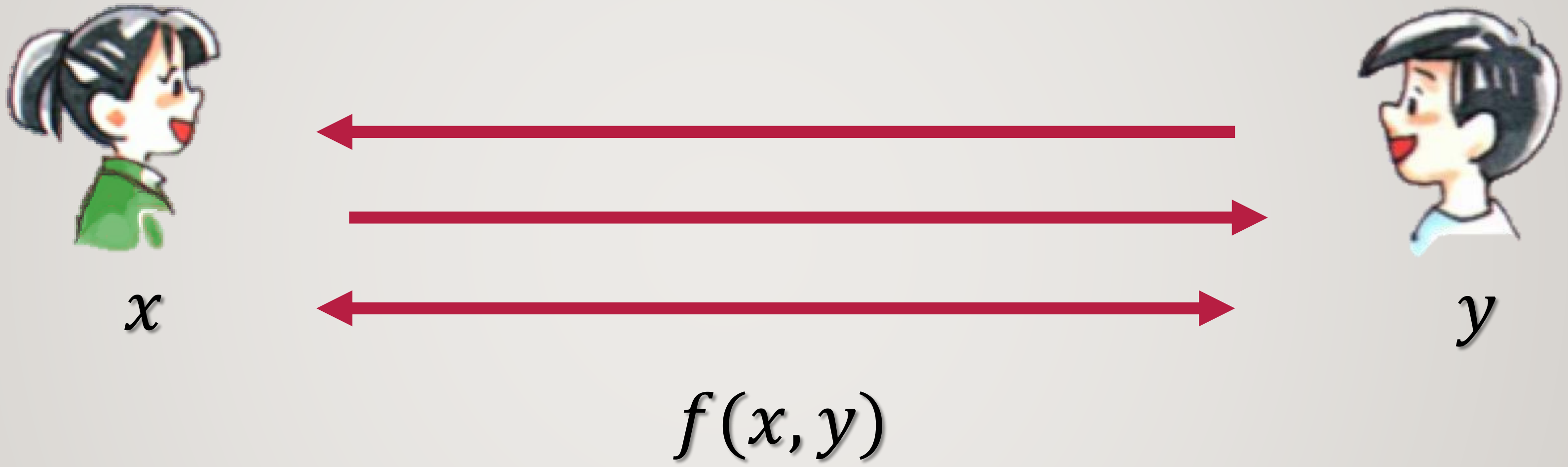
Faster Malicious 2-party Secure Computation with Online/Offline Dual Execution

