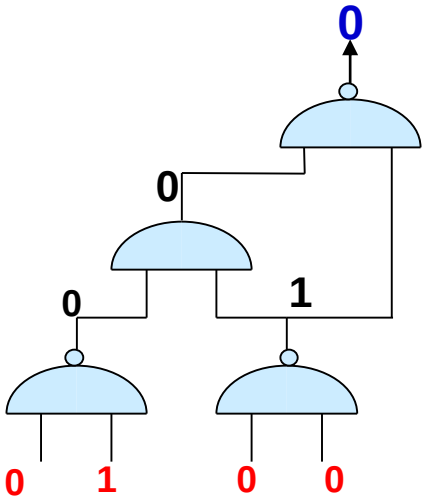
x_1 x_2 x_3 x_4

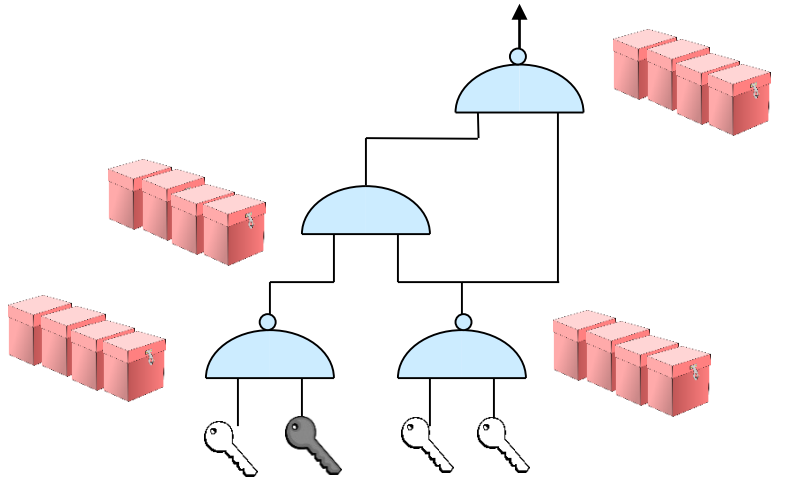
1 2 3 4

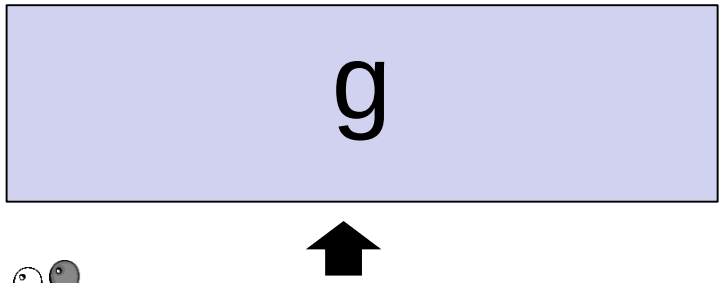


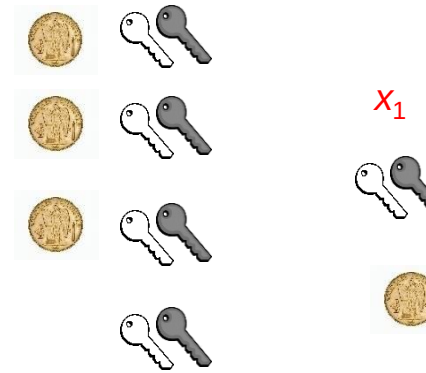
x_1 x_2 x_3 x_4

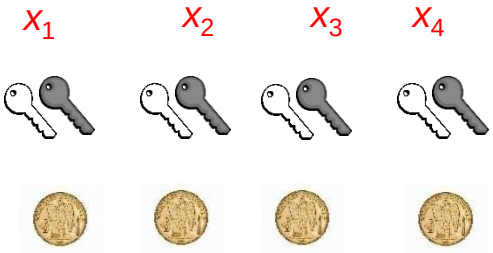

g
x_1
x_2
x_3
x_4


0
0
0
1
0
1
0
0



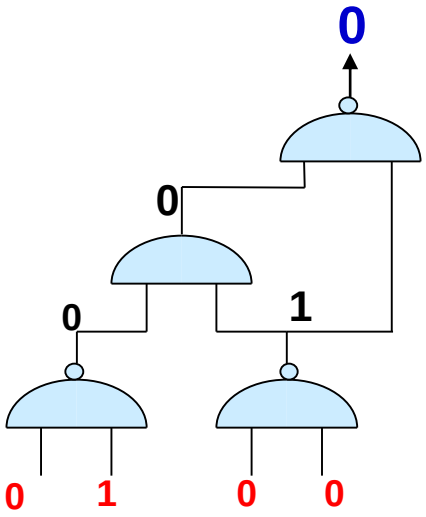






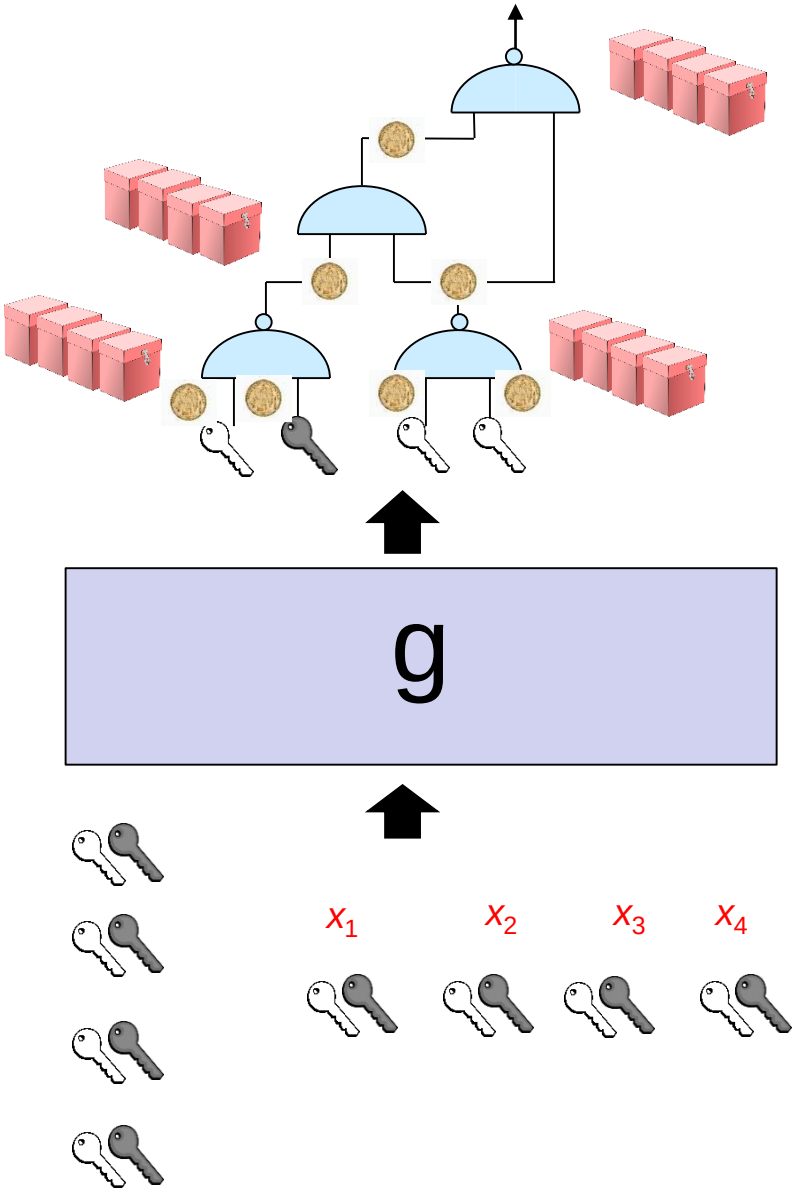
Correctness

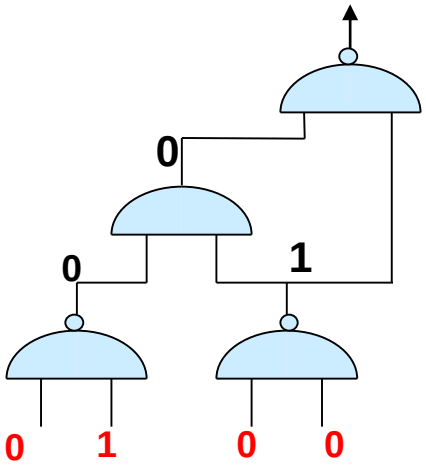
Privacy?



