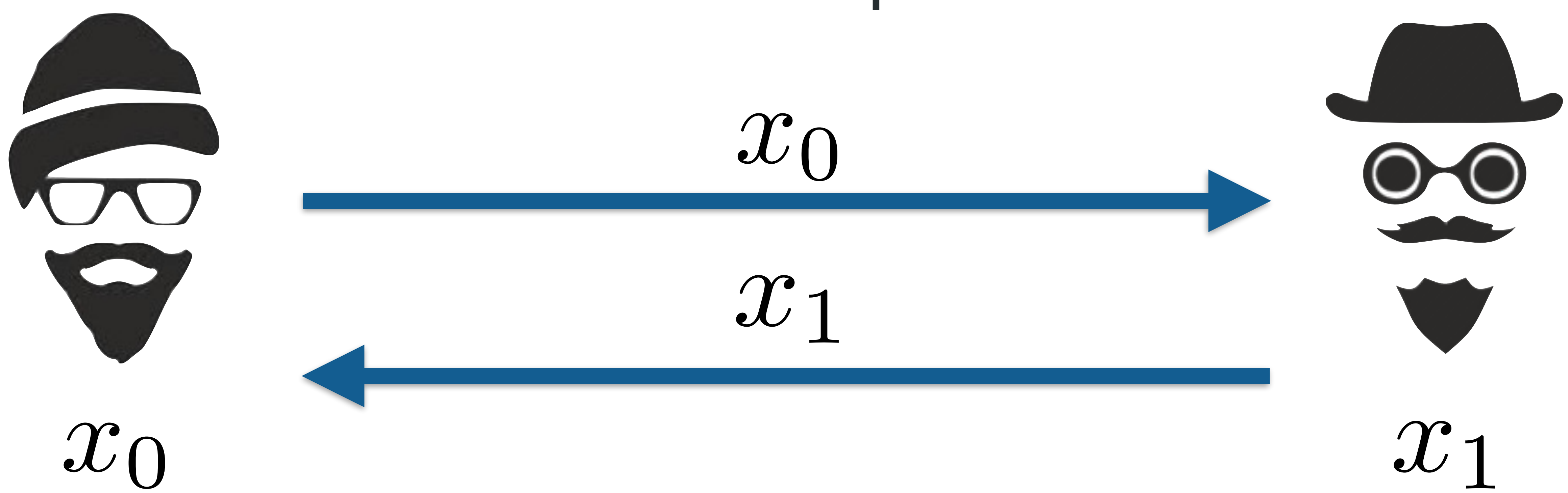
On the Communication Complexity of Multiparty Computation in the Correlated Randomness Model

Geoffroy Couteau

Karlsruher Institut für Technologie

The Quest for MPC with Low Communication

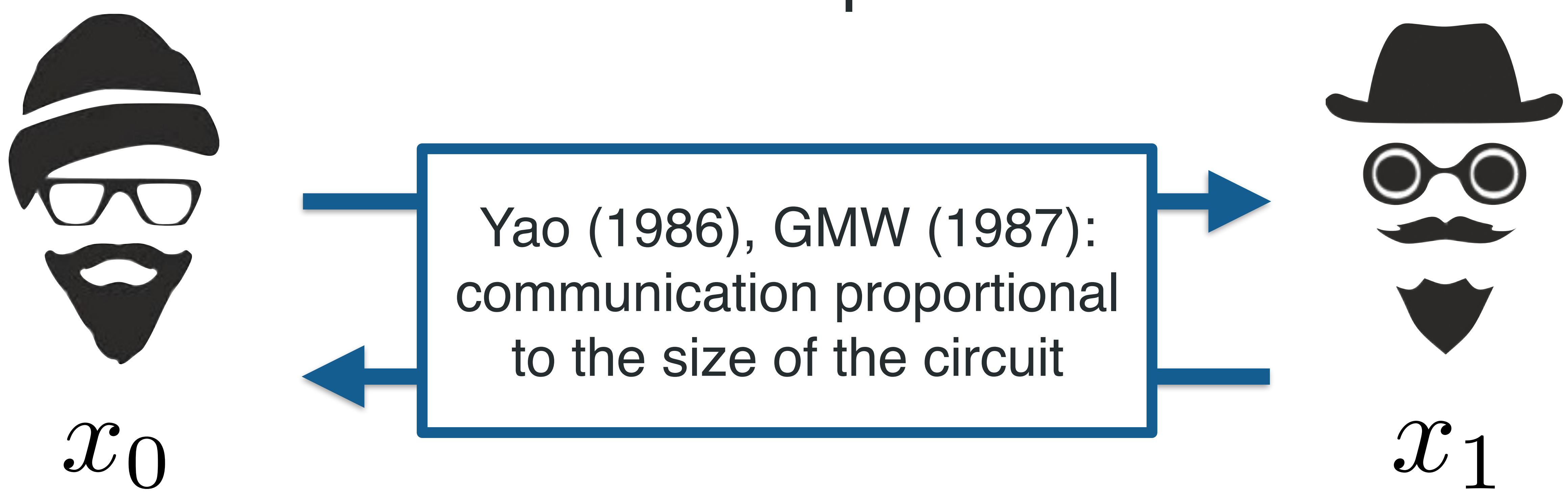
Insecure computation:



$$y = f(x_0, x_1)$$

The Quest for MPC with Low Communication

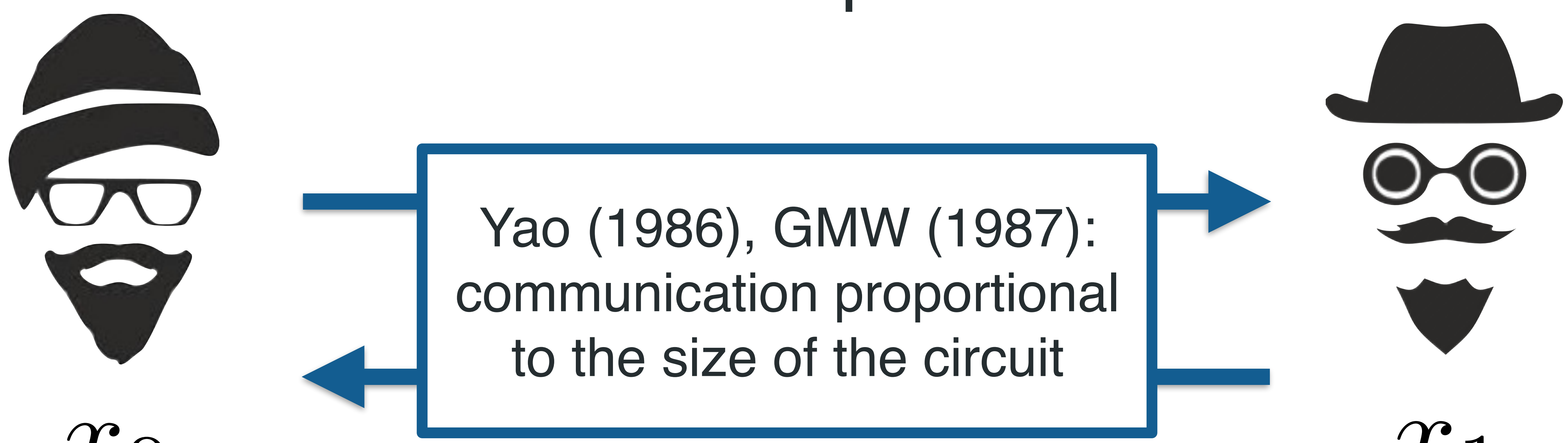
Secure computation:



$$y = f(x_0, x_1)$$

The Quest for MPC with Low Communication

Secure computation:



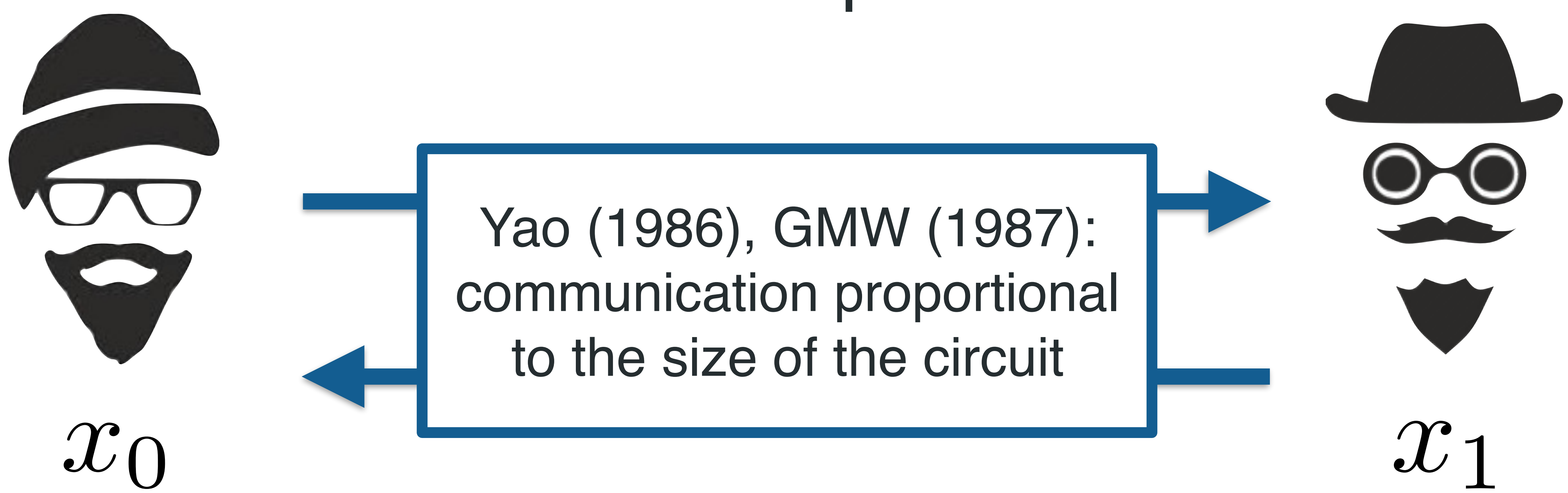
x_0 x_1

$$y = f(x_0, x_1)$$

Does secure computation inherently require so much communication?

The Quest for MPC with Low Communication

Secure computation:



$$y = f(x_0, x_1)$$

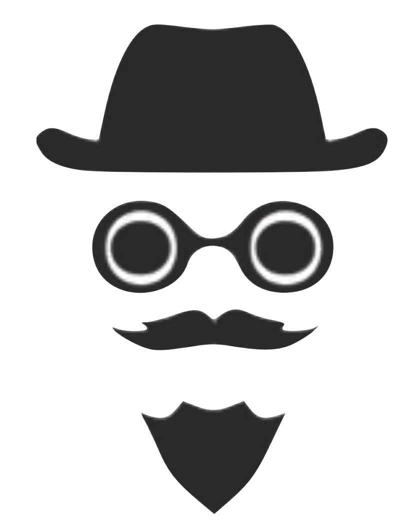
Does secure computation inherently require so much communication?

This work: revisiting this question for MPC with correlated randomness

MPC with Correlated Randomness

Generates and distributes correlated random coins, independent of the inputs of the parties

x_0

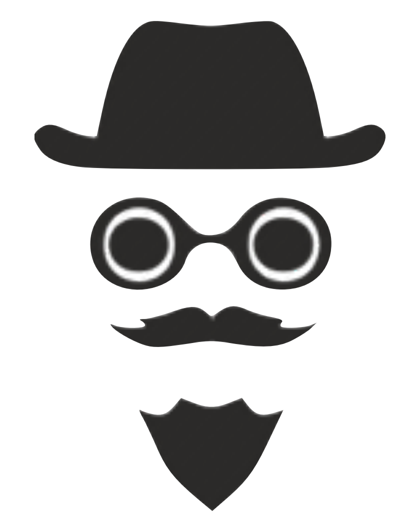


x_1

MPC with Correlated Randomness

Generates and distributes correlated random coins,
independent of the inputs of the parties