Peter Rindal

Mike Rosulek

Oregon State University

2 Party Computation

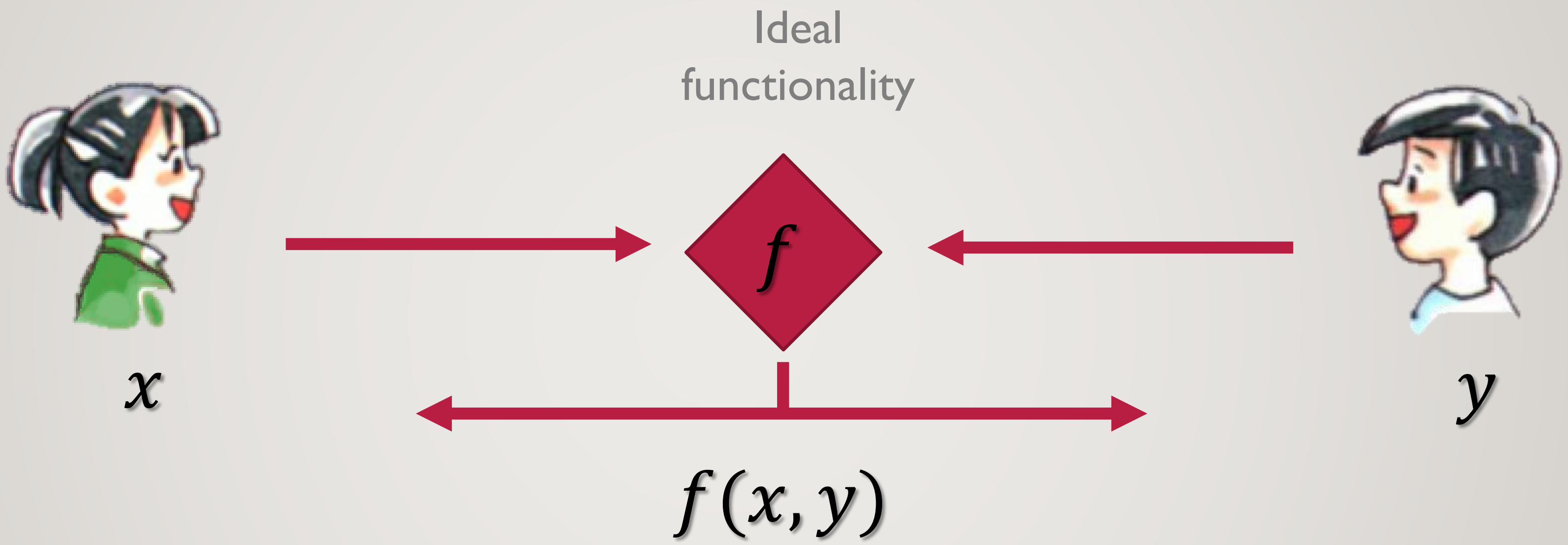


x

y

$f(x,y)$

2 Party Computation



Ideal functionality

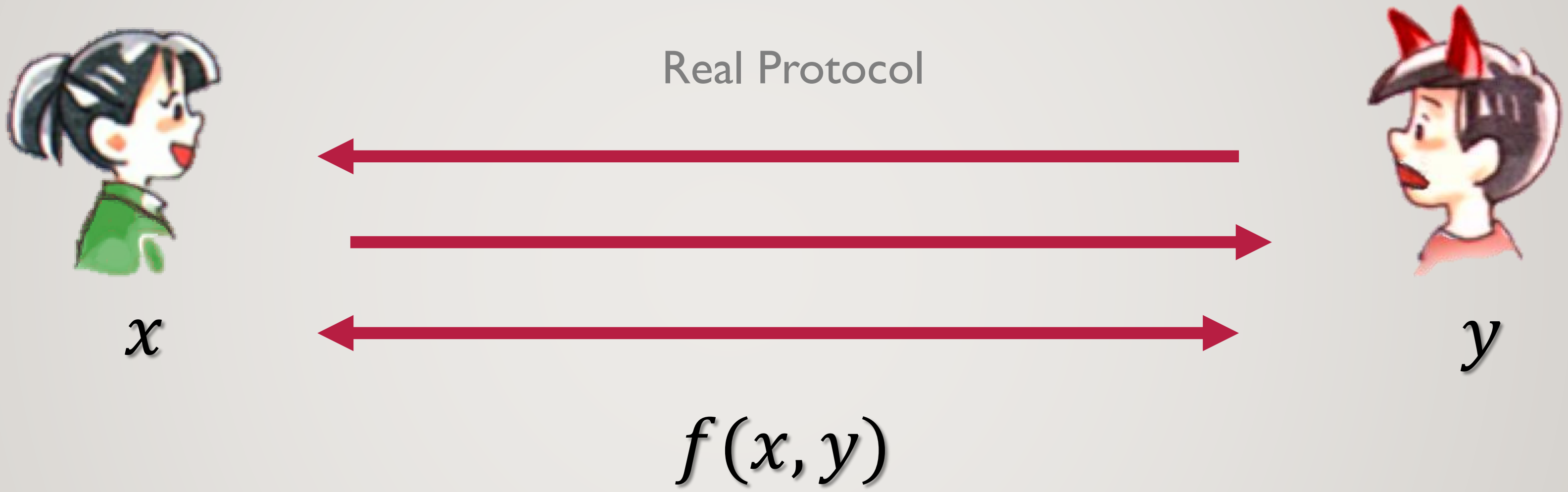
f

x

y

$f(x,y)$

2 Party Computation



Real Protocol

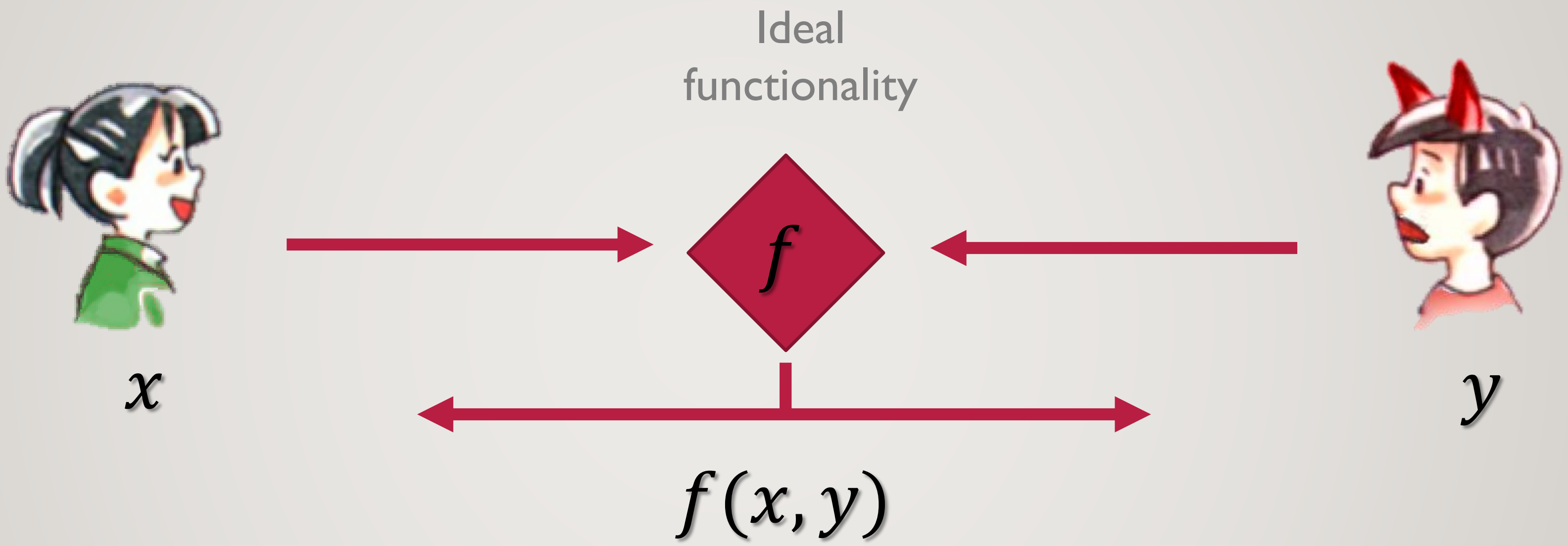
x

y

$f(x, y)$

- Secure against malicious adversaries

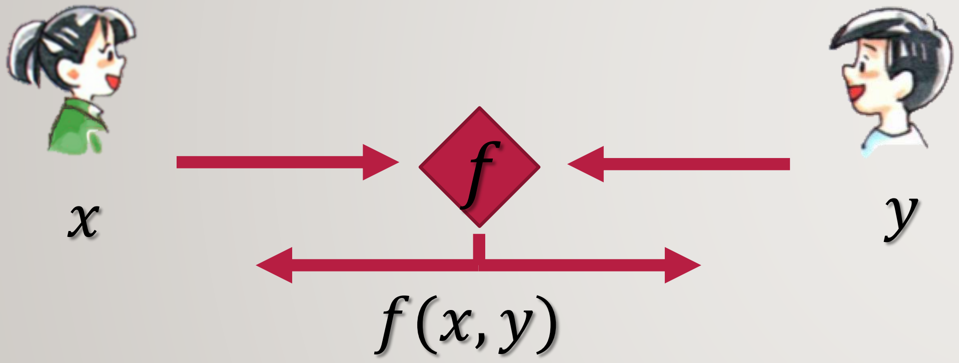
2 Party Computation



- Secure against malicious adversaries

Applications

2-party Secure Computation

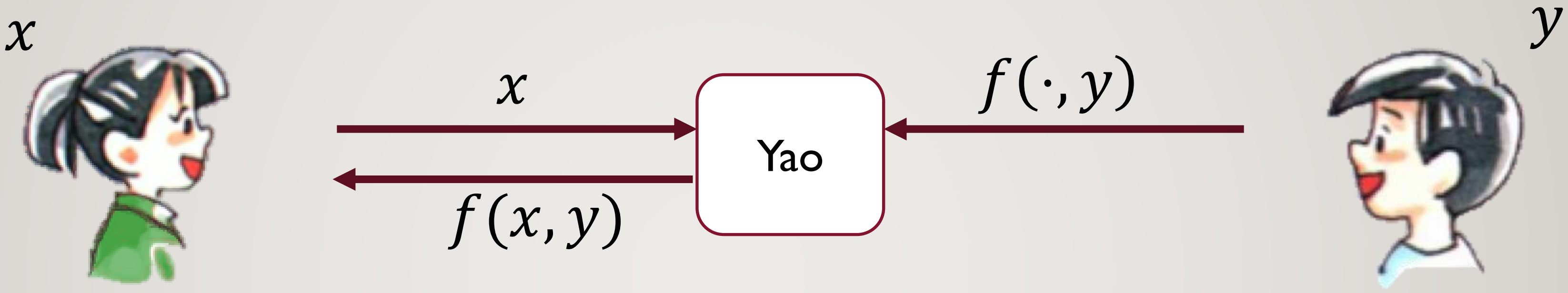


Applications

- Private database querying