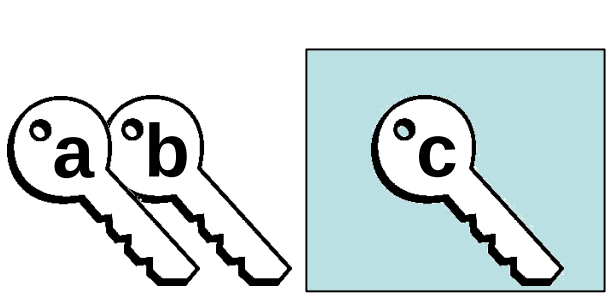
Overall degree 3

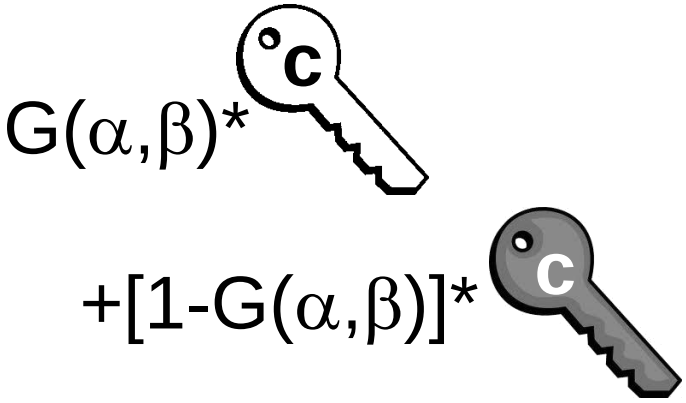
- deg(gate)+1

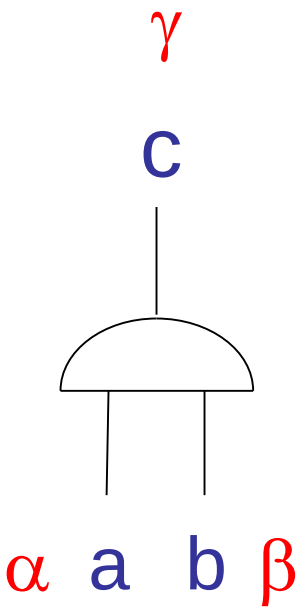




Degree of Garbled Gate







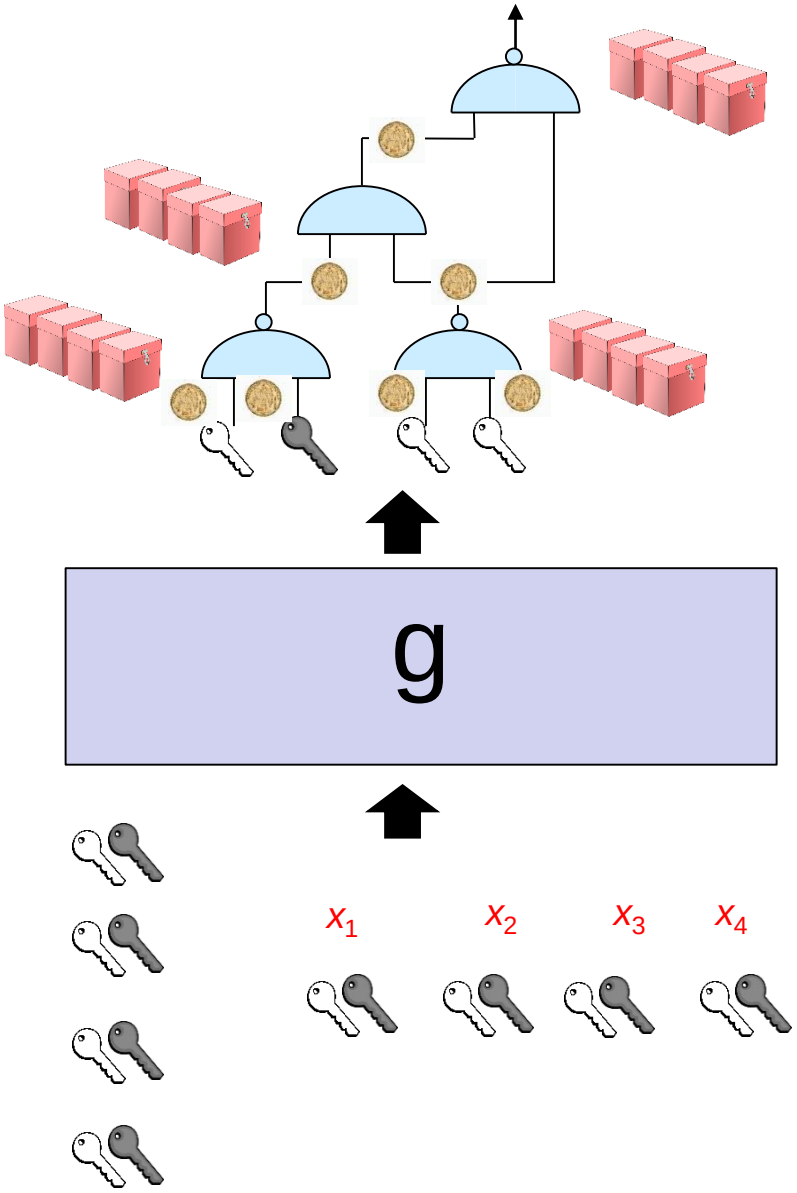