x_0

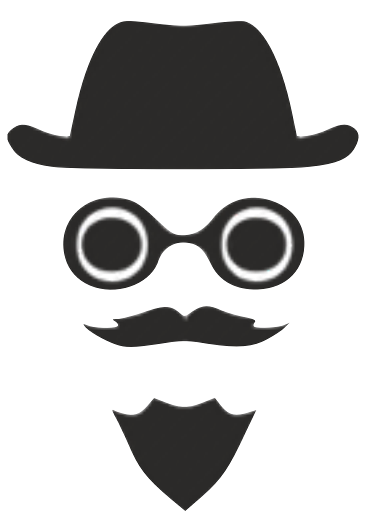


x_1

MPC with Correlated Randomness

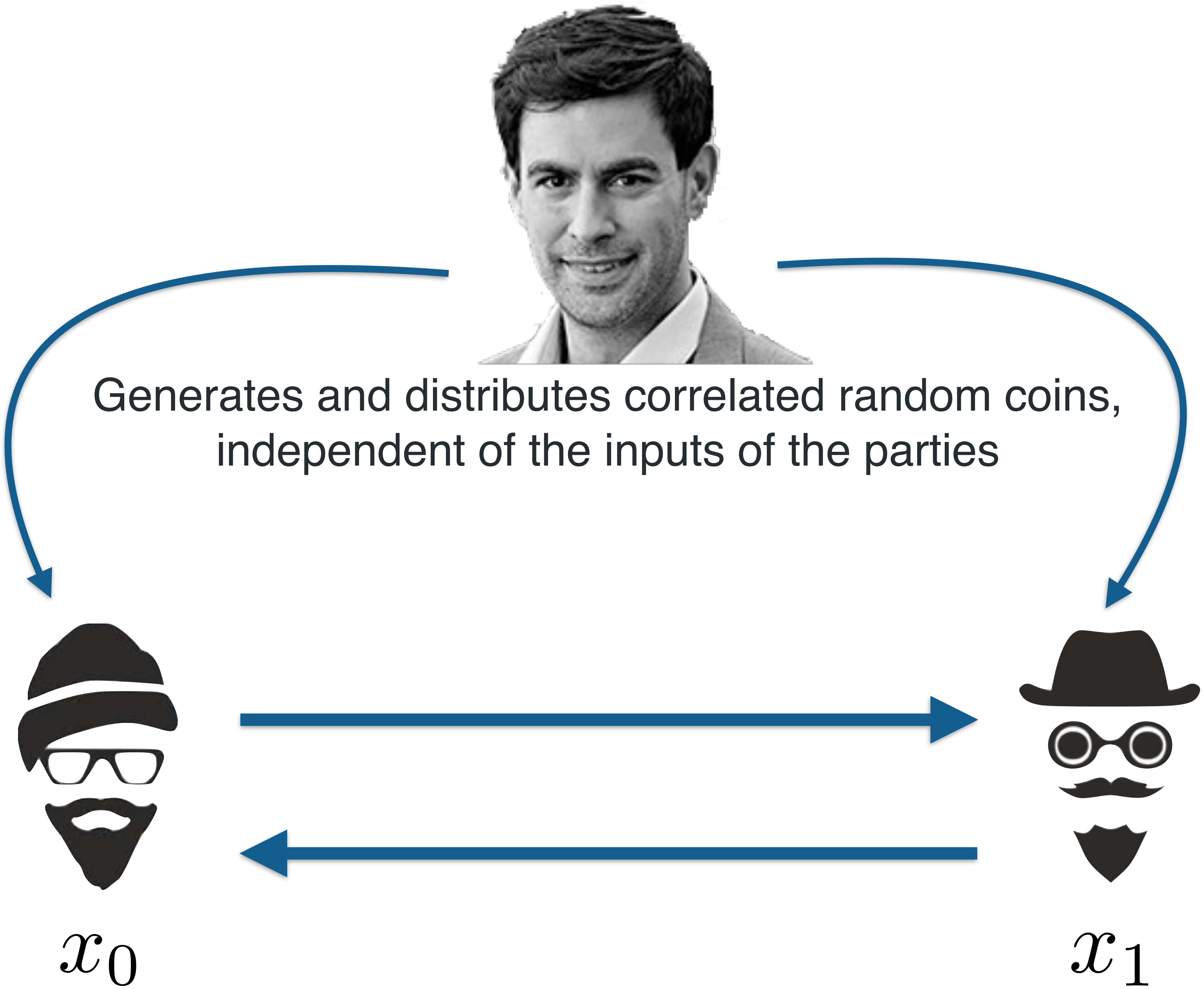
Generates and distributes correlated random coins,
independent of the inputs of the parties

x_0



x_1

MPC with Correlated Randomness

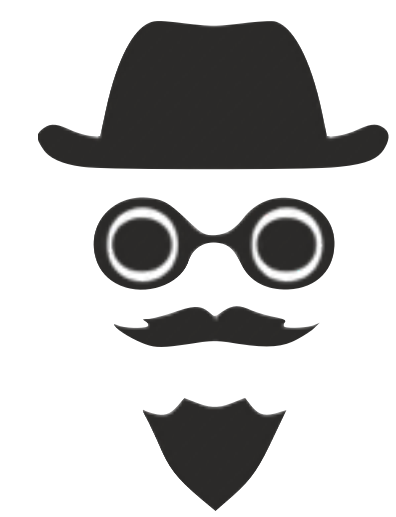


MPC with Correlated Randomness

Generates and distributes correlated random coins, independent of the inputs of the parties

x_0

Beaver (1991): this allows for information-theoretically secure MPC in the online phase



x_1

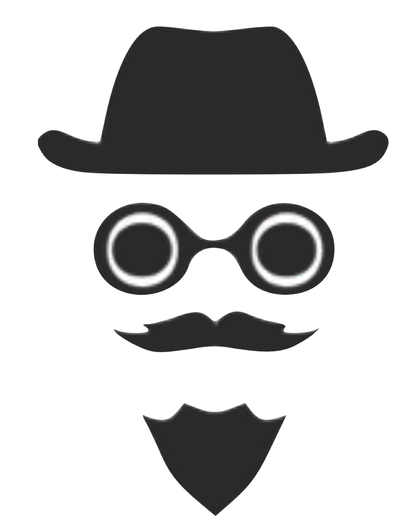
MPC with Correlated Randomness

Generates and distributes correlated random coins, independent of the inputs of the parties

x_0

Beaver (1991): this allows for information-theoretically secure MPC in the online phase

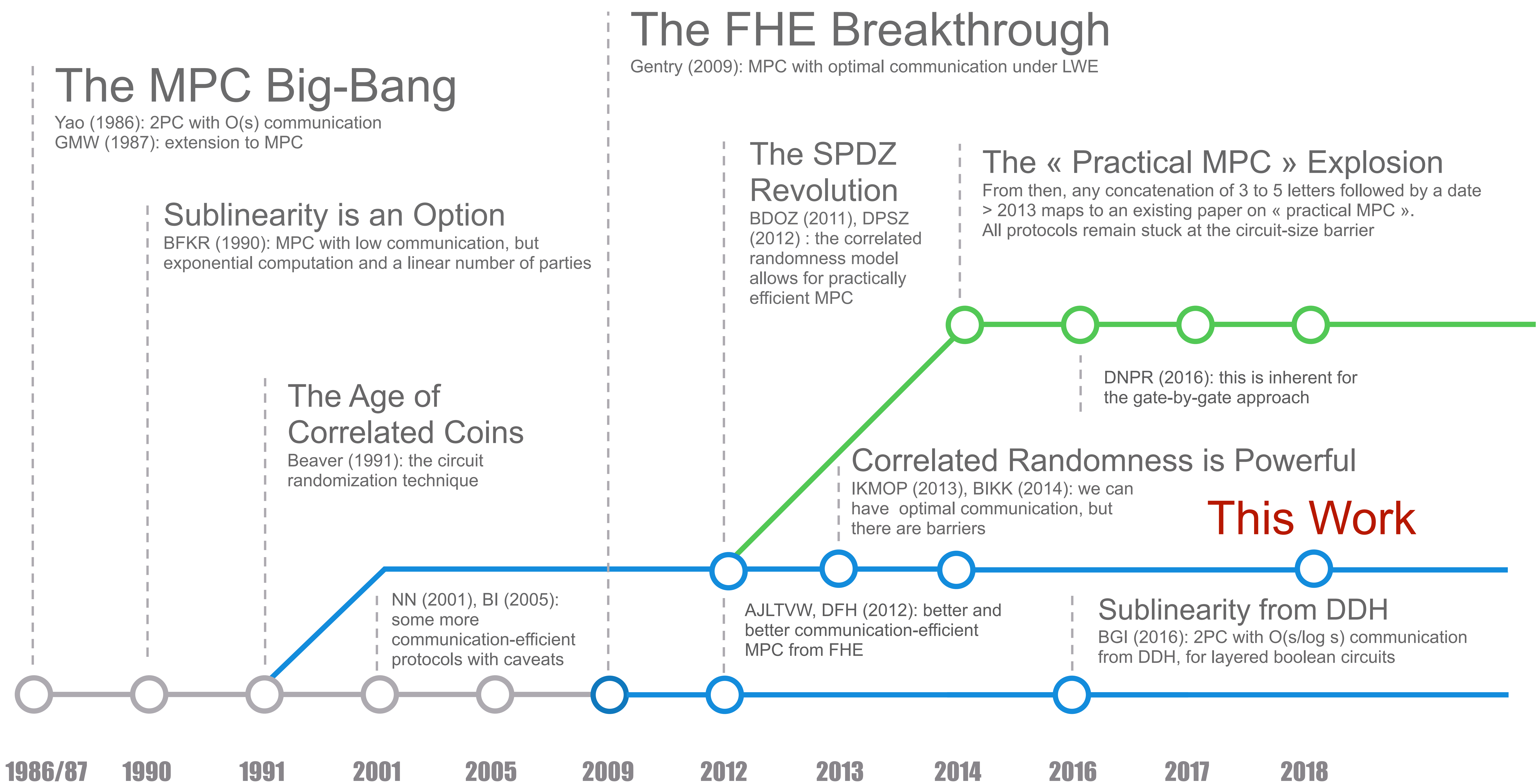
[too many papers to cite them all] (2011 - 2018): this allows for concretely efficient MPC



x_1

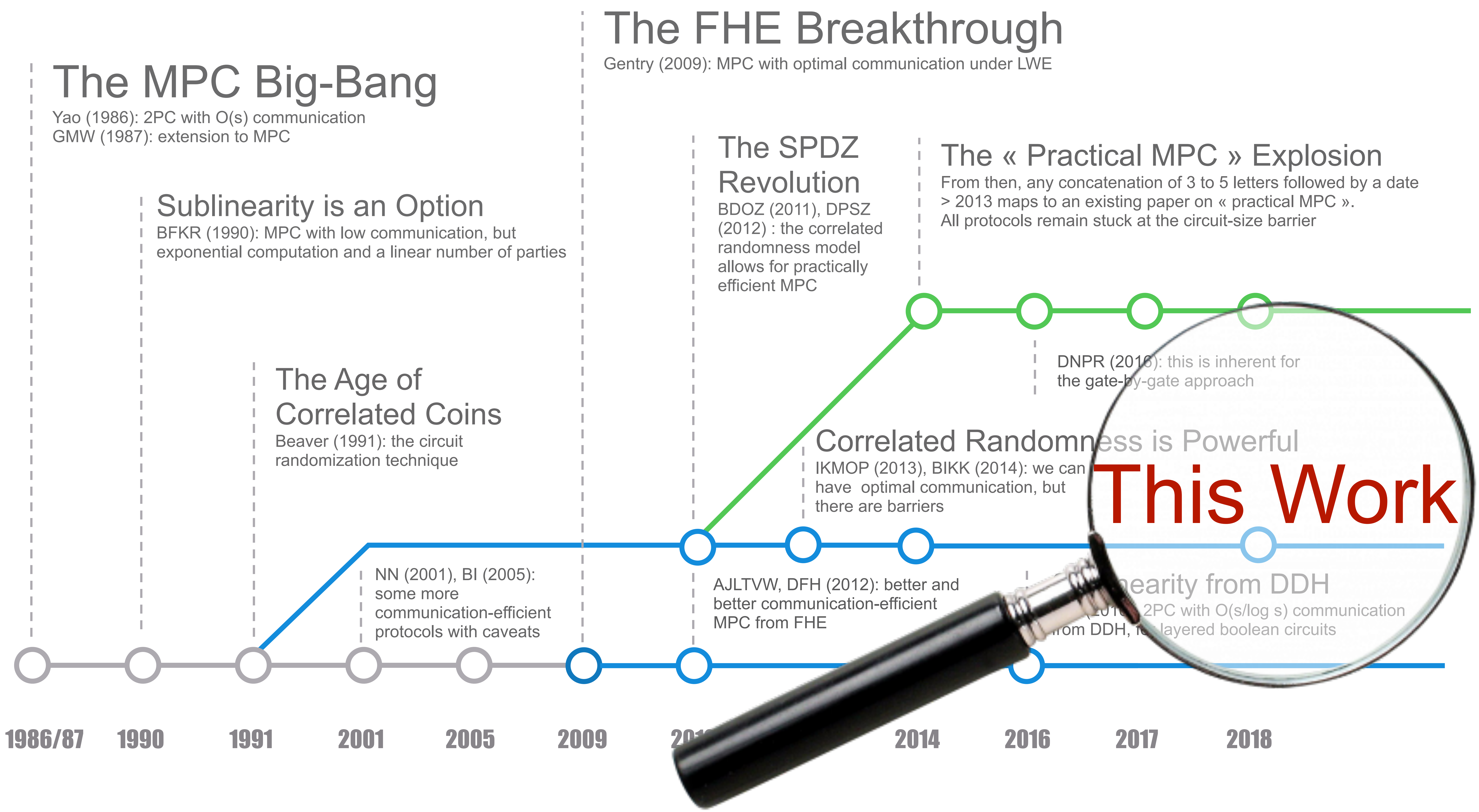
Pushing the Communication Barrier - Timeline

Enter subtitle information text



Pushing the Communication Barrier - Timeline

Enter subtitle information text



Our Result

For any layered boolean circuit C of size s with n inputs and m outputs, there exists an N-party protocol which securely evaluates C in the (function-dependent) correlated randomness model against malicious parties, with adaptive security, and without honest majority, using a polynomial number of correlated random coins and with communication

$$O\left(n + N \cdot \left(m + \frac{s}{\log \log s}\right)\right).$$

Our Result

For any layered boolean circuit C of size s with n inputs and m outputs, there exists an N-party protocol which securely evaluates C in the (function-dependent) correlated randomness model against malicious parties, with adaptive security, and without honest majority, using a polynomial number of correlated random coins and with communication

$$O\left(n + N \cdot \left(m + \frac{s}{\log \log s}\right)\right).$$

+ Extensions to arithmetic circuits, function-independent preprocessing, and tall-and-skinny circuits

Our Result

For any layered boolean circuit C of size s with n inputs and m outputs, there exists an N-party protocol which securely evaluates C in the (function-dependent) correlated randomness model against malicious parties, with adaptive security, and without honest majority, using a polynomial number of correlated random coins and with communication

$$O\left(n + N \cdot \left(m + \frac{s}{\log \log s}\right)\right).$$

+ Extensions to arithmetic circuits, function-independent preprocessing, and tall-and-skinny circuits