- Joint machine learning
- Secure auctions

Yao's Protocol



x

x

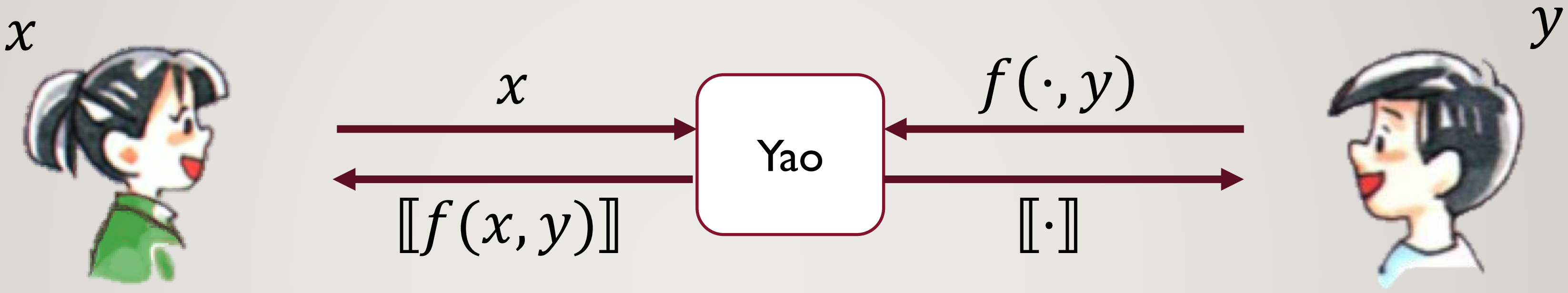
$f(x, y)$

Yao

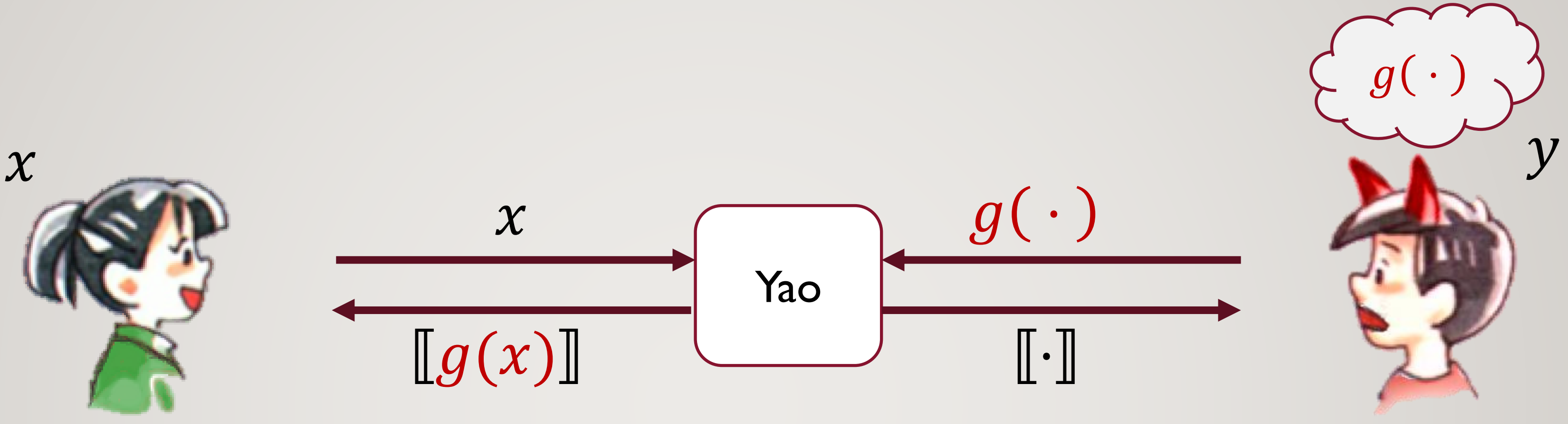
$f(\cdot, y)$

y

Yao’s Protocol



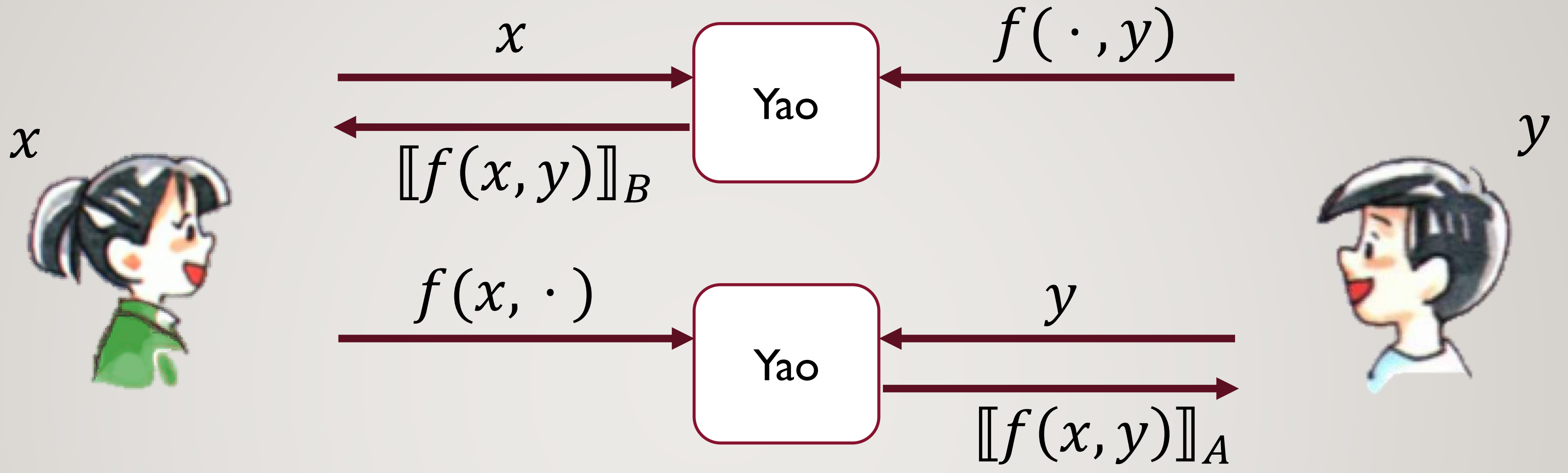
Yao’s Protocol



Problems with malicious Adversaries

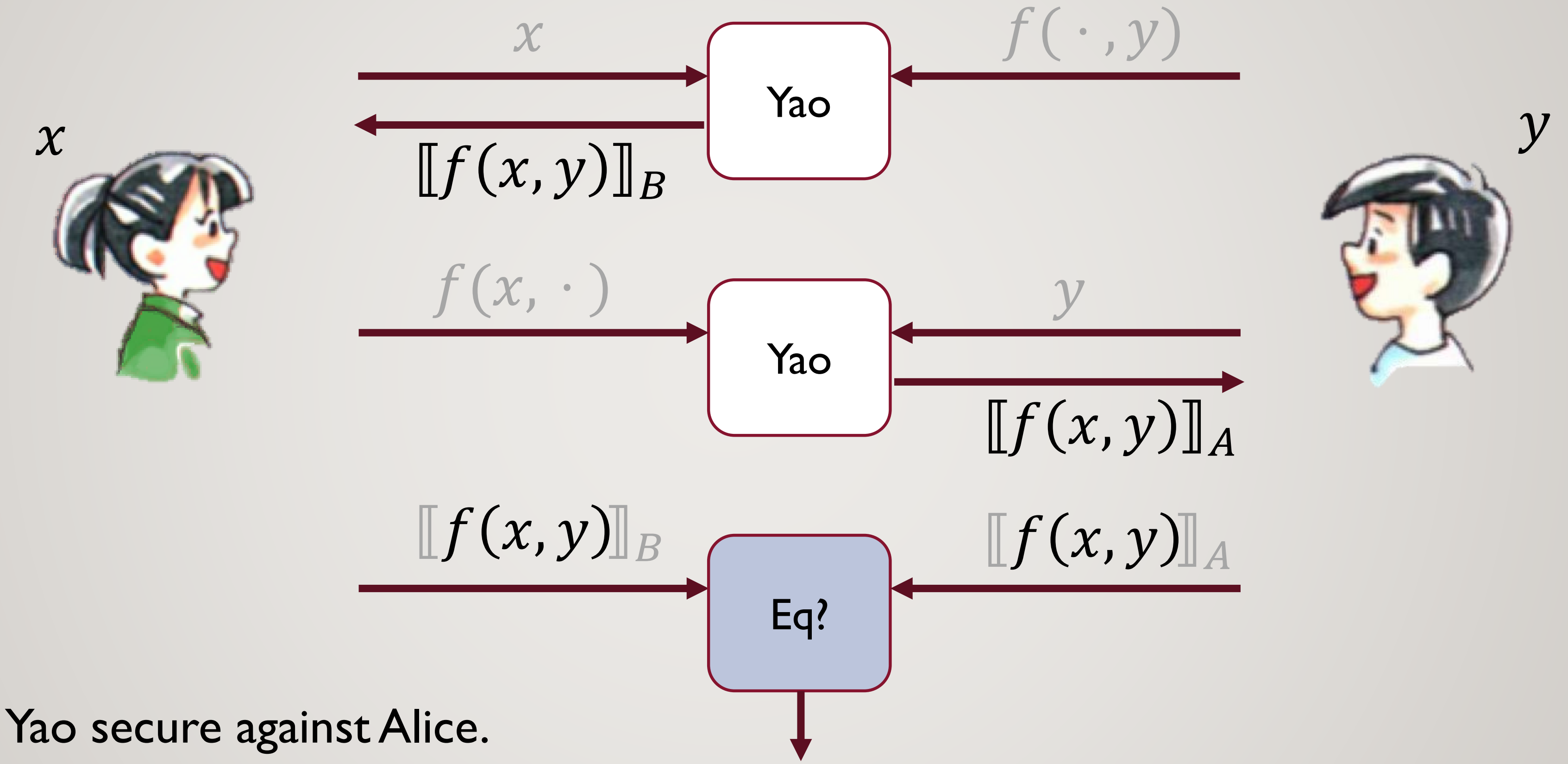
- The circuit may not be correctly constructed
  - E.g. $g(x) \coloneqq x$
- May leak Alice’s input!
- Not always detectable

Dual Execution [MohasselFranklin06]



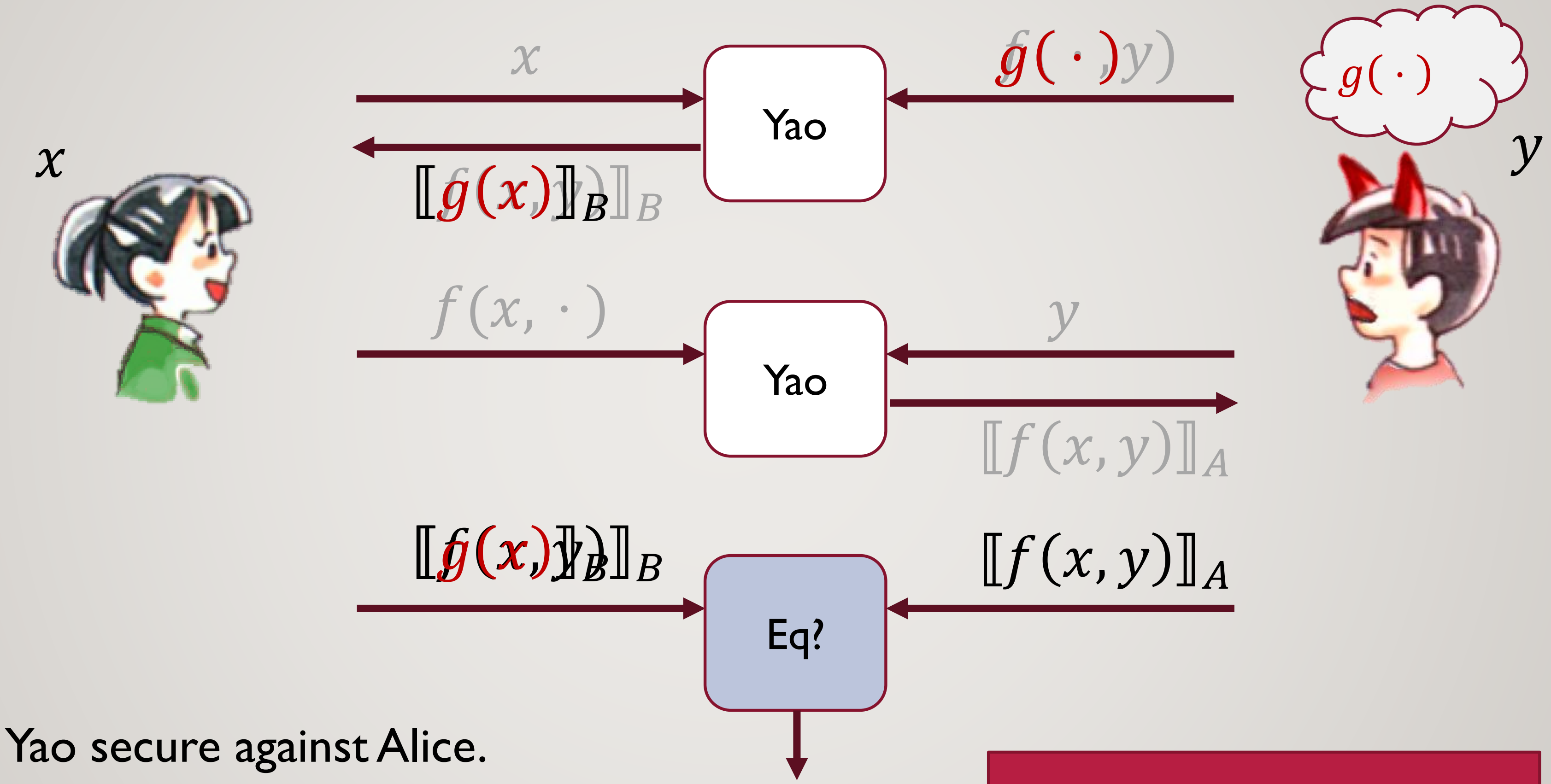
- First Yao secure against Alice.
- Second Yao secure against Bob