per gate:

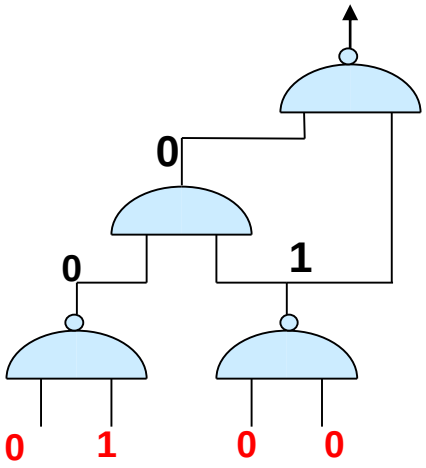
- 4 ciphertexts
- deg-3=deg(gate)+1

Overall degree 3

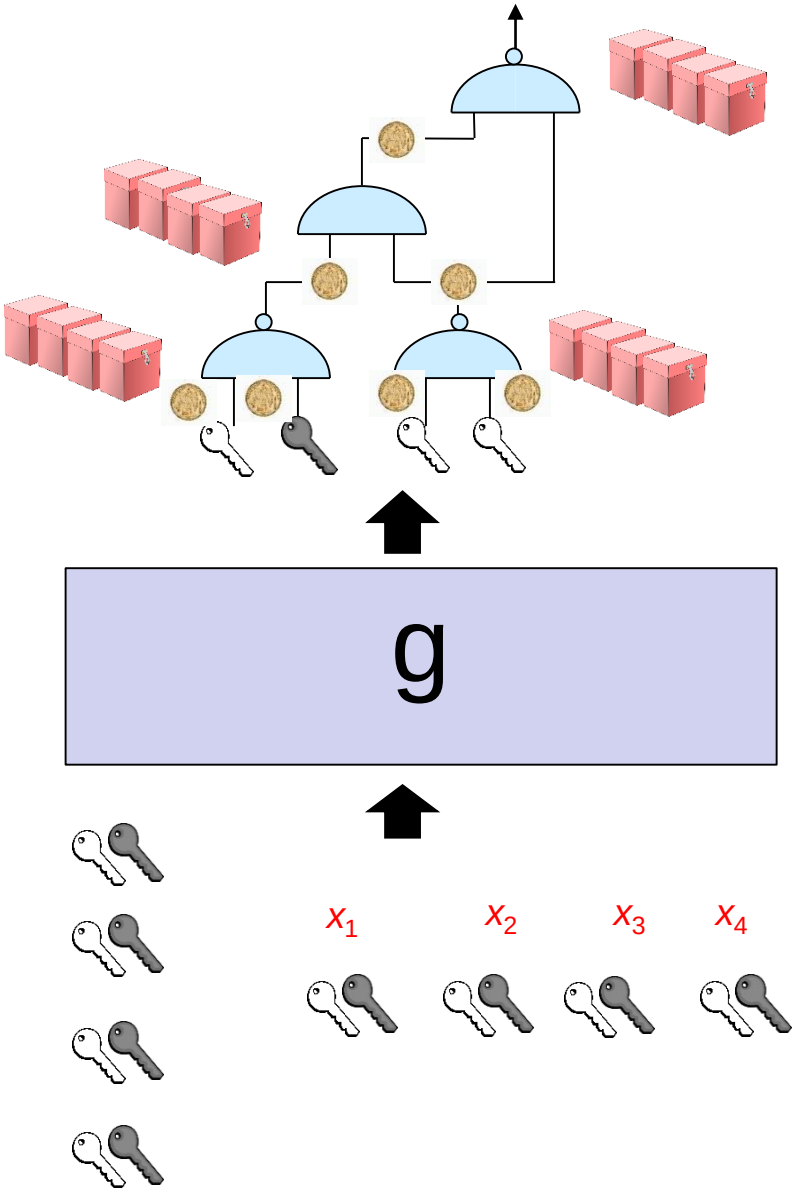
- deg(gate)+1

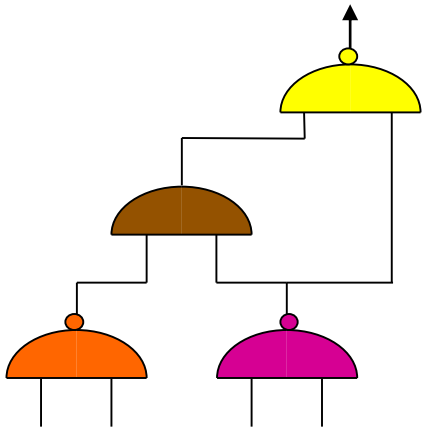
Can we reduce the degree to 2?