We'll focus on 2 parties & semi-honest security here

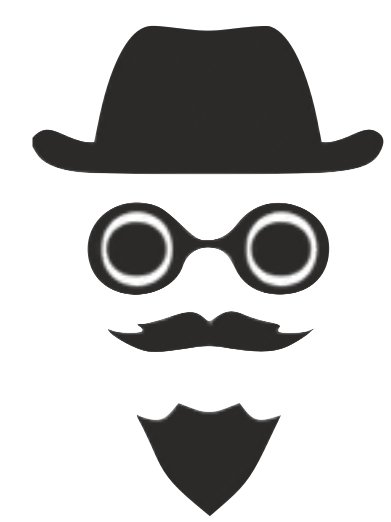
Sharing Truth-Table Correlations

$$f(x) = f(x_0 + x_1)$$

$M =$

f(0)	f(1)	f(2)	f(3)	f(4)	f(5)	...	...	...	...	f(N-5)	f(N-4)	f(N-3)	f(N-2)	f(N-1)	f(N)

x_0



x_1

Sharing Truth-Table Correlations

$$f(x) = f(x_0 + x_1)$$

$M =$

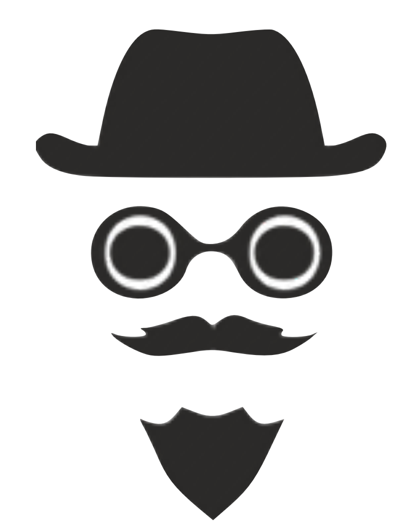
f(0)	f(1)	f(2)	f(3)	f(4)	f(5)	...	...	...	...	f(N-5)	f(N-4)	f(N-3)	f(N-2)	f(N-1)	f(N)

r

picks a random offset

$r = r_0 + r_1$

x_0



x_1

Sharing Truth-Table Correlations

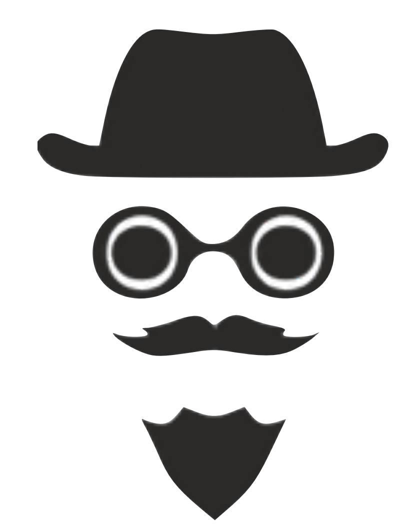
$$f(x + r) = f((x_0 + r_0) + (x_1 + r_1))$$

$M' =$

...	f(N-5)	f(N-4)	f(N-3)	f(N-2)	f(N-1)	f(N)	f(0)	f(1)	f(2)	f(3)	f(4)	f(5)	...	...	...

picks a random offset
$r = r_0 + r_1$

x_0



x_1

Sharing Truth-Table Correlations

$$f(x + r) = f((x_0 + r_0) + (x_1 + r_1))$$

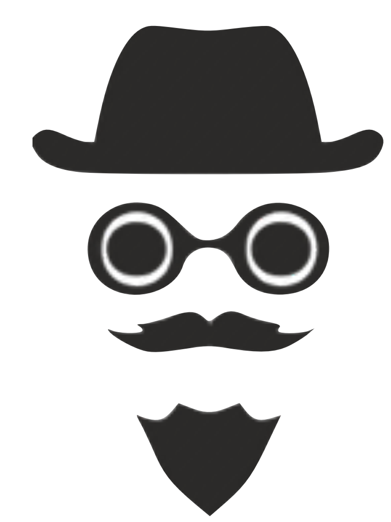
$M' =$

...	f(N-5)	f(N-4)	f(N-3)	f(N-2)	f(N-1)	f(N)	f(0)	f(1)	f(2)	f(3)	f(4)	f(5)	...	...	...

picks a random offset
$r = r_0 + r_1$

shares M’ into
$M' = M'_0 + M'_1$

x_0



x_1

Sharing Truth-Table Correlations

$$f(x+r) = f((x_0 + r_0) + (x_1 + r_1))$$

$M' =$

...	f(N-5)	f(N-4)	f(N-3)	f(N-2)	f(N-1)	f(N)	f(0)	f(1)	f(2)	f(3)	f(4)	f(5)	...	...	...

(r_0, M'_0)

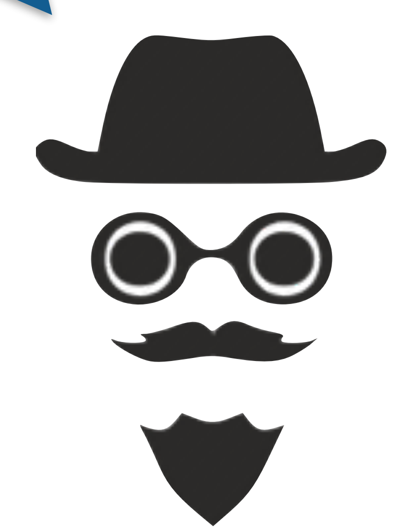
(r_1, M'_1)

picks a random offset
$r = r_0 + r_1$

shares M' into
$M' = M'_0 + M'_1$

x_0

(r_0, M'_0)



x_1

(r_1, M'_1)

Sharing Truth-Table Correlations

$$f(x+r) = f((x_0+r_0)+(x_1+r_1))$$

$M' =$

...	f(N-5)	f(N-4)	f(N-3)	f(N-2)	f(N-1)	f(N)	f(0)	f(1)	f(2)	f(3)	f(4)	f(5)	...	...	...

My job here is done, I can go back to fixing the simple OT protocol.

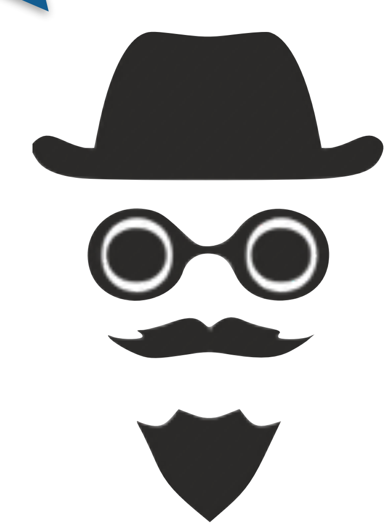
(r_0, M'_0) (r_1, M'_1)

picks a random offset
$r = r_0 + r_1$

shares M' into
$M' = M'_0 + M'_1$

x_0

(r_0, M'_0)



x_1

(r_1, M'_1)

Sharing Truth-Table Correlations

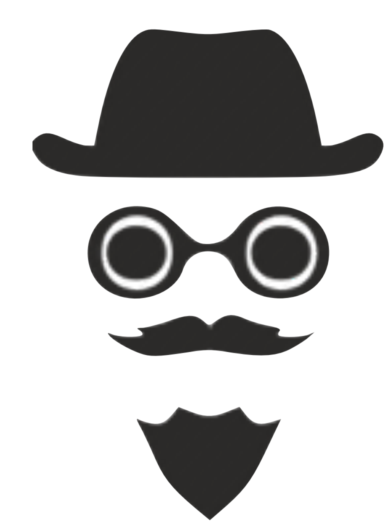
$$f(x+r) = f((x_0 + r_0) + (x_1 + r_1))$$

$M' =$

...	f(N-5)	f(N-4)	f(N-3)	f(N-2)	f(N-1)	f(N)	f(0)	f(1)	f(2)	f(3)	f(4)	f(5)	...	...	...

x_0

(r_0, M'_0)



x_1

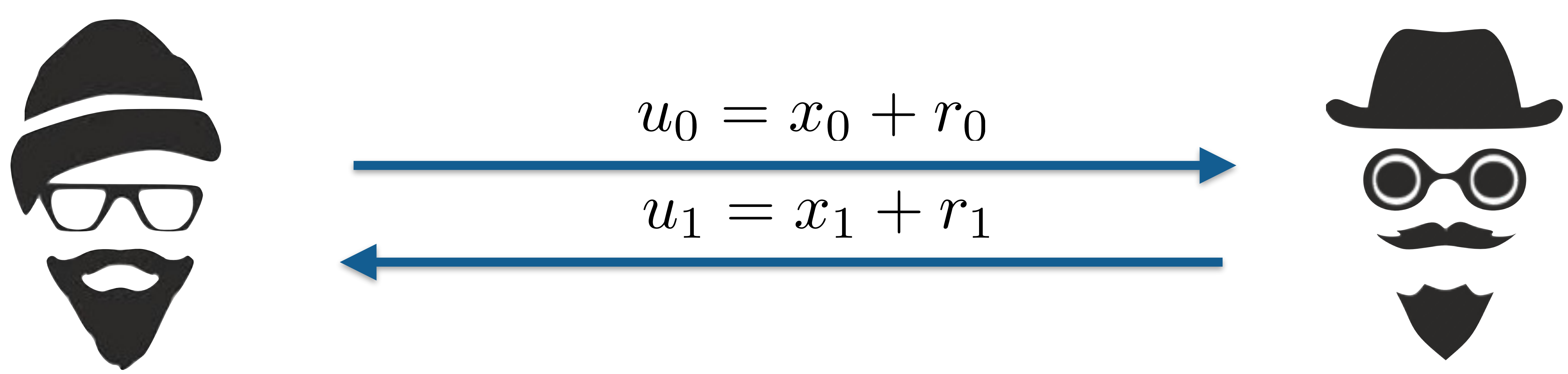
(r_1, M'_1)

Sharing Truth-Table Correlations

$$f(x + r) = f((x_0 + r_0) + (x_1 + r_1))$$

$M' =$

...	f(N-5)	f(N-4)	f(N-3)	f(N-2)	f(N-1)	f(N)	f(0)	f(1)	f(2)	f(3)	f(4)	f(5)	...	...	...



x_0 (r_0, M'_0)

x_1 (r_1, M'_1)

Sharing Truth-Table Correlations

$$f(x+r) = f((x_0 + r_0) + (x_1 + r_1))$$

$M' =$

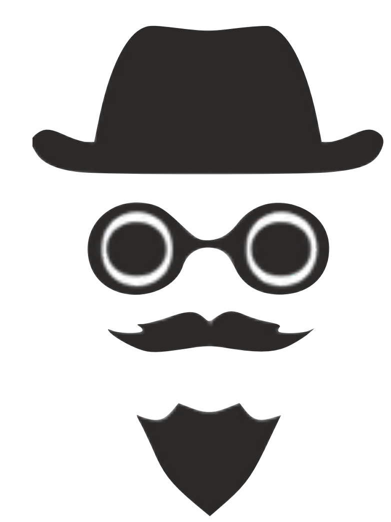
...	f(N-5)	f(N-4)	f(N-3)	f(N-2)	f(N-1)	f(N)	f(0)	f(1)	f(2)	f(3)	f(4)	f(5)	...	...	...

$y_0 \leftarrow M_0'|_{u_0+u_1}$

$y_1 \leftarrow M_1'|_{u_0+u_1}$

$u_0 = x_0 + r_0$

$u_1 = x_1 + r_1$



x_0

(r_0, M_0')

$$y_0 + y_1 = M'|_{x+r} = f(x)$$

x_1

(r_1, M_1')

Sharing Truth-Table Correlations

$$f(x + r) = f((x_0 + r_0) + (x_1 + r_1))$$

$M' =$

...	f(N-5)	f(N-4)	f(N-3)	f(N-2)	f(N-1)	f(N)	f(0)	f(1)	f(2)	f(3)	f(4)	f(5)	...	...	...

communication: $2n$

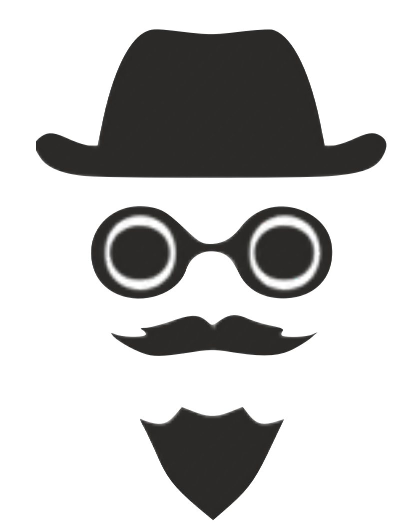
storage: $m \cdot 2^n + n$

$y_0 \leftarrow M'_0|_{u_0+u_1}$

$y_1 \leftarrow M'_1|_{u_0+u_1}$

$u_0 = x_0 + r_0$ →

← $u_1 = x_1 + r_1$



x_0

(r_0, M'_0)

$$y_0 + y_1 = M'|_{x+r} = f(x)$$

x_1

(r_1, M'_1)

Sharing Truth-Table Correlations

$$f(x+r) = f((x_0 + r_0) + (x_1 + r_1))$$

$M' =$

...	f(N-5)	f(N-4)	f(N-3)	f(N-2)	f(N-1)	f(N)	f(0)	f(1)	f(2)	f(3)	f(4)	f(5)	...	...	...