Dual Execution [MohasselFranklin06]



- First Yao secure against Alice.
- Second Yao secure against Bob

Dual Execution [MohasselFranklin06]

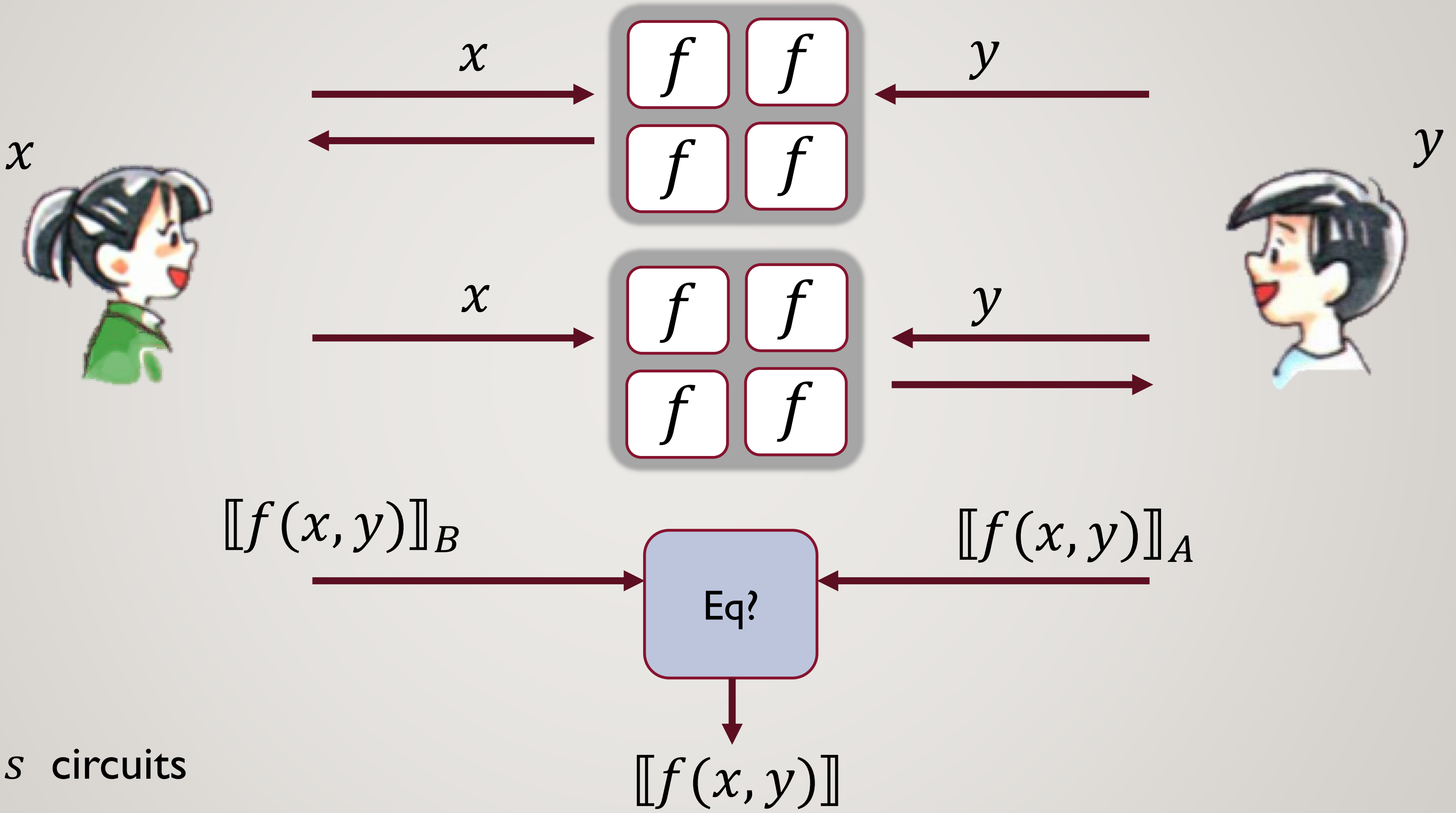


- First Yao secure against Alice.
- Second Yao secure against Bob
- Equality leaks $g(x) = f(x,y)$

Malicious secure.

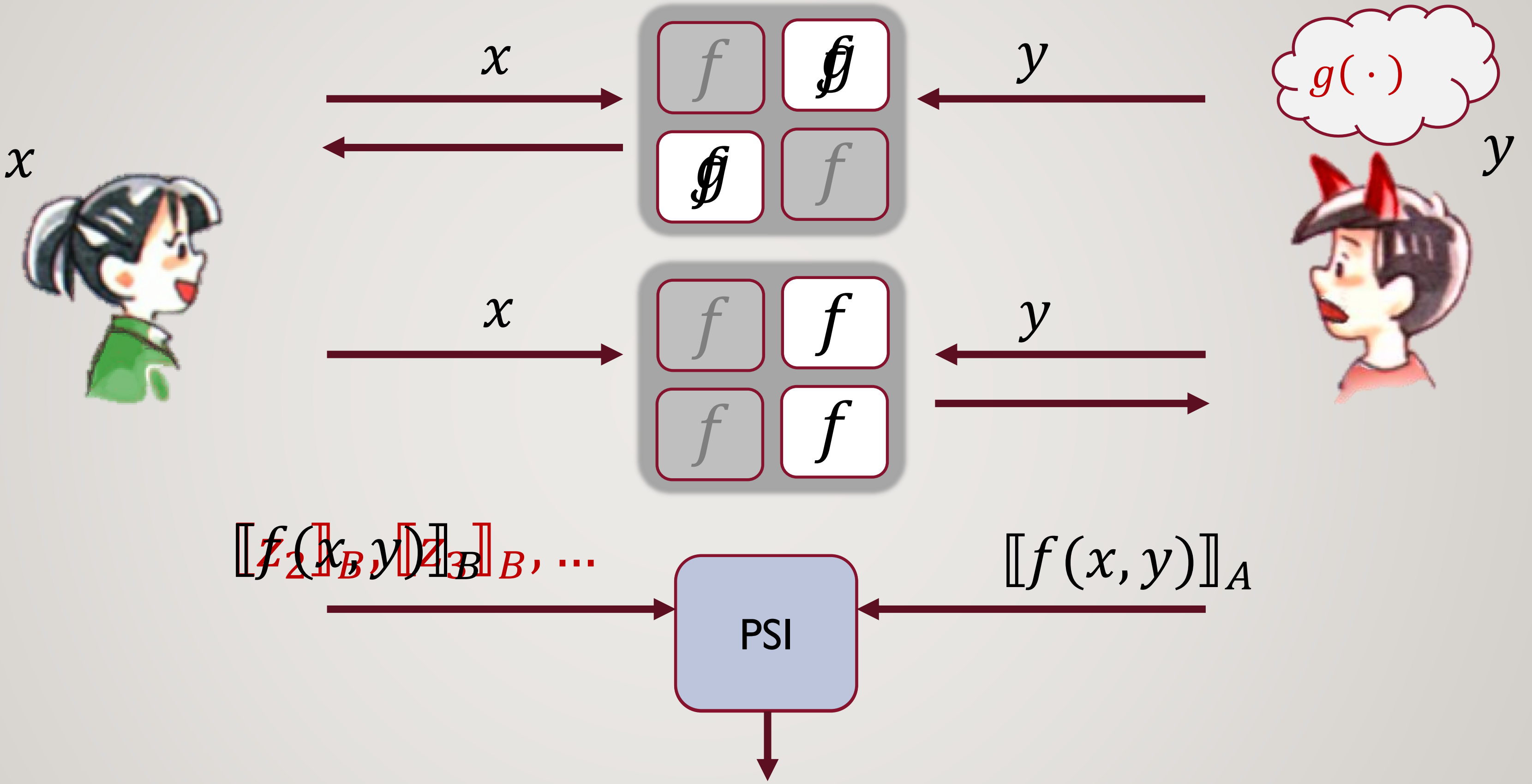
Leaks only a single bit!