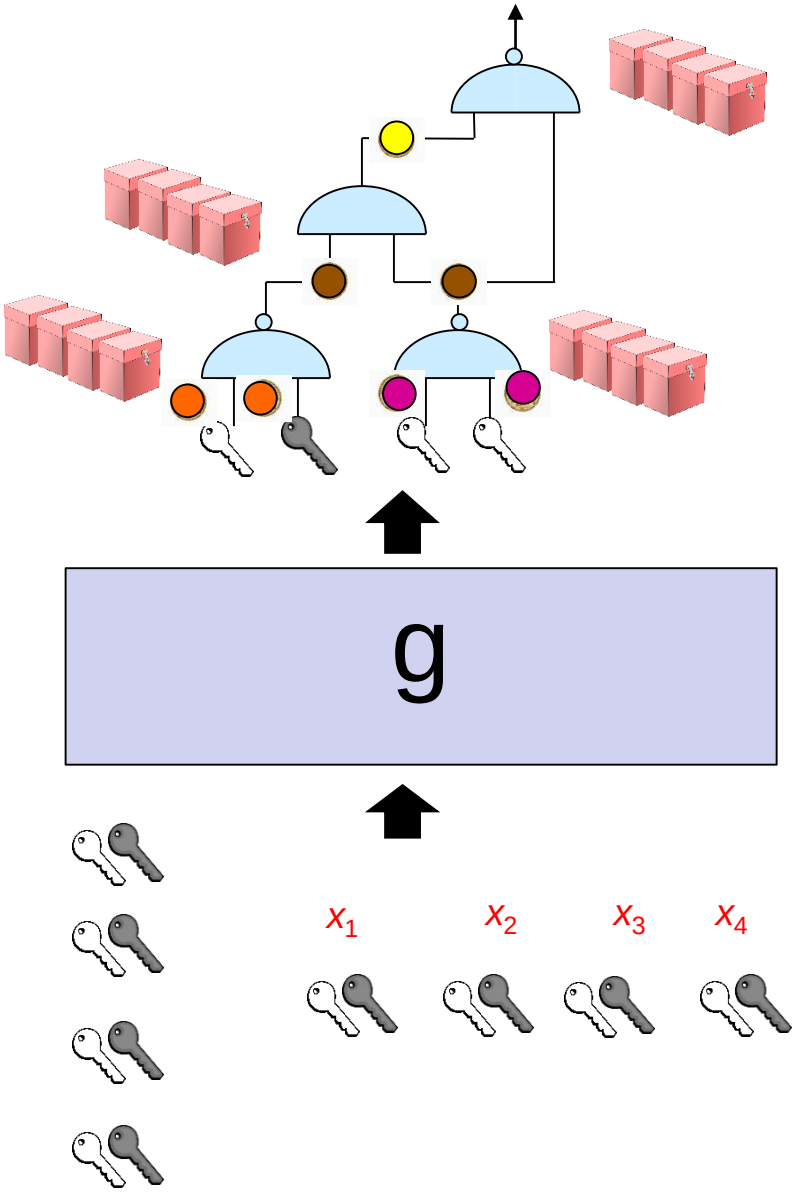


What if…?



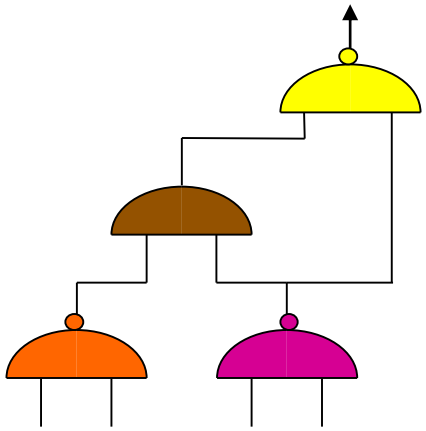


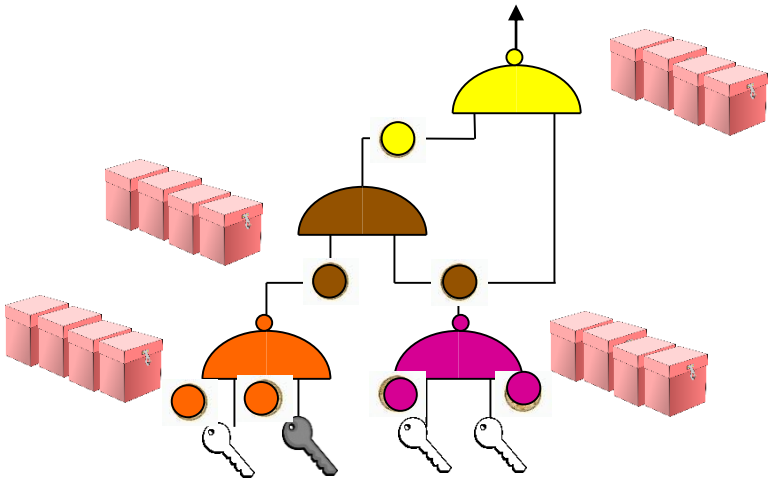
After local preprocessing,
MPRE with degree = 2 !

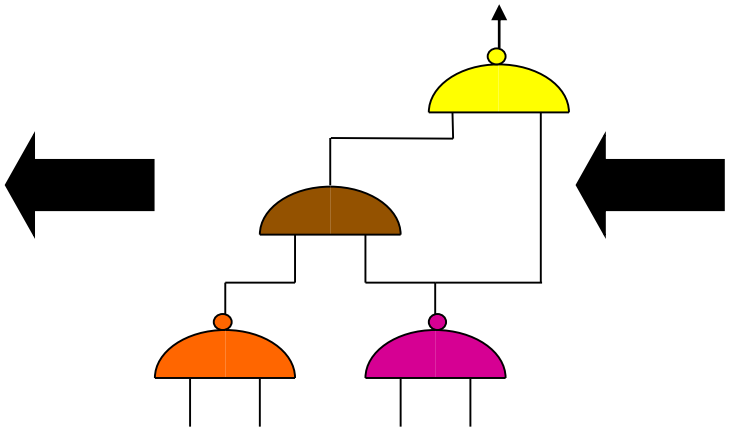


What if…?