that's great

communication: $2n$

storage: $m \cdot 2^n + n$

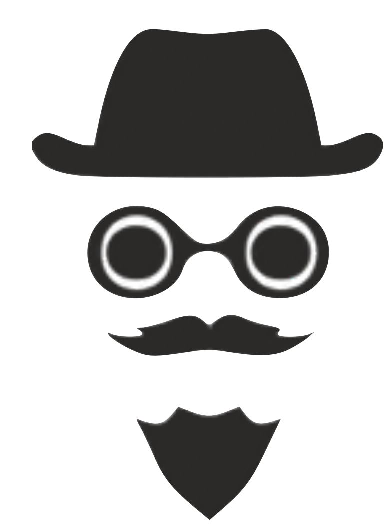
that's bad

$y_0 \leftarrow M'_0|_{u_0+u_1}$

$y_1 \leftarrow M'_1|_{u_0+u_1}$

$u_0 = x_0 + r_0$

$u_1 = x_1 + r_1$



x_0

(r_0, M'_0)

$y_0 + y_1 = M'|_{x+r} = f(x)$

x_1

(r_1, M'_1)

Sharing Truth-Table Correlations

$$f(x+r) = f((x_0 + r_0) + (x_1 + r_1))$$

$M' =$

...	f(N-5)	f(N-4)	f(N-3)	f(N-2)	f(N-1)	f(N)	f(0)	f(1)	f(2)	f(3)	f(4)	f(5)	...	...	...

that's great

that's bad

communication: $2n$

storage: $m \cdot 2^n + n$

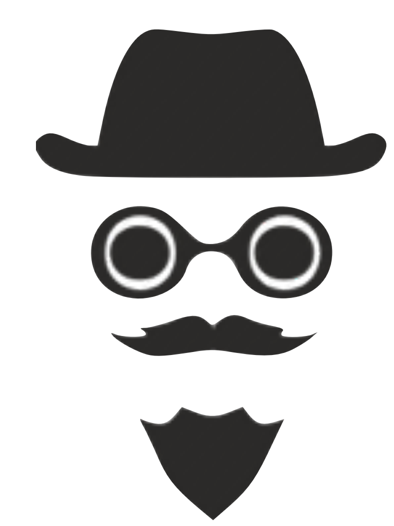
IKMOP (2013): a polynomial storage for all functions would imply a breakthrough in information-theoretic PIR

$y_0 \leftarrow M'_0|_{u_0+u_1}$

$y_1 \leftarrow M'_1|_{u_0+u_1}$

$u_0 = x_0 + r_0$

$u_1 = x_1 + r_1$



$$y_0 + y_1 = M'|_{x+r} = f(x)$$

x_0

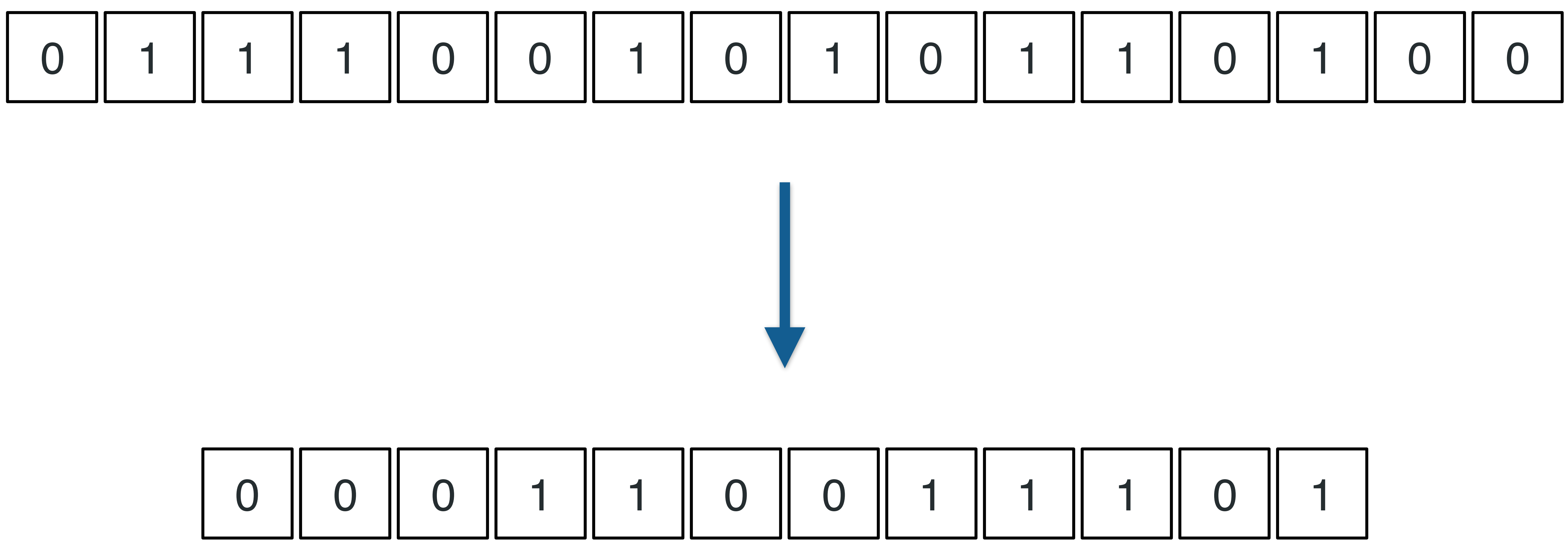
(r_0, M'_0)

x_1

(r_1, M'_1)

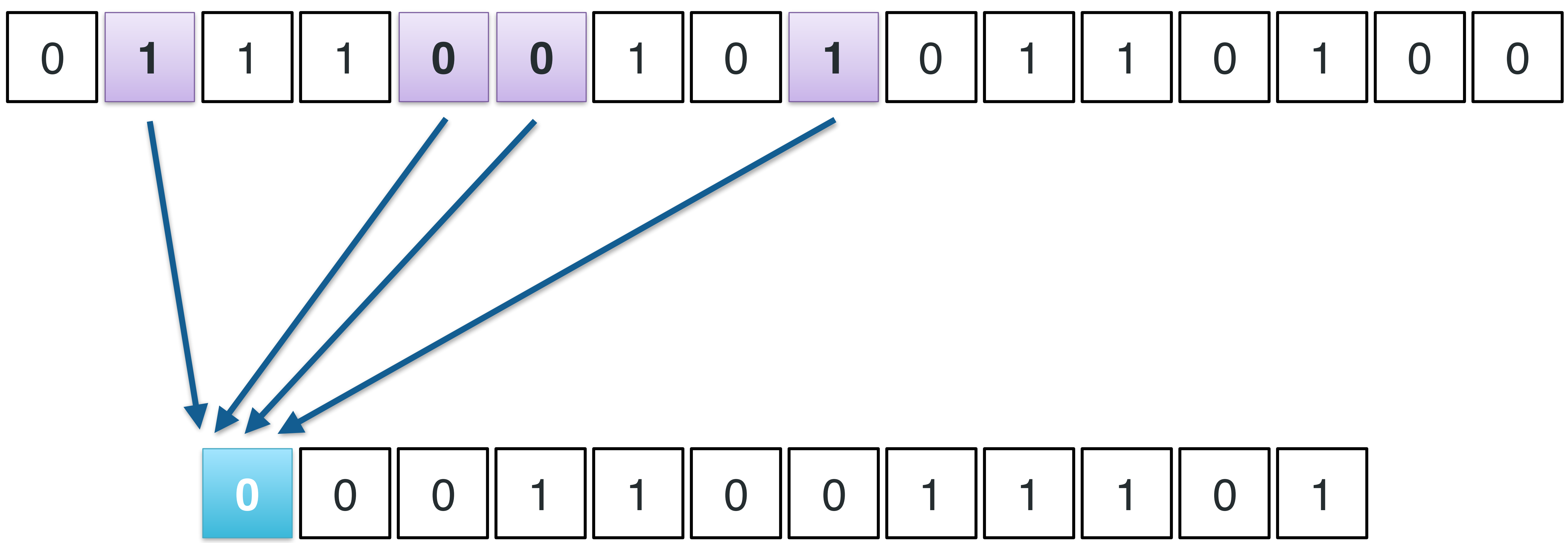
The Core Lemma

Let f be a c-local function, with input of size n and output of size m. Then there exists a protocol Π which securely computes shares of f in the correlated randomness model, with optimal communication $O(n)$ and storage $m \cdot 2^c + n$.



The Core Lemma

Let f be a c-local function, with input of size n and output of size m. Then there exists a protocol Π which securely computes shares of f in the correlated randomness model, with optimal communication $O(n)$ and storage $m \cdot 2^c + n$.



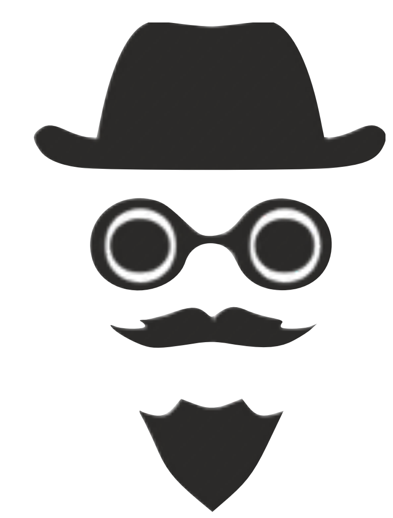
$$f(x) = (f_1(x[S_1]), f_2(x[S_2]), \cdots, f_m(x[S_m]))$$

$$\forall i, \ |S_i| = c$$

The Core Lemma

Let f be a c-local function, with input of size n and output of size m. Then there exists a protocol Π which securely computes shares of f in the correlated randomness model, with optimal communication $O(n)$ and storage $m \cdot 2^c + n$.

x_0



x_1

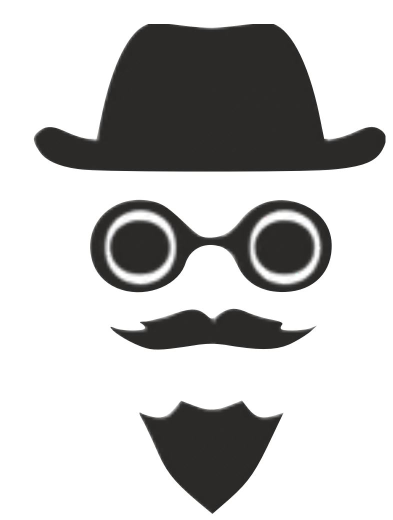
$$f(x) = (f_1(x[S_1]), f_2(x[S_2]), \cdots, f_m(x[S_m]))$$
$$\forall i,\ |S_i| = c$$

The Core Lemma

Let f be a c-local function, with input of size n and output of size m. Then there exists a protocol Π which securely computes shares of f in the correlated randomness model, with optimal communication $O(n)$ and storage $m \cdot 2^c + n$.

$(M_0, M_1, \cdots, M_m)$

x_0

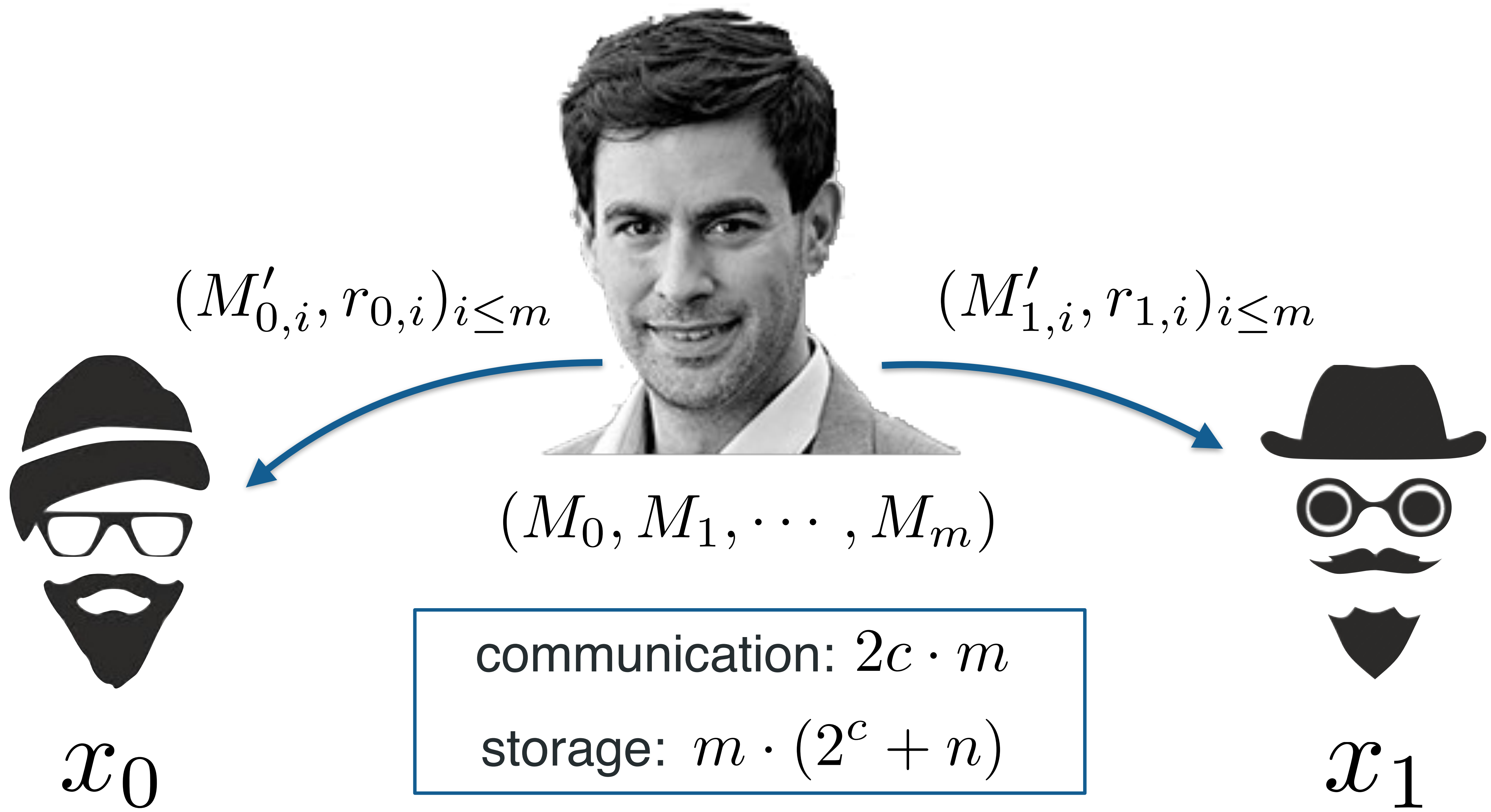


x_1

$$f(x) = (f_1(x[S_1]), f_2(x[S_2]), \cdots, f_m(x[S_m]))$$
$$\forall i,\ |S_i| = c$$