Dual Execution [KolesnikovMohasselRivaRosulek15]



- Send s circuits
- Check some for correctness

Dual Execution [KolesnikovMohasselRivaRosulek15]

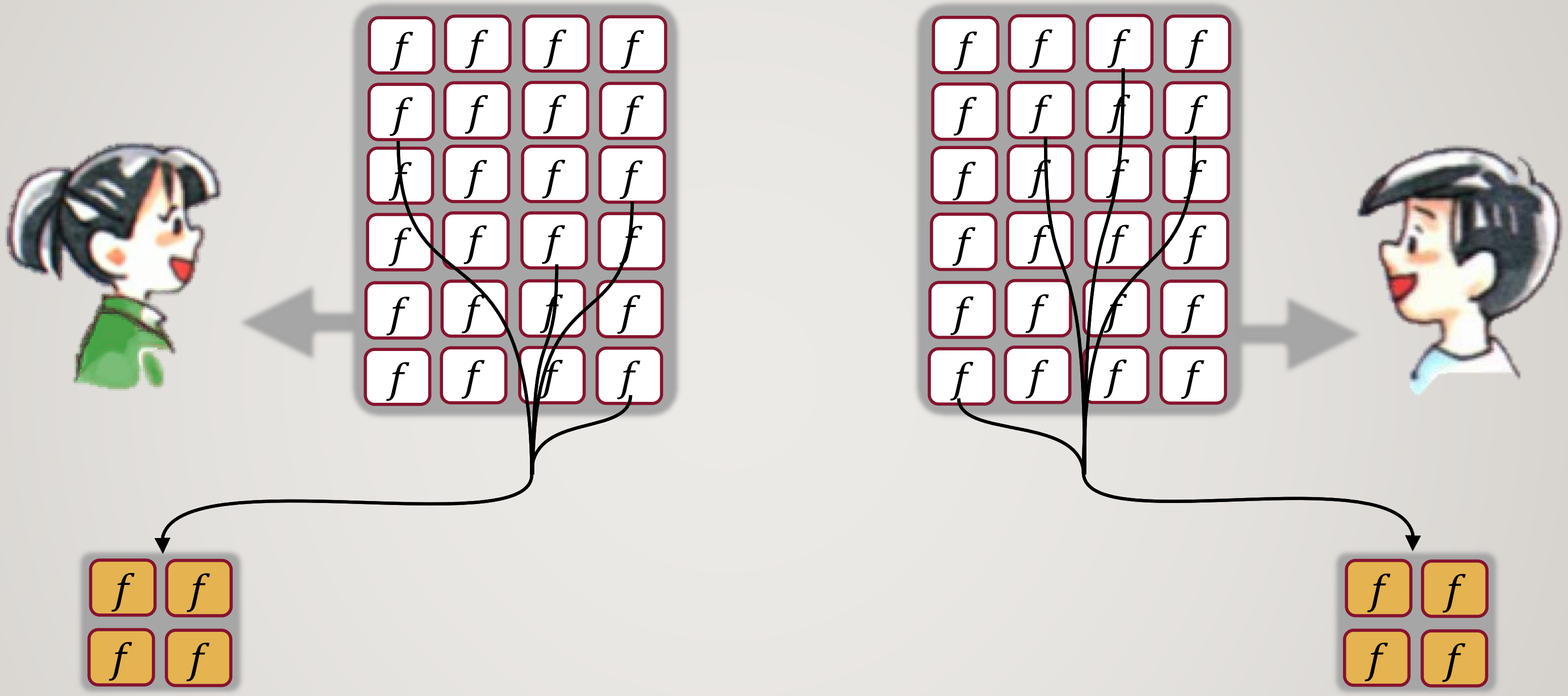


$[\![f(x,y)]\!]$ or $\emptyset$

- Send s circuits
- Check some for correctness
- PSI leaks $\forall\, i \;:\; g_i(x,y) \neq f(x,y)$

$\mathbf{Pr}[\boldsymbol{leak\ a\ bit}] = \mathbf{2^{-s}}$

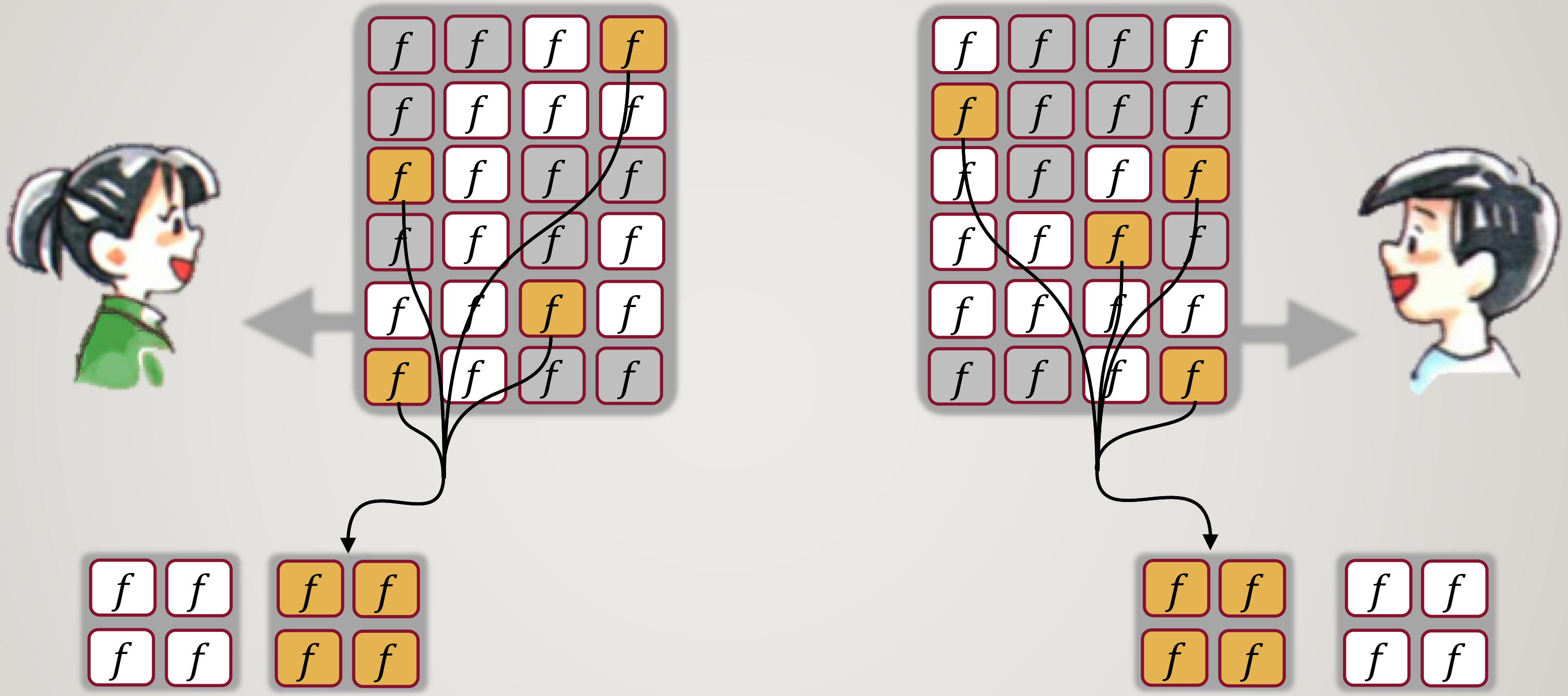
Online – Offline [LindellRiva14,NeilsenOrlandi08,**R**Rosulek16]



- Want to perform N executions of f
  - Construct enough circuits for all N executions
  - Check some for correctness

- Randomly map the rest into bins
  - **$\log N$ times fewer circuits**

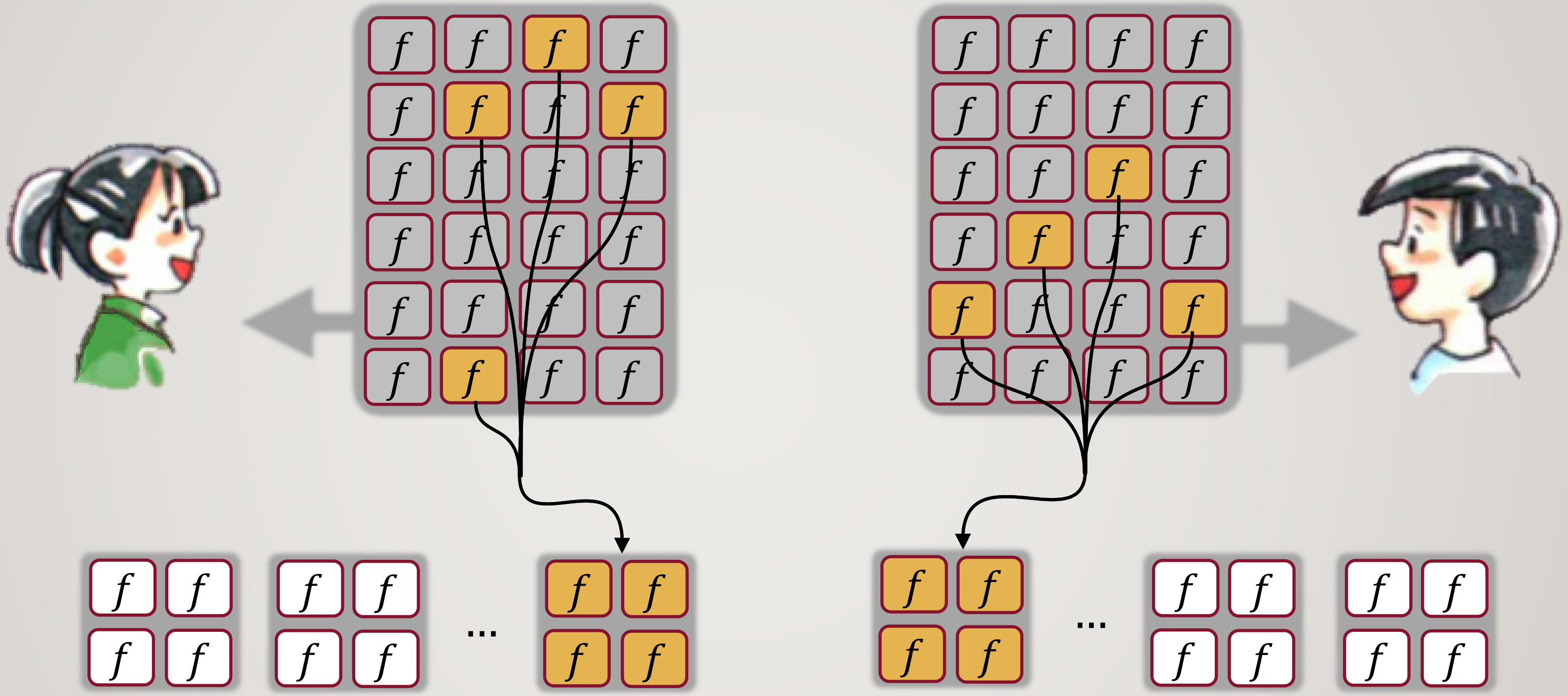
Online – Offline [LindellRiva14,NeilsenOrlandi08,RRosulek16]