Can we enforce such structure?





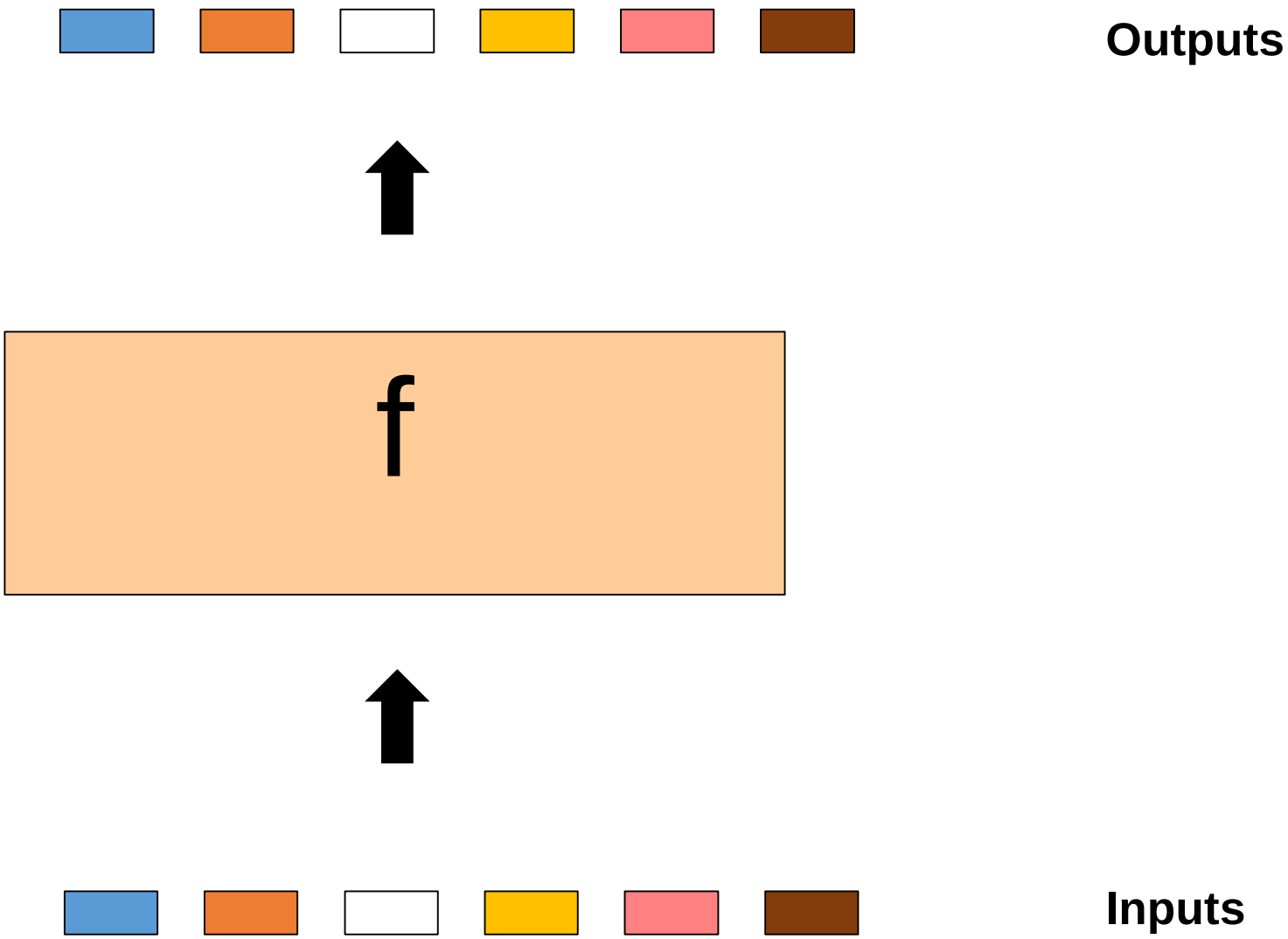


f

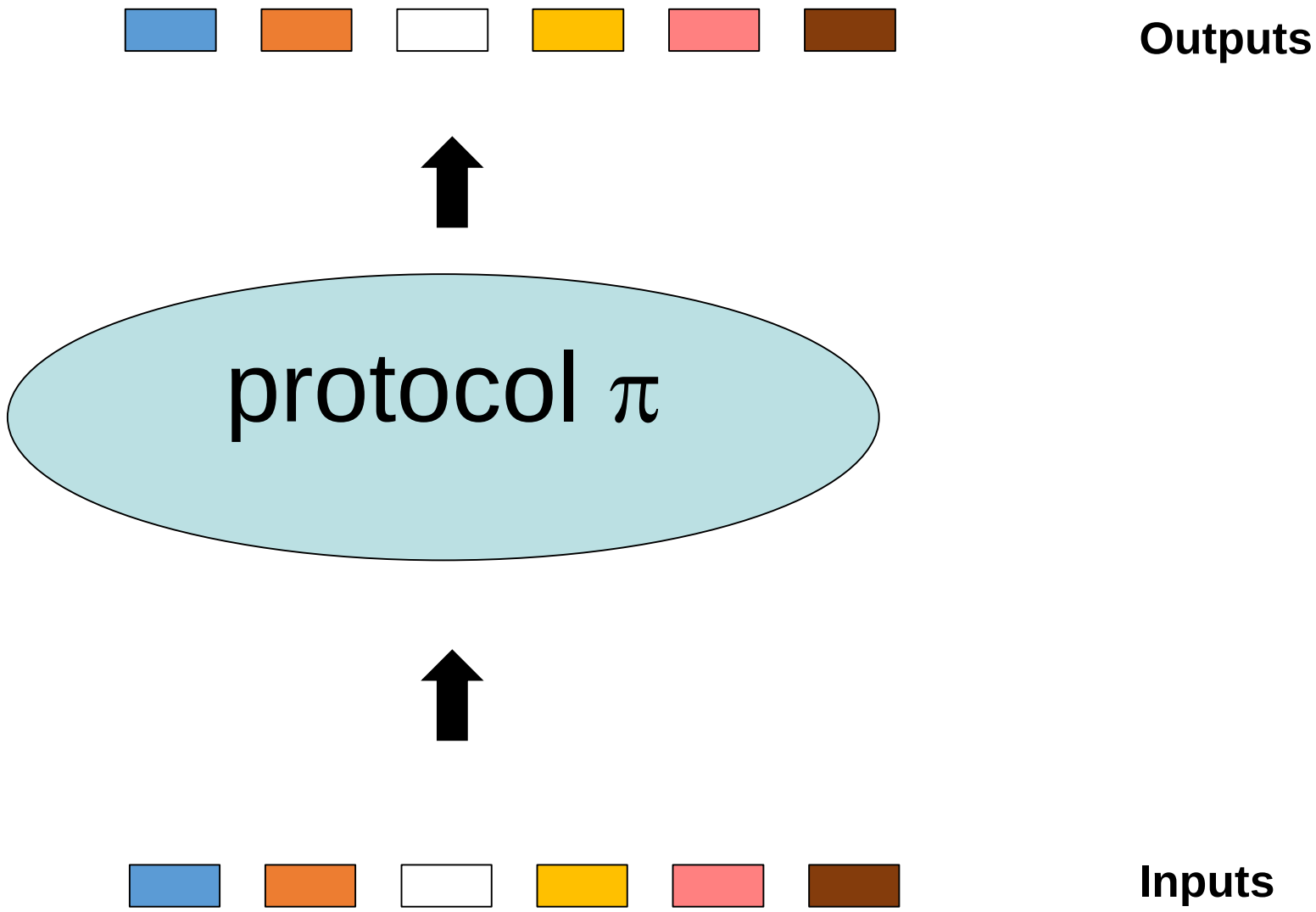
MPRE with degree = 2

nice MPRE

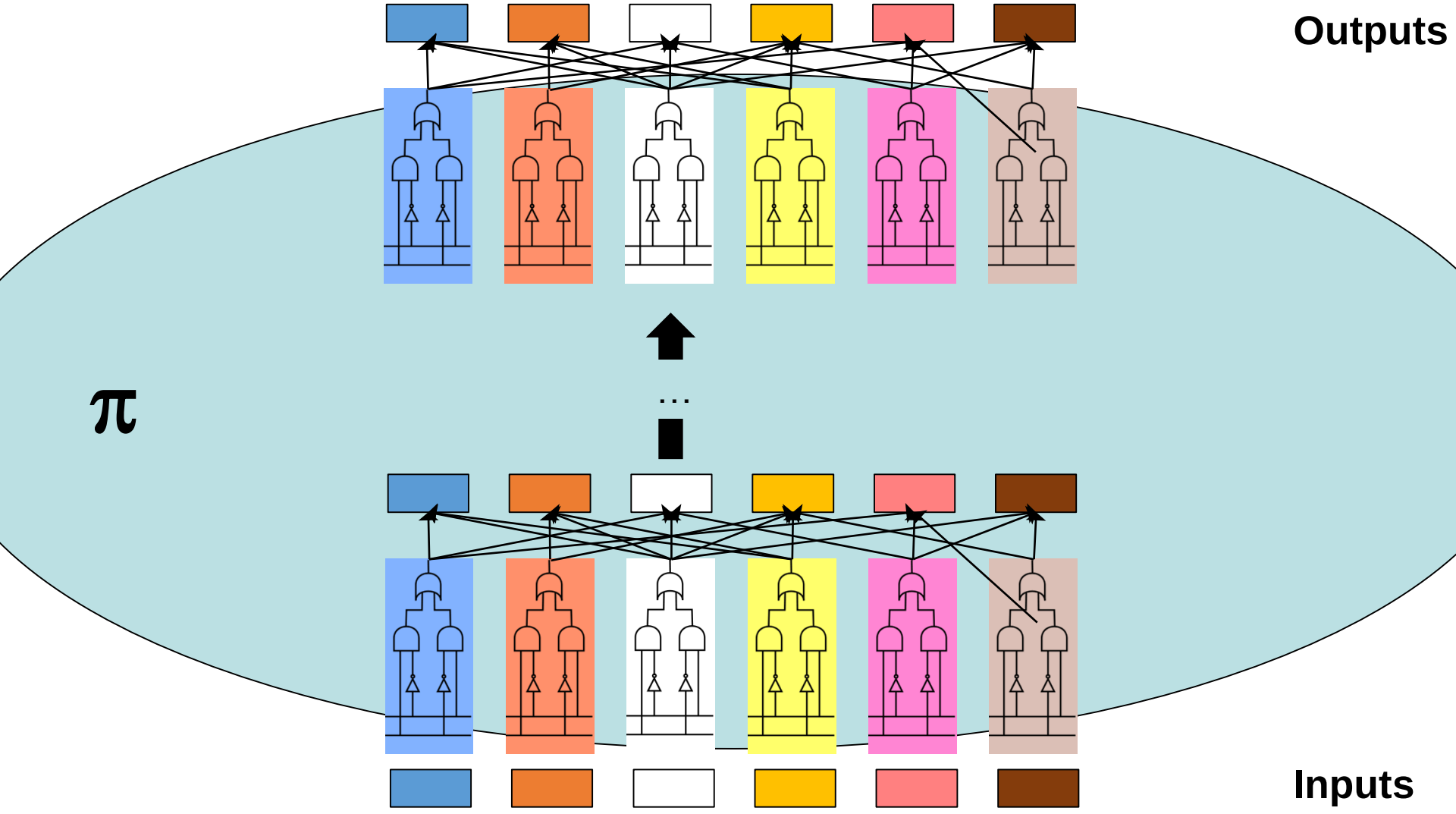