The Core Lemma

Let f be a c-local function, with input of size n and output of size m. Then there exists a protocol Π which securely computes shares of f in the correlated randomness model, with optimal communication $O(n)$ and storage $m \cdot 2^c + n$.



$$f(x) = (f_1(x[S_1]), f_2(x[S_2]), \cdots, f_m(x[S_m]))$$
$$\forall i, \ |S_i| = c$$

The Core Lemma

Let f be a c-local function, with input of size n and output of size m. Then there exists a protocol Π which securely computes shares of f in the correlated randomness model, with optimal communication $O(n)$ and storage $m \cdot 2^c + n$.

$$f(x) = (f_1(x[S_1]), f_2(x[S_2]), \cdots, f_m(x[S_m]))$$

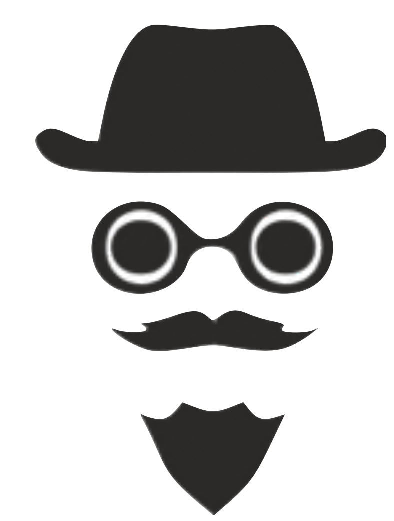
M_1 , M_2 ... M_m

$f_1(1)$	$f_1(2)$	...	$f_1(2^c)$	,	$f_2(1)$	$f_2(2)$	...	$f_2(2^c)$	...	$f_m(1)$	$f_m(2)$	...	$f_m(2^c)$

r_1 r_2 r_m

$$\forall i, \ |r_i| = c$$

$(x_0[s_i] + r_{0,i})_i$



$(x_1[s_i] + r_{1,i})_i$

The Core Lemma

Let f be a c-local function, with input of size n and output of size m. Then there exists a protocol Π which securely computes shares of f in the correlated randomness model, with optimal communication $O(n)$ and storage $m \cdot 2^c + n$.

$$f(x) = (f_1(x[S_1]), f_2(x[S_2]), \cdots, f_m(x[S_m]))$$

M_1 , M_2 ... M_m

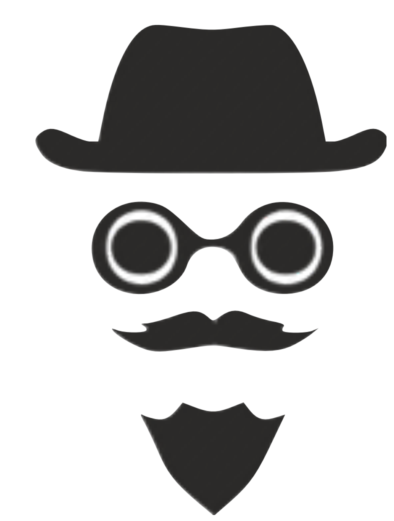
$f_1(1)$	$f_1(2)$	...	$f_1(2^c)$	,	$f_2(1)$	$f_2(2)$	...	$f_2(2^c)$	...	$f_m(1)$	$f_m(2)$	...	$f_m(2^c)$

r_1 r_2 r_m

$$\forall i, \; |r_i| = c$$

$x_0 + r_0$

Idea: pick a single global offset r, and set $r_i \leftarrow r[S_i]$



$x_1 + r_1$

The Core Lemma

Let f be a c-local function, with input of size n and output of size m. Then there exists a protocol Π which securely computes shares of f in the correlated randomness model, with optimal communication $O(n)$ and storage $m \cdot 2^c + n$.

$$f(x) = (f_1(x[S_1]), f_2(x[S_2]), \cdots, f_m(x[S_m]))$$

M_1 , M_2 ... M_m

$f_1(1)$	$f_1(2)$	...	$f_1(2^c)$	,	$f_2(1)$	$f_2(2)$	...	$f_2(2^c)$	...	$f_m(1)$	$f_m(2)$	...	$f_m(2^c)$

r_1 r_2 r_m

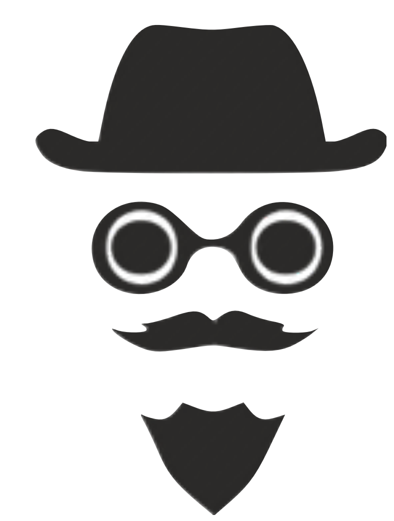
$x_0 + r_0$

$$\forall i, \ |r_i| = c$$

Idea: pick a single global offset r, and set $r_i \leftarrow r[S_i]$

communication: $2n$

storage: $m \cdot 2^c + n$

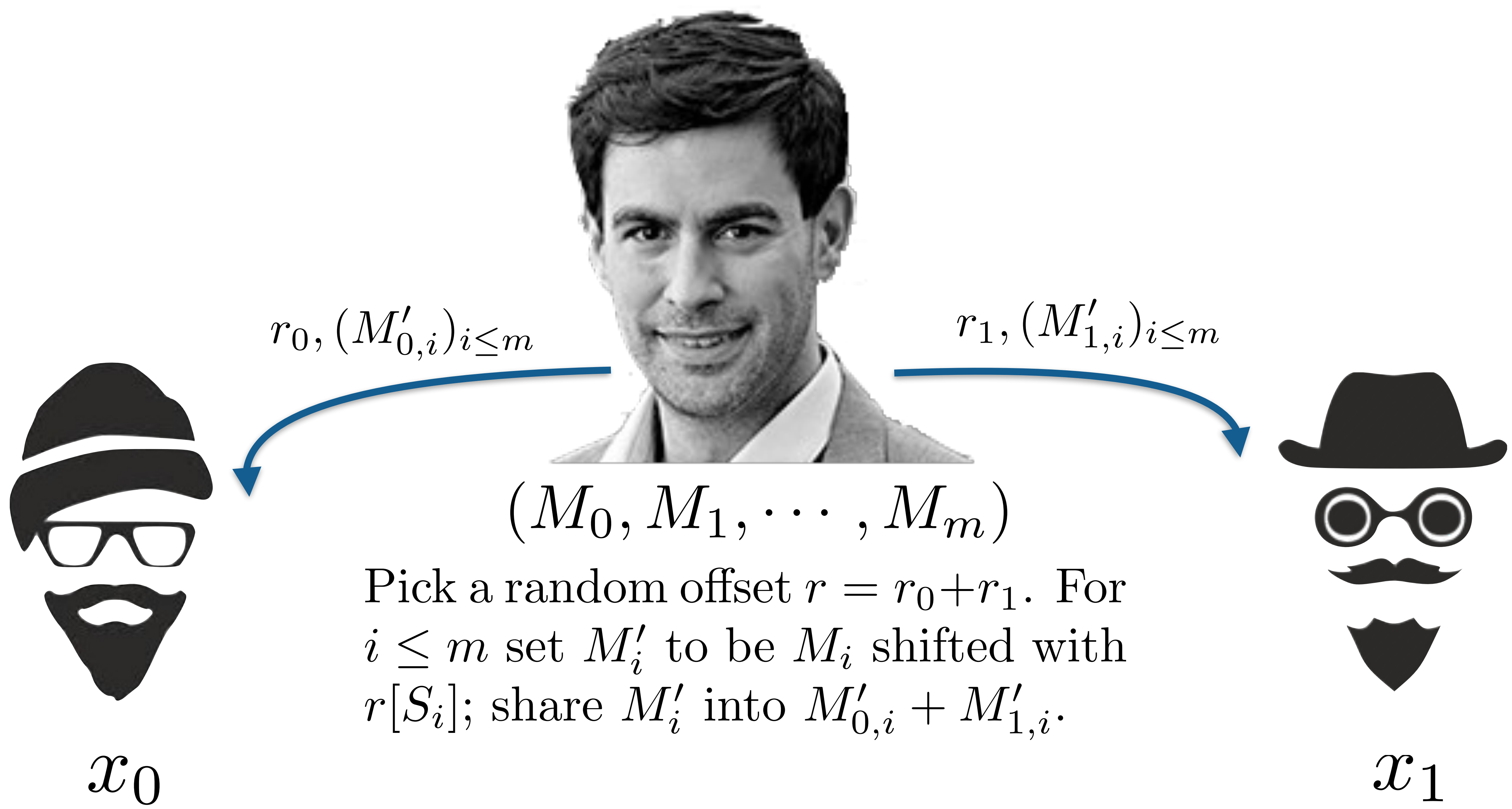


$x_1 + r_1$

The Core Lemma

Let f be a c-local function, with input of size n and output of size m. Then there exists a protocol Π which securely computes shares of f in the correlated randomness model, with optimal communication $O(n)$ and storage $m \cdot 2^c + n$.

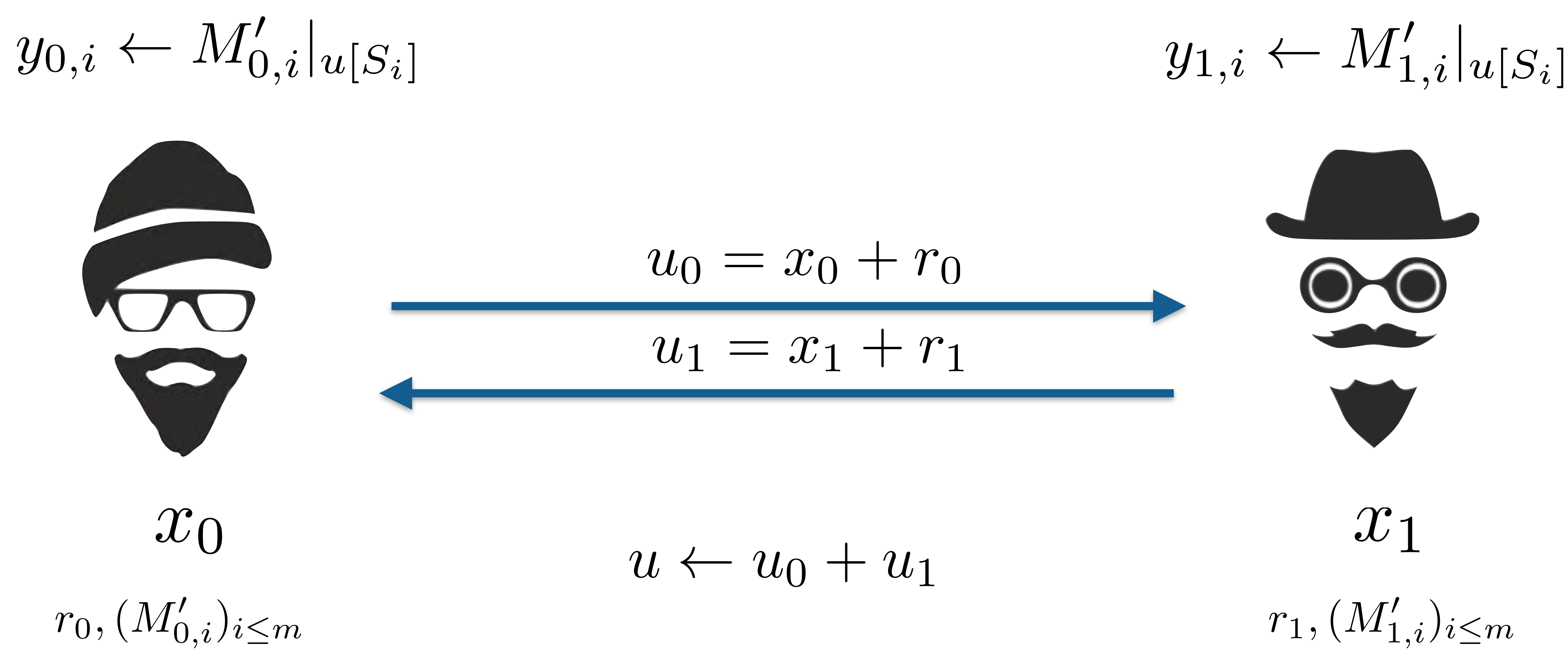
$$f(x) = (f_1(x[S_1]), f_2(x[S_2]), \cdots, f_m(x[S_m]))$$