- Want to perform N executions of f
  - Construct enough circuits for all N executions
  - Check some for correctness

- Randomly map the rest into bins
  - **$\log N$ times fewer circuits**

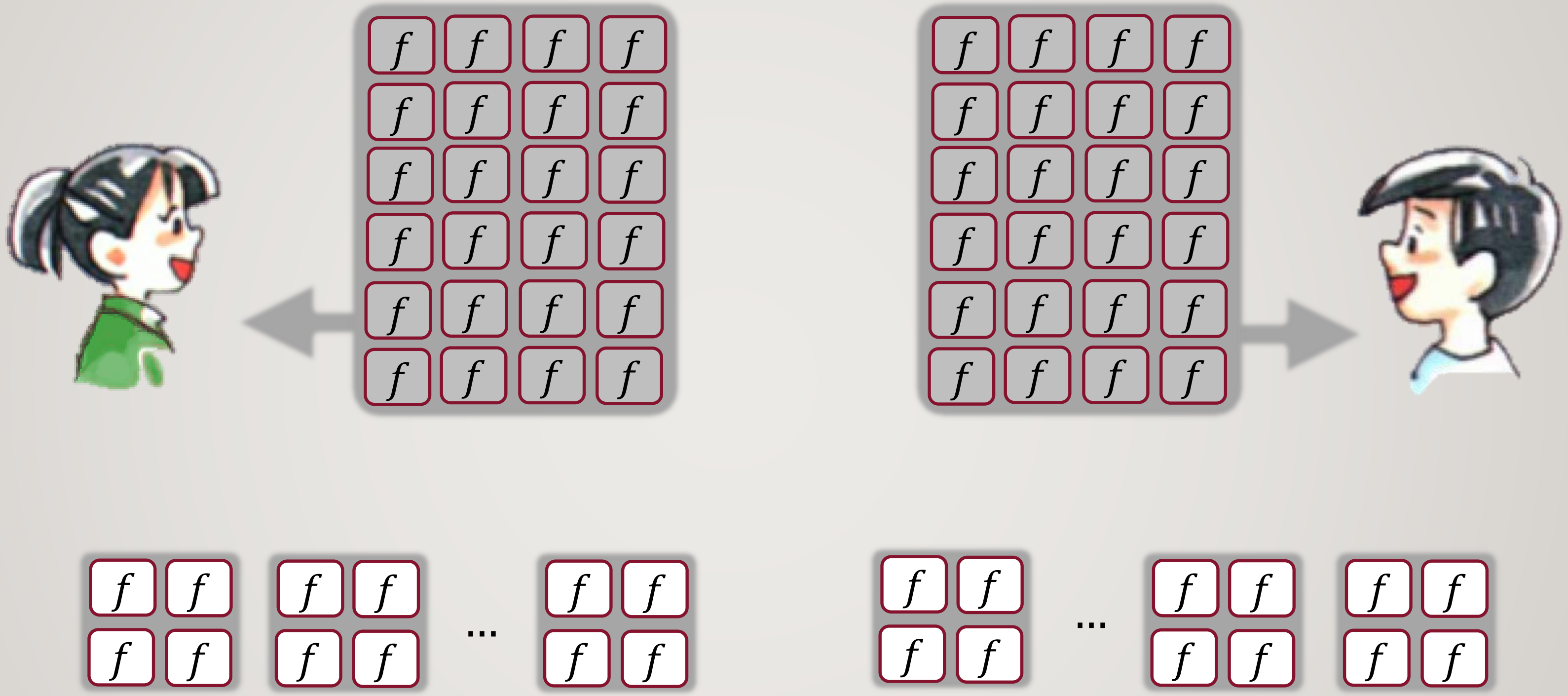
Online – Offline [LindellRiva14,NeilsenOrlandi08,RRosulek16]



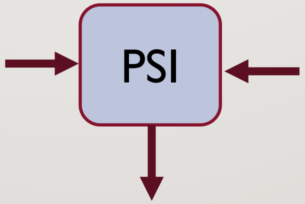
- Want to perform N executions of f
  - Construct enough circuits for all N executions
  - Check some for correctness

- Randomly map the rest into bins
  - **log N times fewer circuits**

Online – Offline [LindellRiva14,NeilsenOrlandi08,RRosulek16]



- Use one bin per evaluation

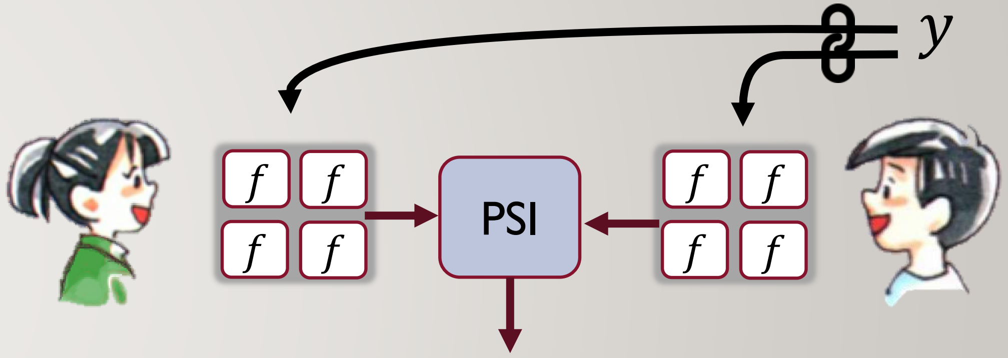


$$[\![f(x,y)]\!]$$

Challenge #1: Input Consistency [RRosulek16]

How to ensure Bob used the same y in all circuits?

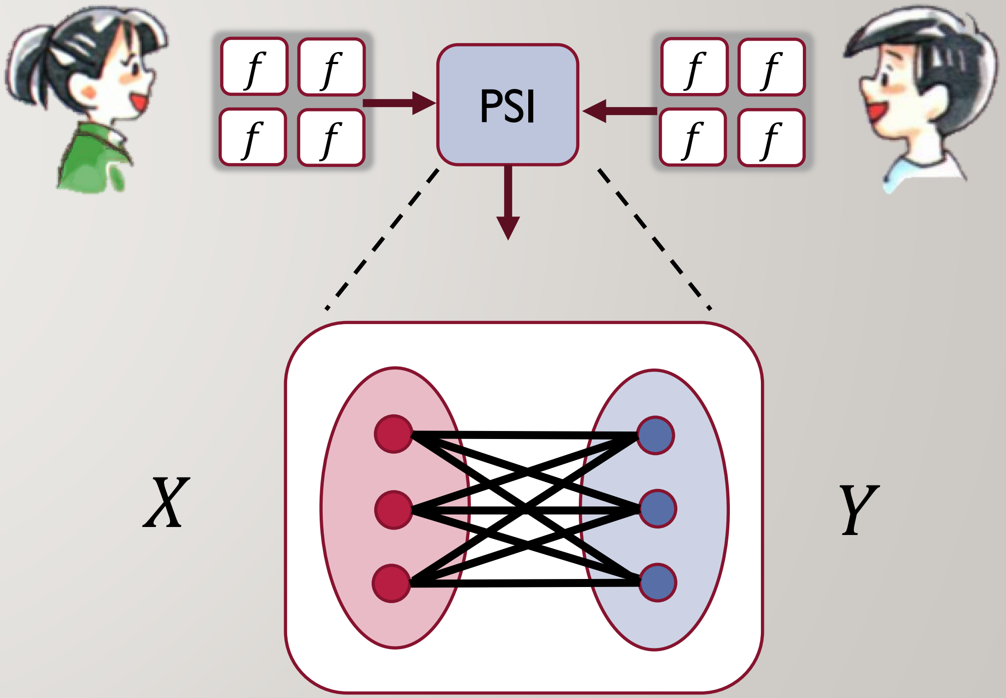
- Bob will have consistent inputs for Alice's circuits.
  - Enforced by the oblivious transfer protocol