Protocol Induced MPRE



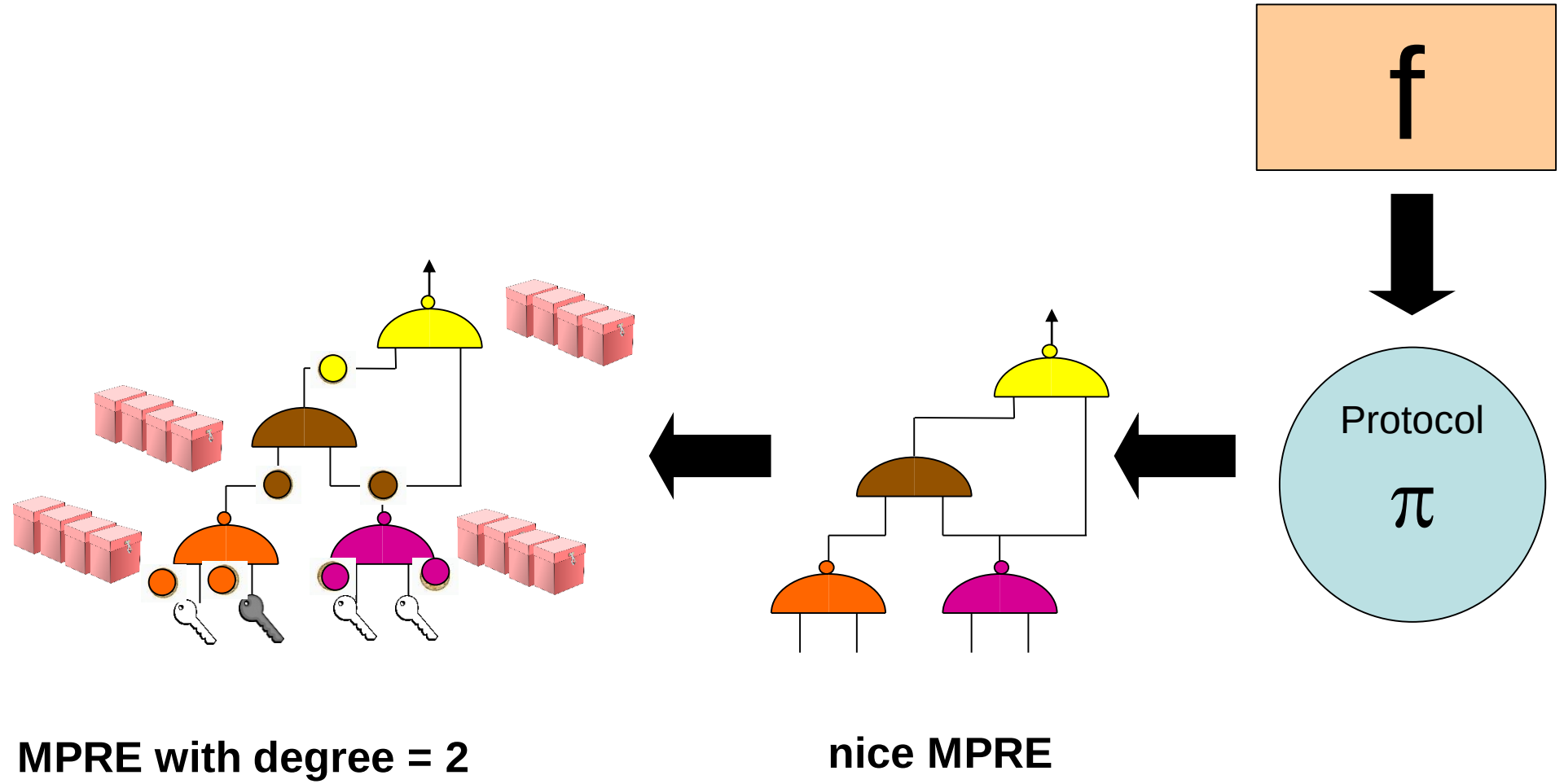
Protocol Induced MPRE



Protocol Induced MPRE



Putting It All Together



Round Complexity of MPC