The Core Lemma

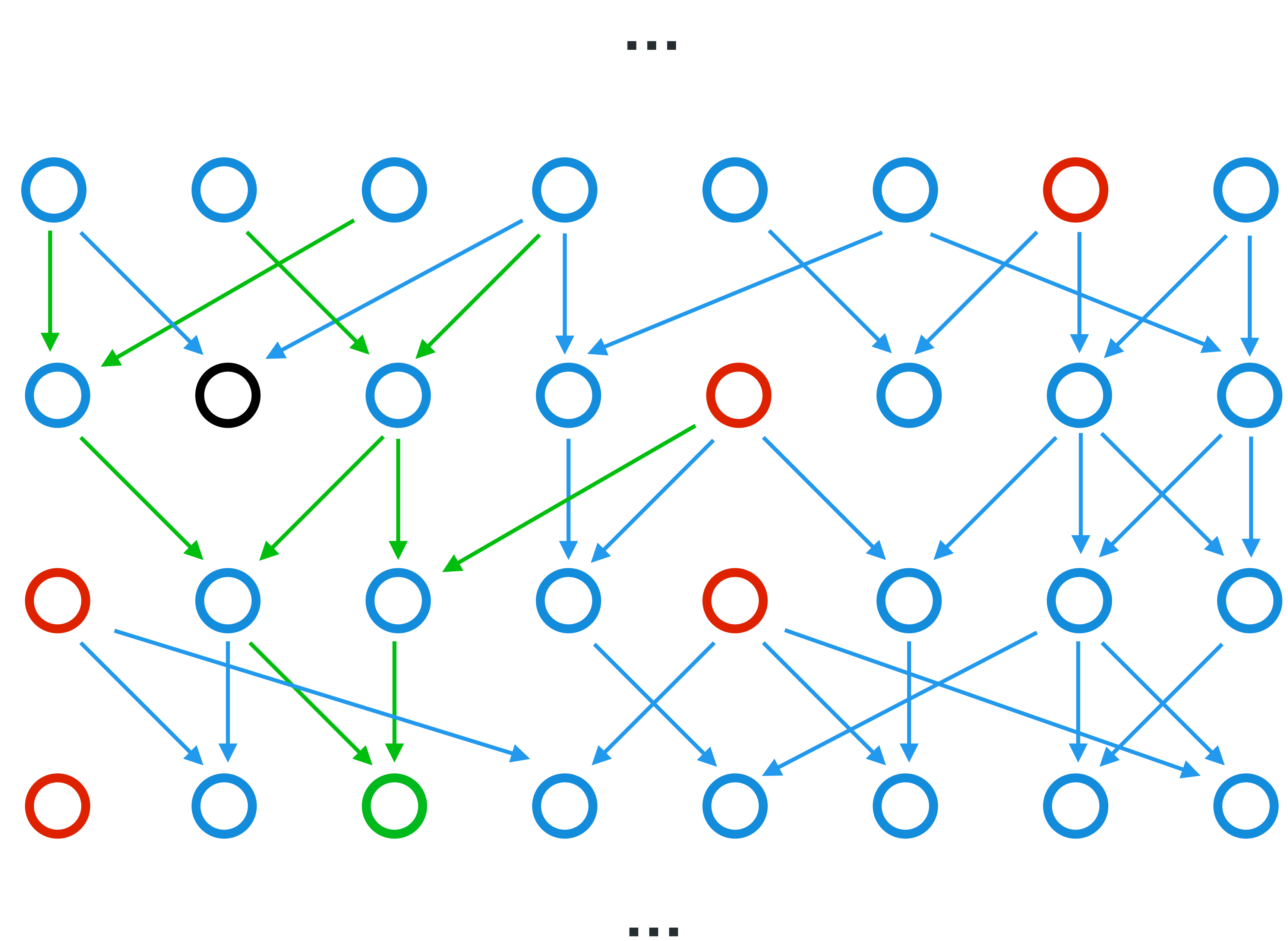
Let f be a c-local function, with input of size n and output of size m. Then there exists a protocol Π which securely computes shares of f in the correlated randomness model, with optimal communication $O(n)$ and storage $m \cdot 2^c + n$.