- How to enforce consistency on other circuit
  - In the offline, Bob tells Alice the relationship between the two arrows
  - Check in the cut and choose
  - Consistent with the relationship ⇒ used same y in all circuits
    - Requires **no crypto operations**

Challenge #2: Private Set Intersection (PSI) [RRosulek16]

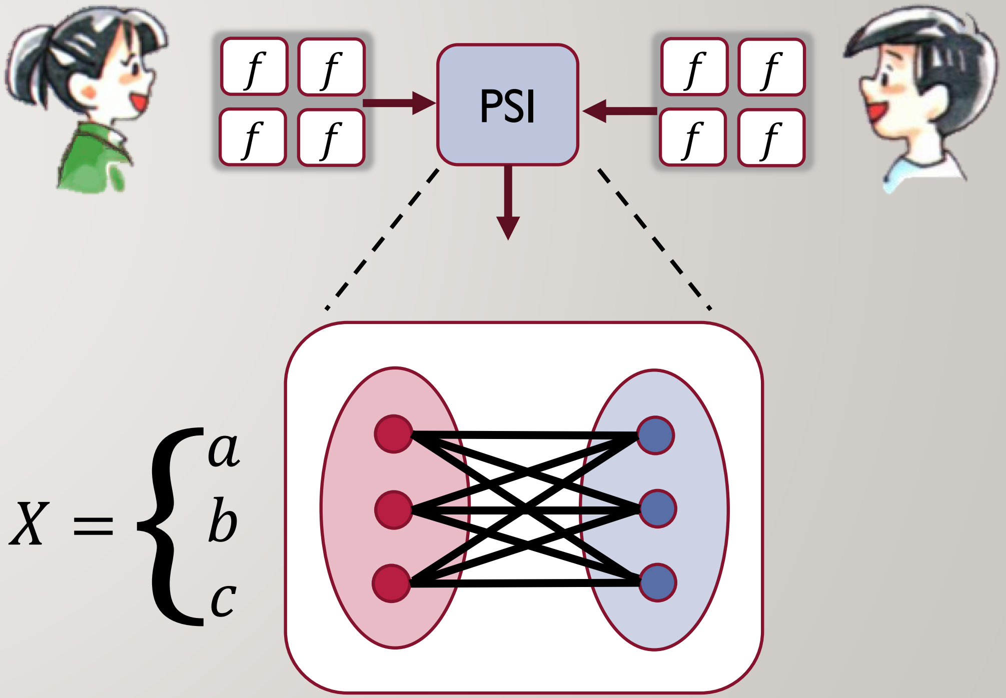
- Build PSI from Private Equality Test [PinkasSchneiderZohner14]



Challenge #2: Private Set Intersection (PSI) [RRosulek16]

- Build PSI from Private Equality Test [PinkasSchneiderZohner14]

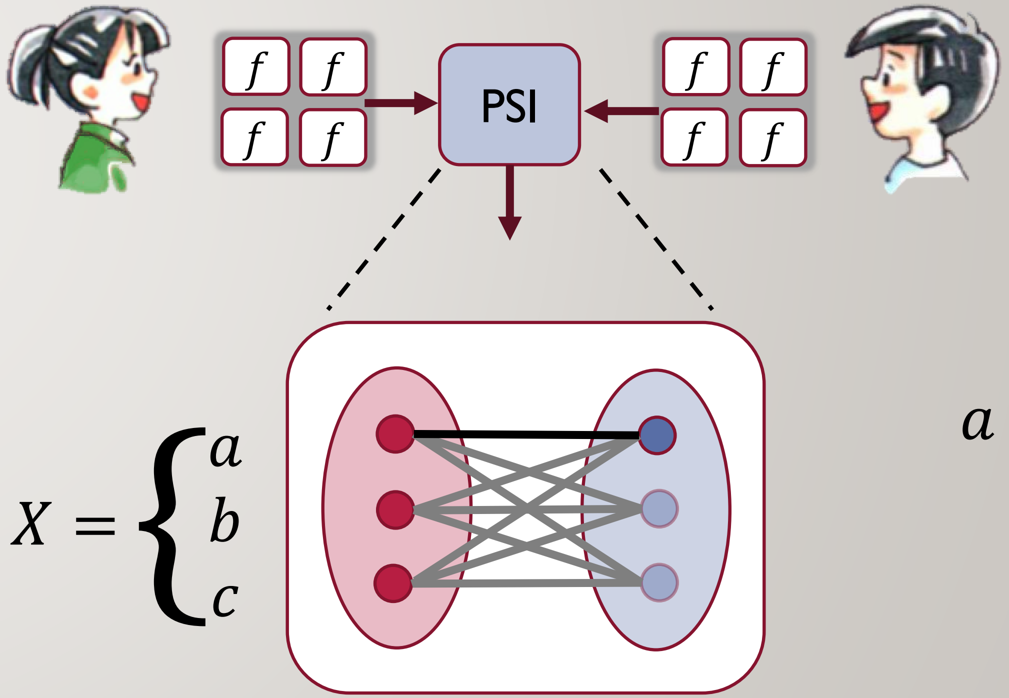
- Issues: Not malicious secure in general
  - Can not be simulated



Challenge #2: Private Set Intersection (PSI) [RRosulek16]

- Build PSI from Private Equality Test [PinkasSchneiderZohner14]

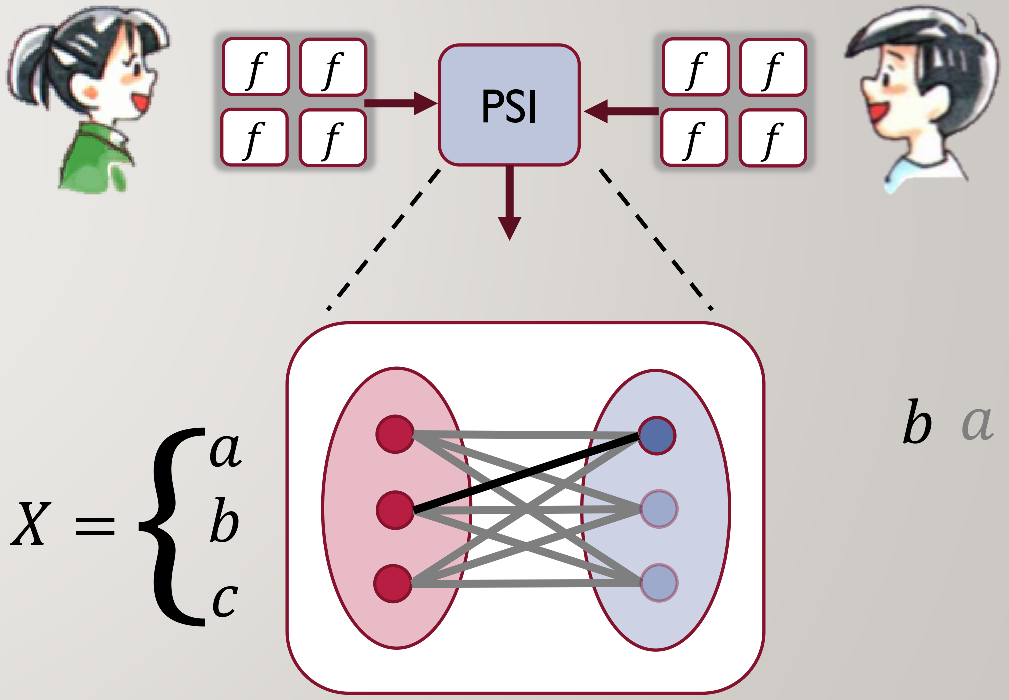
- Issues: Not malicious secure in general
  - Can not be simulated



Challenge #2: Private Set Intersection (PSI) [RRosulek16]

- Build PSI from Private Equality Test [PinkasSchneiderZohner14]