Assume a protocol with T-security for f

Then f reduces to degree-2 computation with T-security

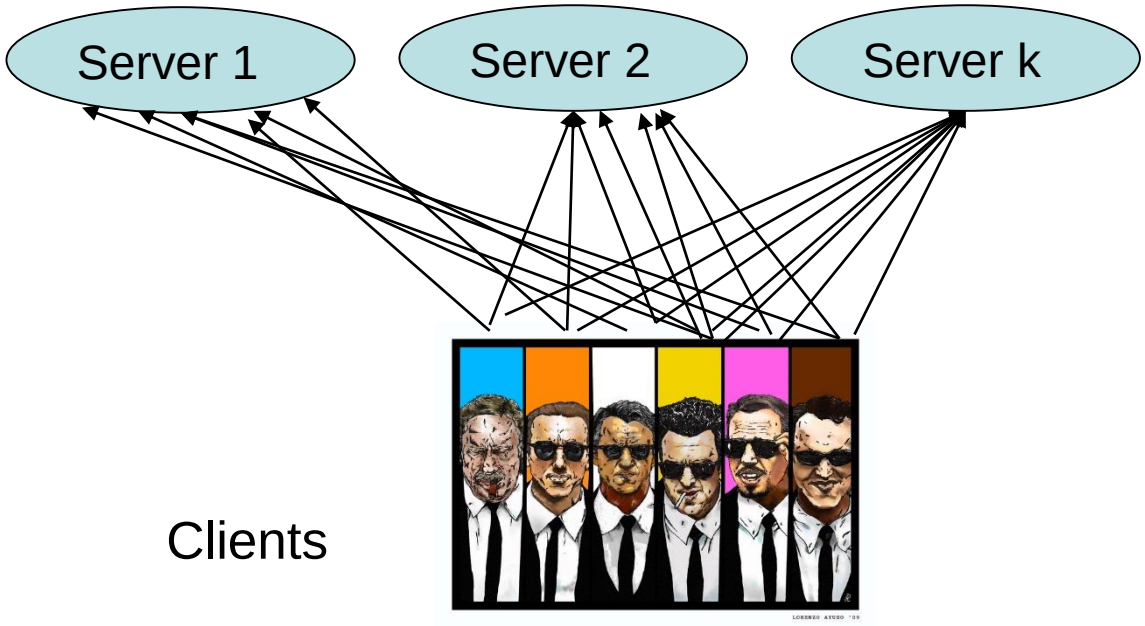
Assuming honest majority and passive adversary:

Every function has perfect 2-round protocol

- Efficient for NC1, log-space
- Computational variant for poly-size circuits using OWFs

Two-Round Protocol

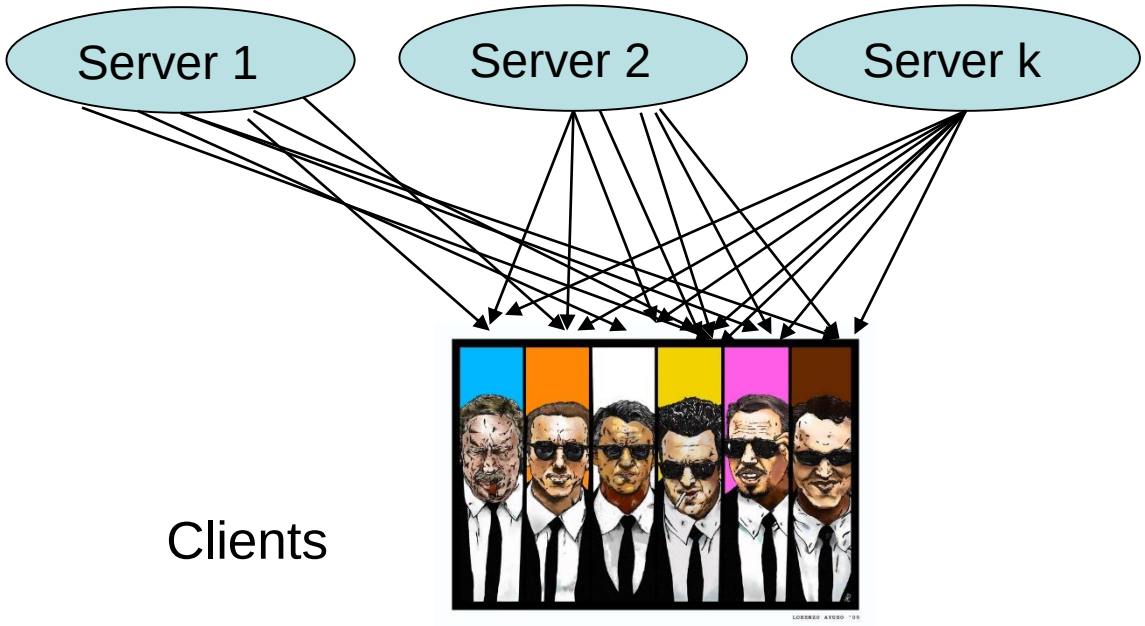
- Practical relevance?
  - 2-round protocols easily transfer to client-server model
    [Ishai-Damgard ‘05]



Two-Round Protocol

- 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



Active Adversary

Assume a protocol with T-security for f

Then f reduces to degree-2 computation with T-security

Extensions:

- Active adversaries [ABT19]

Active Adversary

Assume a protocol with T-security for f

Then f reduces to degree-2 computation with T-security

Extensions:

- Active adversaries [ABT19]
- Larger fields [AKP19]

Perfect protocols for all functions

- Passive T<N/2: 2 rounds [ABT18,GIS18]
- Active T<N/4: 3 rounds [ABT19]
- Active T<N/3: 4 rounds [AKP19]

Active Adversary

Assume a protocol with T-security for f

Then f reduces to degree-2 computation with T-security

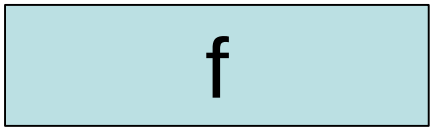
Extensions:

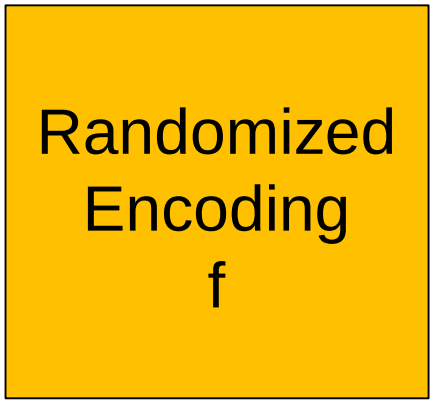
- Active adversaries [ABT19]
- Larger fields [AKP19]

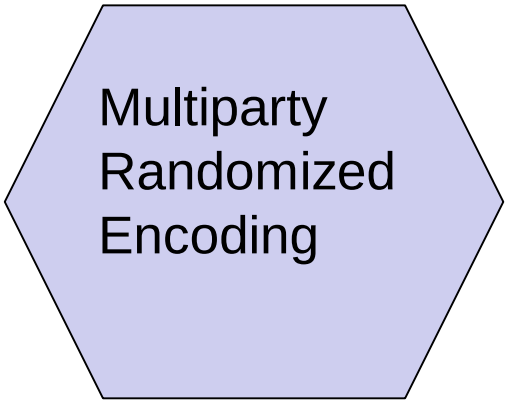
Perfect protocols for all functions

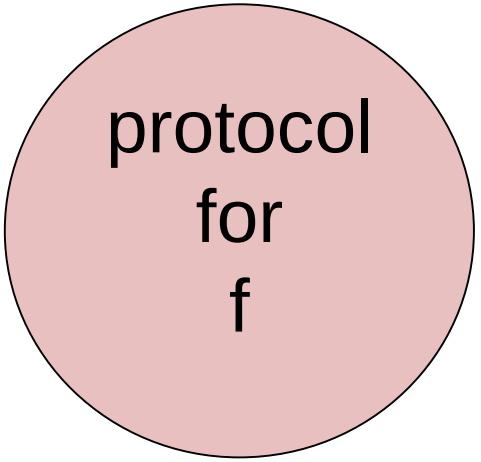
- Passive T<N/2: 2 rounds [A 18]
- Active T<N/4: 3 rounds [A
- Active T<N/3: 4 rounds [AKP 0]

Conclusion


f


Randomized
Encoding
f


Multiparty
Randomized
Encoding


protocol
for
f

Take Home Message

Abstraction is powerful

but tight results

may require refined tools

Thank You