$$f(x) = (f_1(x[S_1]), f_2(x[S_2]), \cdots, f_m(x[S_m]))$$



Construction

Layered boolean circuit, size s, depth d, width w, n inputs and m outputs



: node

: input node

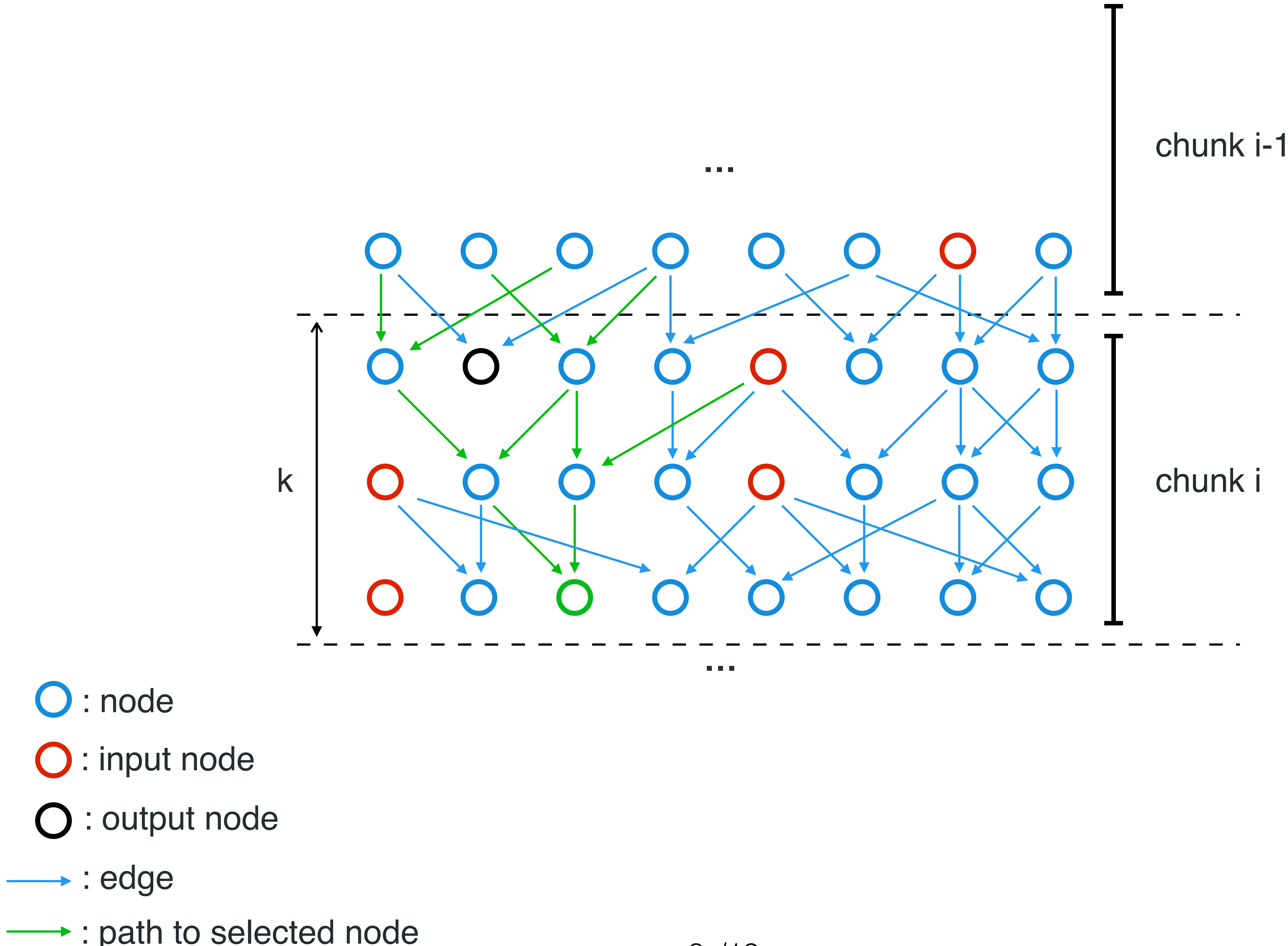
: output node

: edge

: path to selected node

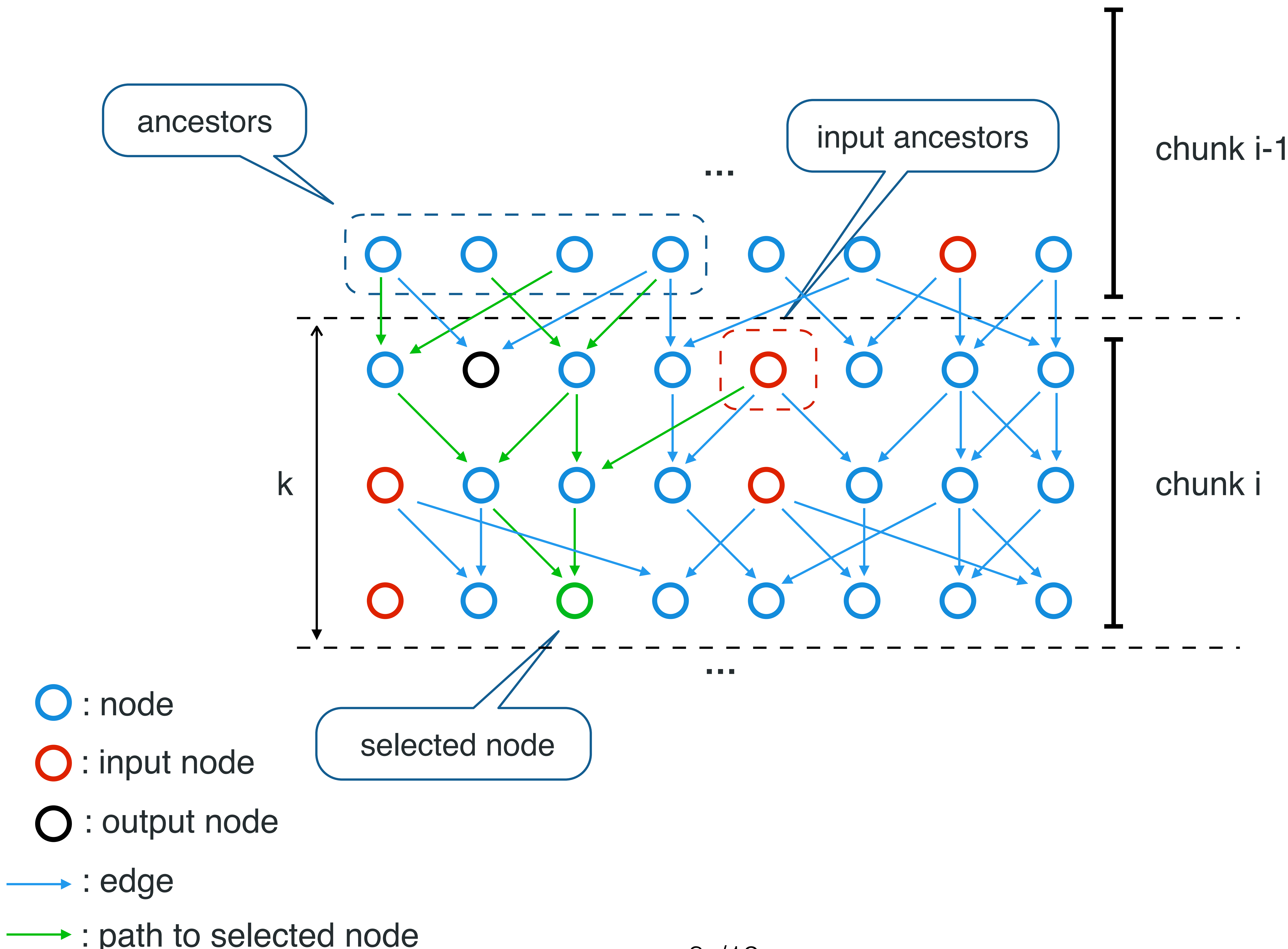
Construction

Layered boolean circuit, size s, depth d, width w, n inputs and m outputs



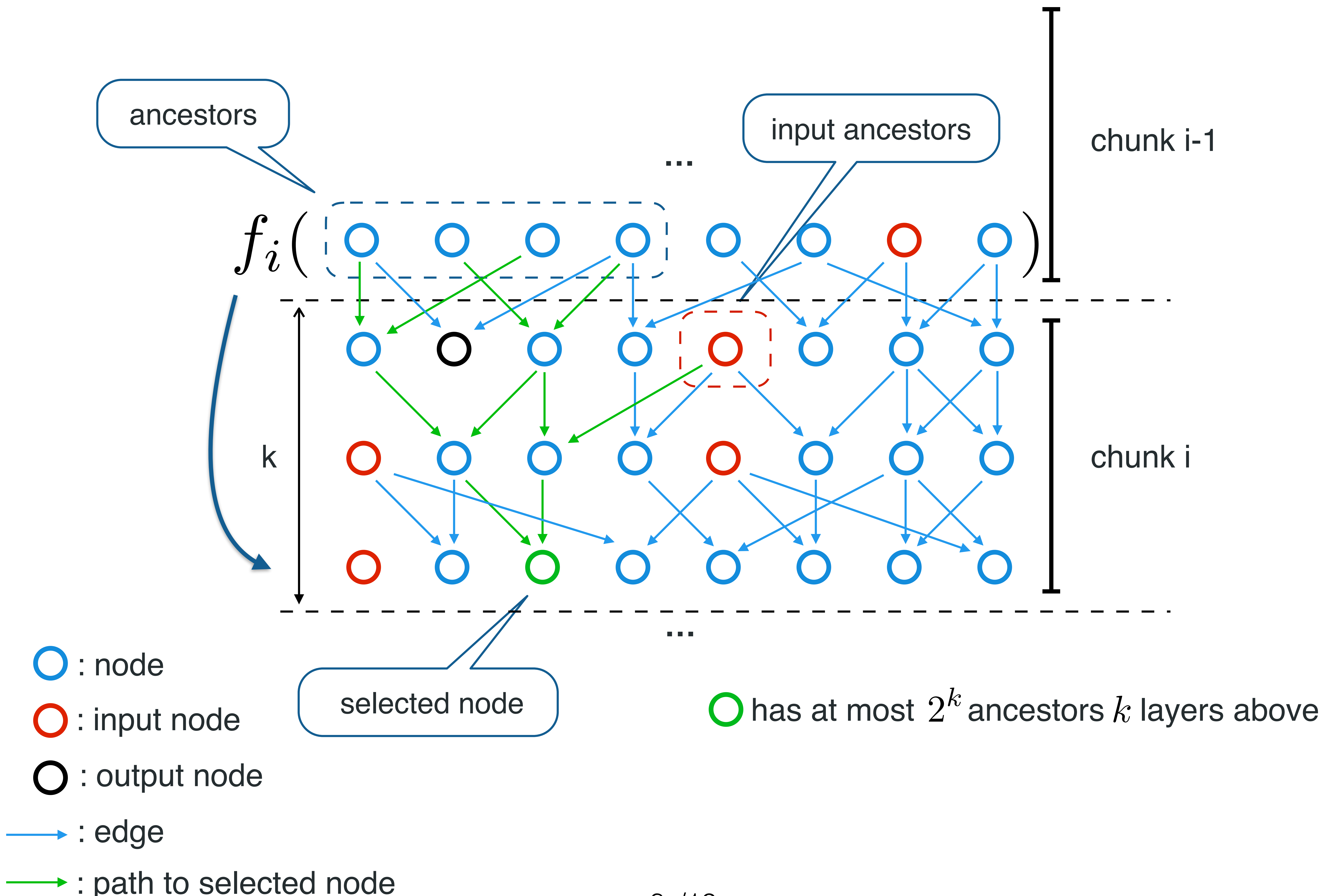
Construction

Layered boolean circuit, size s, depth d, width w, n inputs and m outputs



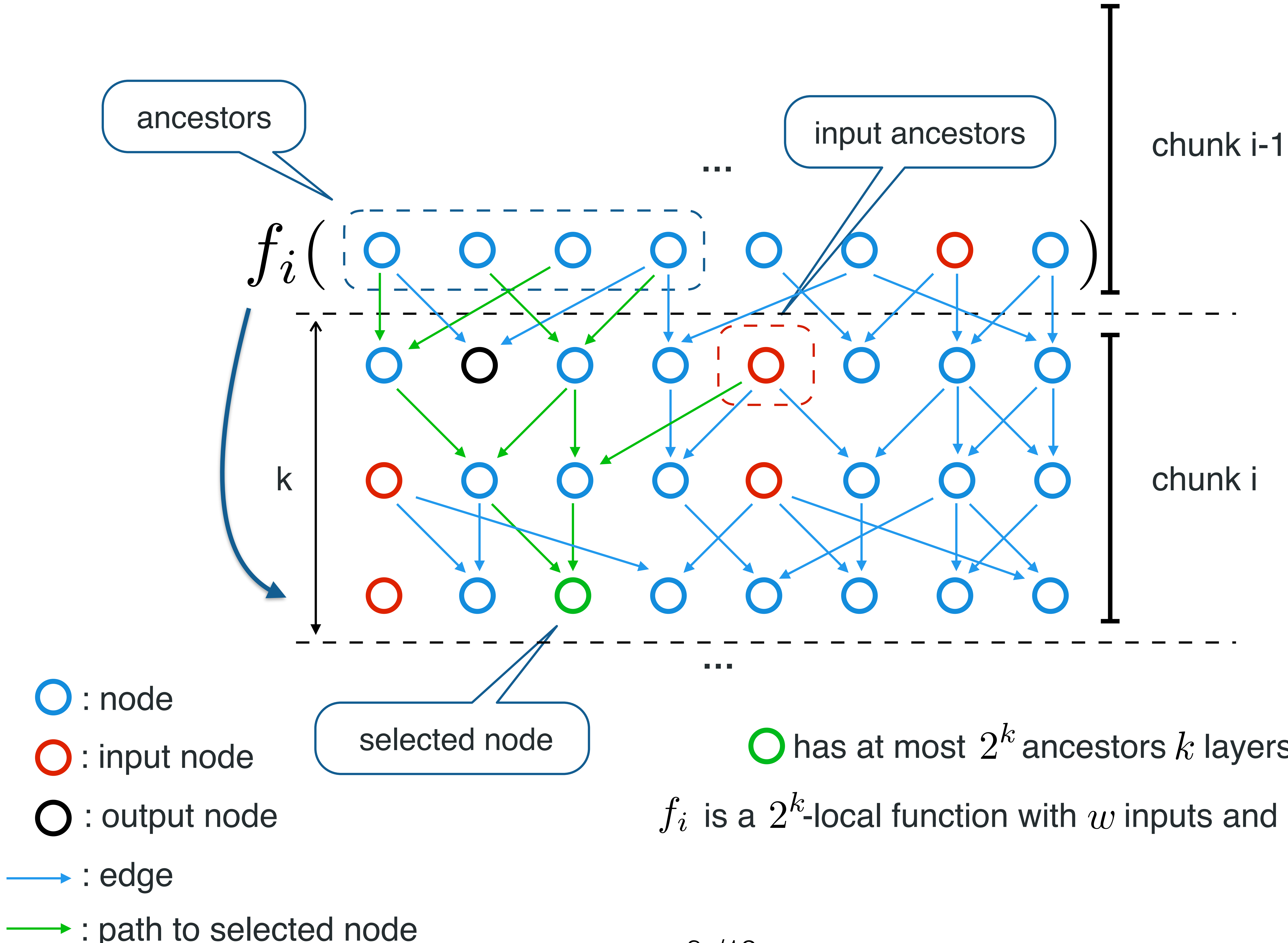
Construction

Layered boolean circuit, size s, depth d, width w, n inputs and m outputs



Construction

Layered boolean circuit, size s, depth d, width w, n inputs and m outputs



has at most 2^k ancestors k layers above

f_i is a 2^k-local function with w inputs and outputs

Construction

Layered boolean circuit, size s, depth d, width w, n inputs and m outputs

Let f be a c-local function, with input of size n and output of size m. Then there exists a protocol Π which securely computes shares of f in the correlated randomness model, with optimal communication $O(n)$ and storage $m \cdot 2^c + n$.

f_i is a 2^k-local function with w inputs and outputs

We can securely compute shares of f_i with communication $O(w)$ and storage $O(w \cdot 2^{2^k})$

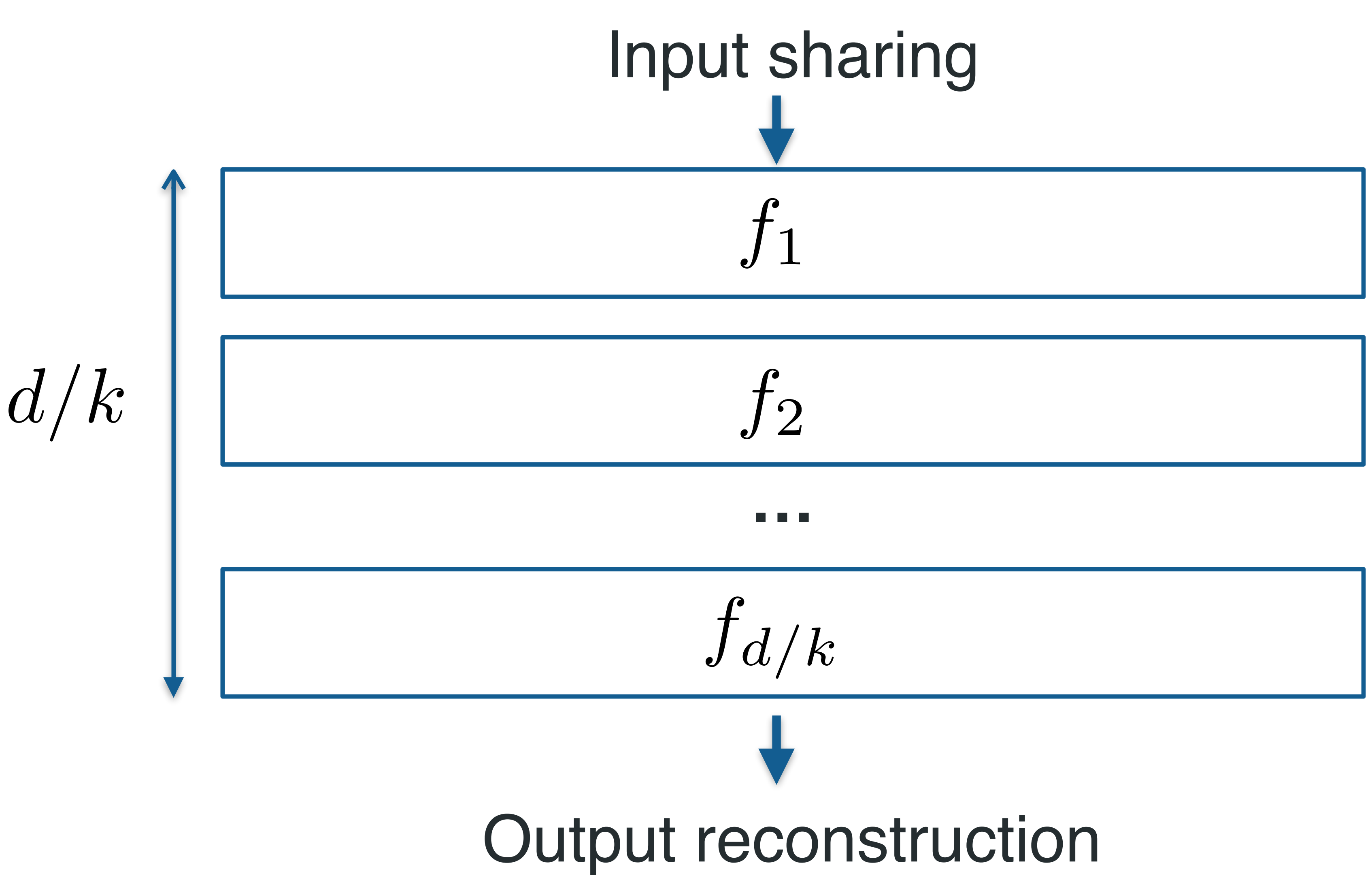
Construction

Layered boolean circuit, size s, depth d, width w, n inputs and m outputs

Let f be a c-local function, with input of size n and output of size m. Then there exists a protocol Π which securely computes shares of f in the correlated randomness model, with optimal communication $O(n)$ and storage $m \cdot 2^c + n$.

f_i is a 2^k-local function with w inputs and outputs

We can securely compute shares of f_i with communication $O(w)$ and storage $O(w \cdot 2^{2^k})$



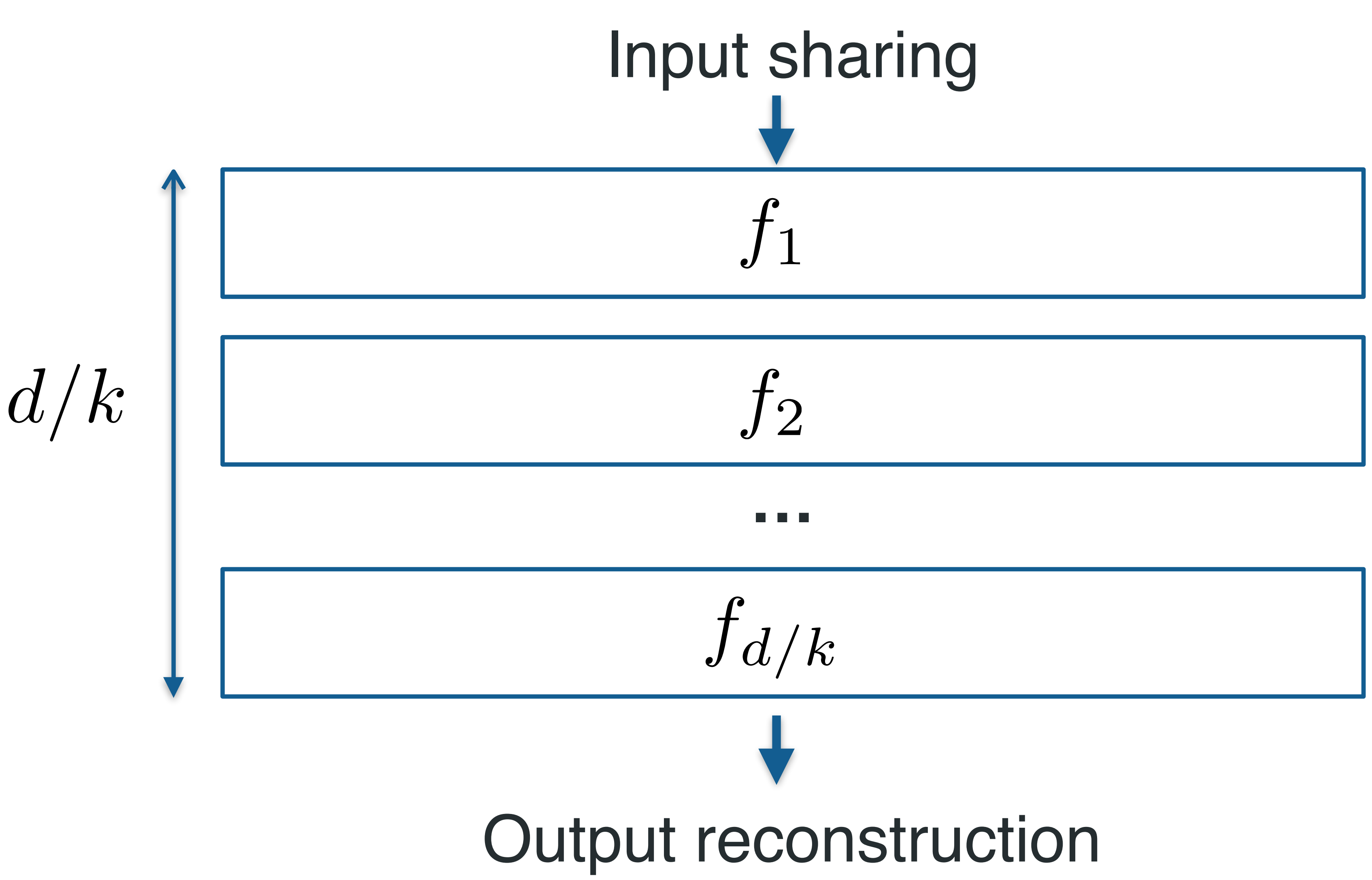
Construction

Layered boolean circuit, size s, depth d, width w, n inputs and m outputs

Let f be a c-local function, with input of size n and output of size m. Then there exists a protocol Π which securely computes shares of f in the correlated randomness model, with optimal communication $O(n)$ and storage $m \cdot 2^c + n$.