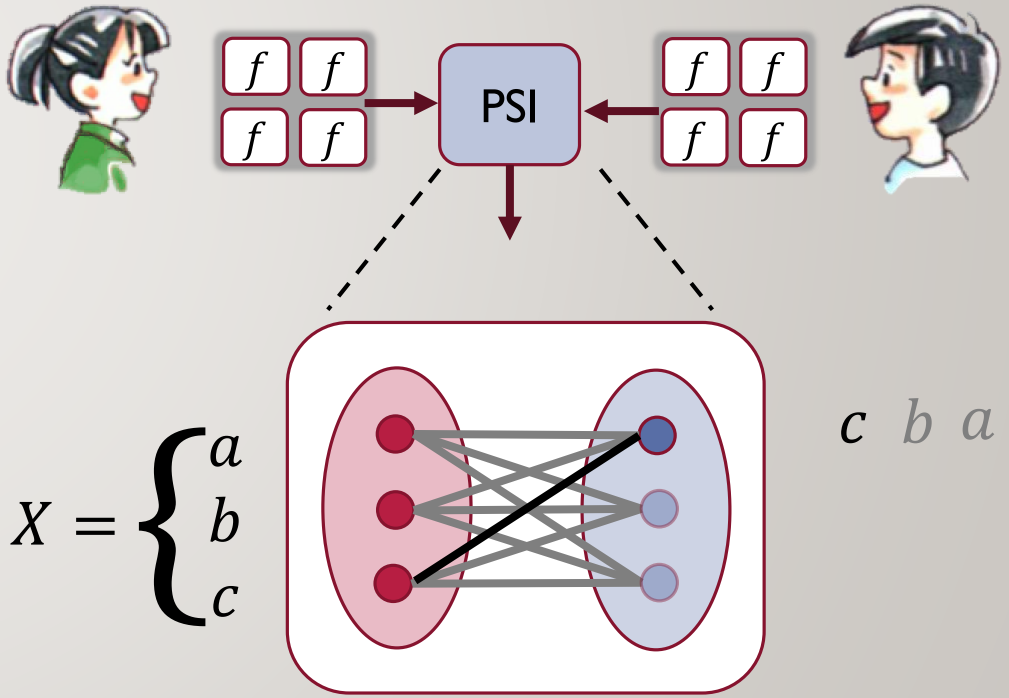
- Issues: Not malicious secure in general
  - Can not be simulated



Challenge #2: Private Set Intersection (PSI) [RRosulek16]

- Build PSI from Private Equality Test [PinkasSchneiderZohner14]

- Issues: Not malicious secure in general
  - Can not be simulated



Challenge #2: Private Set Intersection (PSI) [RRosulek16]

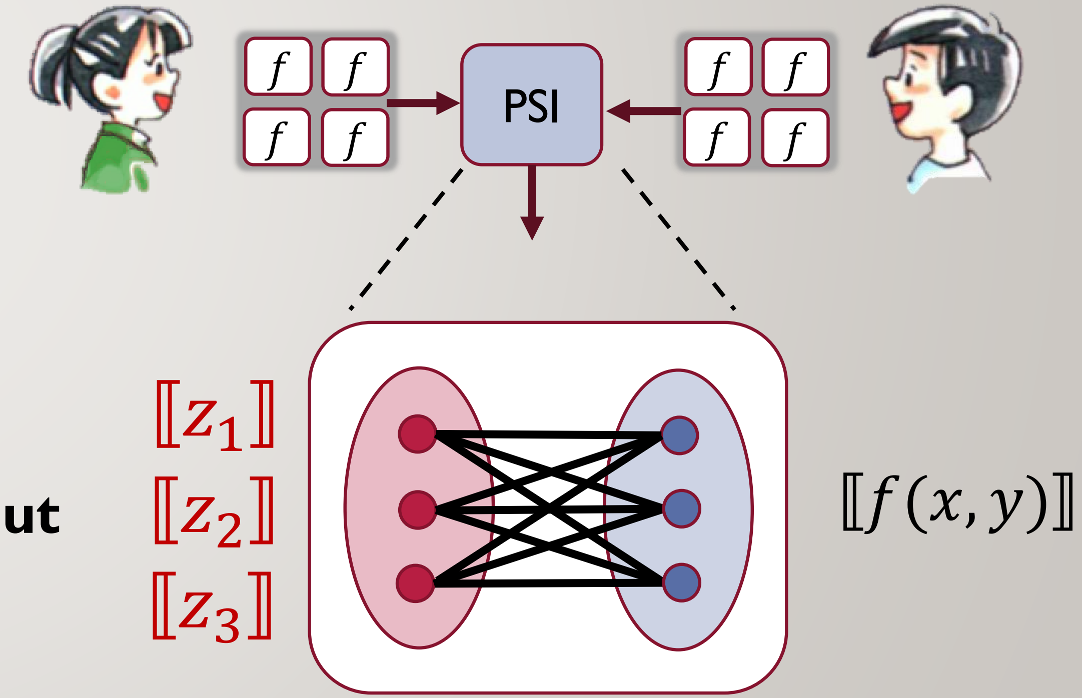
- Build PSI from Private Equality Test [PinkasSchneiderZohner14]

- Issues: Not malicious secure in general
  - Can not be simulated

- Ideal: Bob only knows one valid PSI input

$$[\![f(x,y)]\!]$$

- Simulator **doesn't need to extract Bob input**
  - Just test if it contains $[\![f(x,y)]\!]$

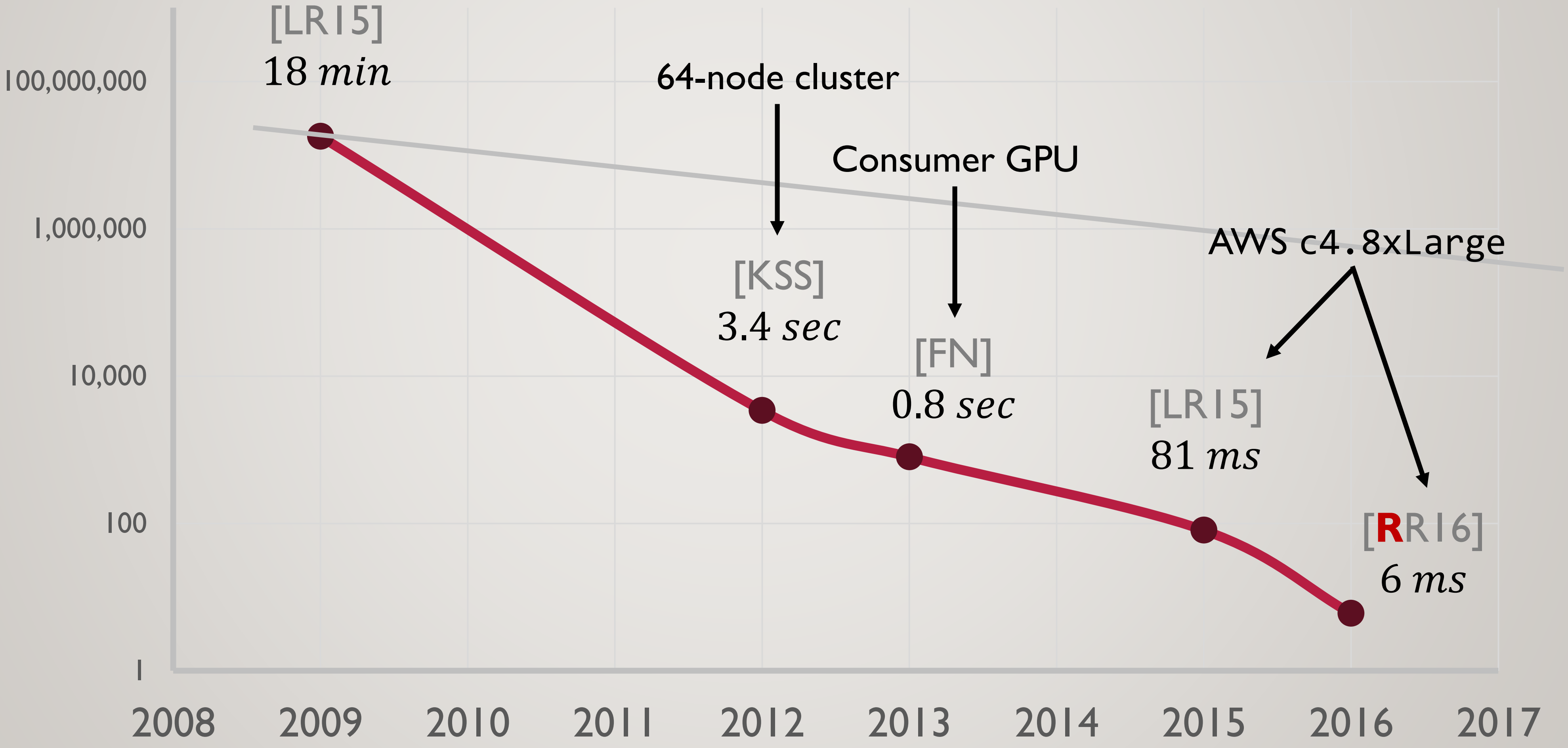


Performance

Function	[RRosulek16]		[LindellRiva15]		[DamgårdZakarias15]	
	Offline	Online	Offline	Online	Offline	Online
AES	$\mathbf{5.1}\ \boldsymbol{ms}$	$\mathbf{1.3}\ \boldsymbol{ms}$	$74\ ms$	$7\ ms$	high?	$6\ ms$
SHA-256	$\mathbf{48.0}\ \boldsymbol{ms}$	$\mathbf{8.1}\ \boldsymbol{ms}$	$206\ ms$	$33\ ms$	-	-

- Amortized cost for $N = 1{,}024$ evaluations
  - Amazon `c4.8xLarge` = 36 core, 64GB RAM
  - Statistical security $\kappa = 40$
- Maximum **throughput**: $0.26\ ms$ / AES block (3800+ Hz)
  - [DamgårdZakarias15] report $0.4\ ms$

Total Protocol Times for AES



The End

Thanks

Faster Malicious 2-party Secure Computation with Online/Offline Dual Execution

github.com/osu-crypto/batchDualEx

Peter Rindal
Mike Rosulek

Oregon State University