f_i is a 2^k-local function with w inputs and outputs

We can securely compute shares of f_i with communication $O(w)$ and storage $O(w \cdot 2^{2^k})$



Communication: $O(w \cdot d/k) = O(s/k)$

Storage: $O(w \cdot 2^{2^k} \cdot d/k) = O(s \cdot 2^{2^k}/k)$

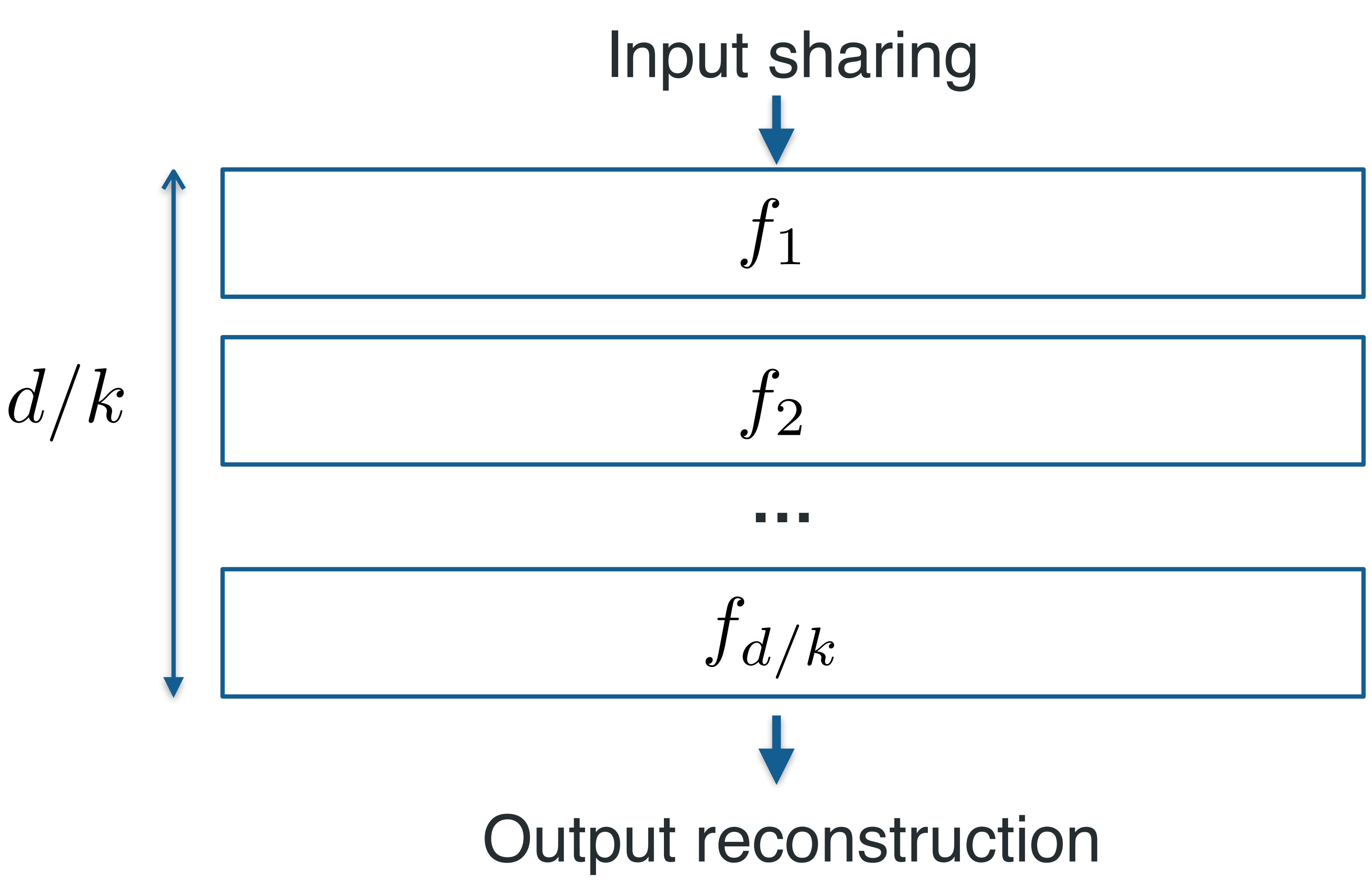
Construction

Layered boolean circuit, size s, depth d, width w, n inputs and m outputs

Let f be a c-local function, with input of size n and output of size m. Then there exists a protocol Π which securely computes shares of f in the correlated randomness model, with optimal communication $O(n)$ and storage $m \cdot 2^c + n$.

f_i is a 2^k-local function with w inputs and outputs

We can securely compute shares of f_i with communication $O(w)$ and storage $O(w \cdot 2^{2^k})$



Communication: $O(w \cdot d/k) = O(s/k)$

Storage: $O(w \cdot 2^{2^k} \cdot d/k) = O(s \cdot 2^{2^k}/k)$

There exist a protocol to evaluate any LBC, with polynomial storage and total communication:

$$O\left(n + m + \frac{s}{\log \log s}\right)$$

Arithmetic Setting

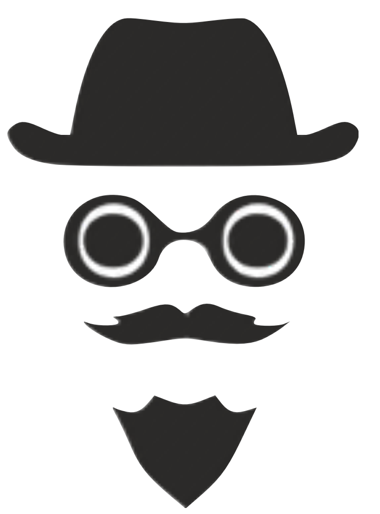
There is a very natural extension of this protocol to arithmetic circuits (apparently, was not observed before)

Idea: replace truth-tables by multivariate polynomials

Arithmetic Setting

$$P(\vec{X})$$

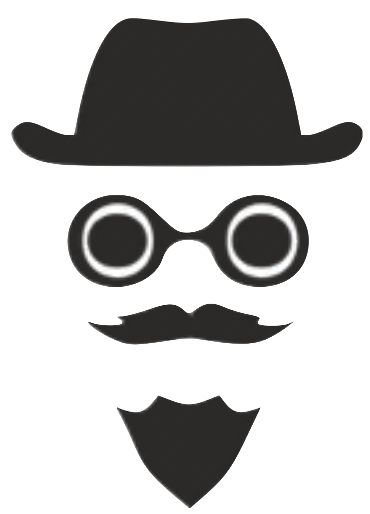
$$\vec{u} = \vec{x} + \vec{r}$$



Arithmetic Setting

$$P(\vec{X})$$

$$\vec{u} = \vec{x} + \vec{r}$$



Arithmetic Setting

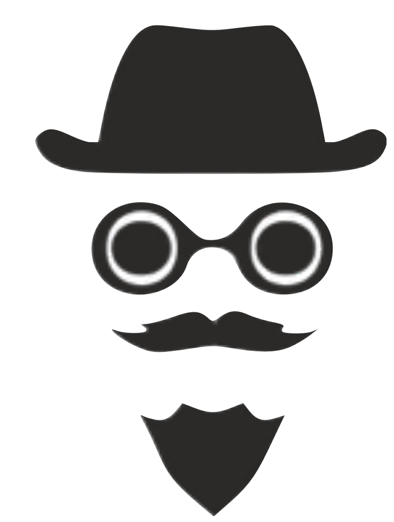
$$P(\vec{X})$$

$$P'_0(\vec{X}) + P'_1(\vec{X}) = P(\vec{X} - \vec{r}) + \vec{s}$$

- -

$$\vec{u} = \vec{x} + \vec{r}$$

$$P'_0(\vec{X})$$



$$P'_1(\vec{X})$$

Arithmetic Setting

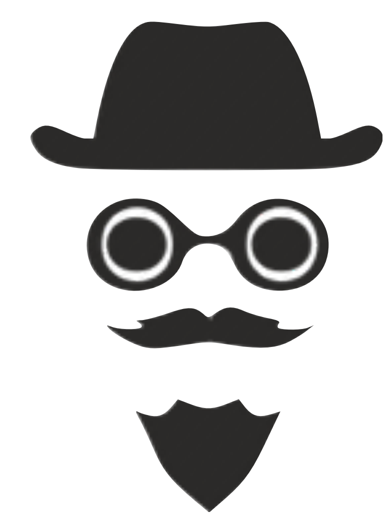
$$P(\vec{X})$$

$$P'_0(\vec{X}) + P'_1(\vec{X}) = P(\vec{X} - \vec{r}) + \vec{s}$$

$$\vec{u} = \vec{x} + \vec{r}$$

$$\vec{v}_0 = P'_0(\vec{u})$$

$$\vec{v}_1 = P'_1(\vec{u})$$



$$P'_0(\vec{X})$$

$$P'_1(\vec{X})$$

$$\vec{v}_0 + \vec{v}_1 = P(\vec{x}) + \vec{s}$$

What to Concretely Take out of that?

What to Concretely Take out of that?

- MPC from truth-table correlations gives great concrete numbers

TinyTable: only 2 bits per AND gate (and 4 bits of storage*), and 0 bit per XOR gates

This work: can get *1 bit* per AND gate in total (amortized) and 0 per XOR gates, at a cost of 8x more storage and 4x more computation

best candidates for concrete efficiency so far?

What to Concretely Take out of that?

- MPC from truth-table correlations gives great concrete numbers

TinyTable: only 2 bits per AND gate (and 4 bits of storage*), and 0 bit per XOR gates

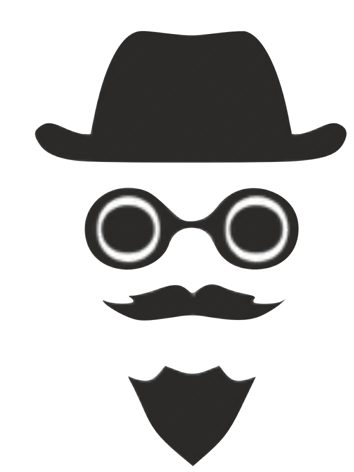
This work: can get *1 bit* per AND gate in total (amortized) and 0 per XOR gates, at a cost of 8x more storage and 4x more computation

best candidates for concrete efficiency so far?

- There is some cool paradigm shift going on there!

M_0'

$$u = x + r$$



M_1'

What to Concretely Take out of that?

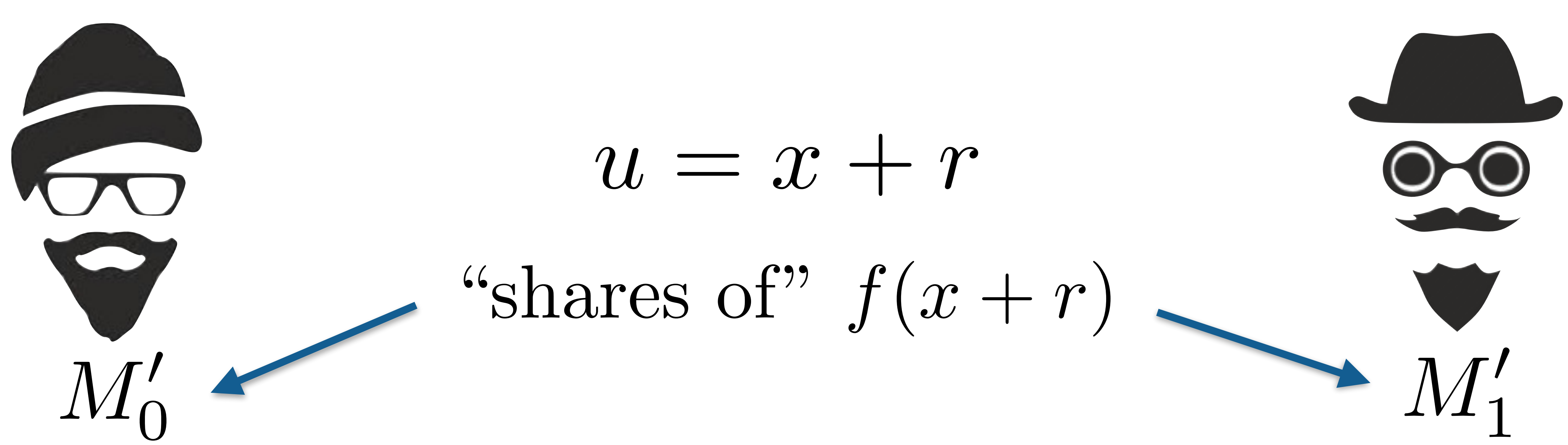
- MPC from truth-table correlations gives great concrete numbers

TinyTable: only 2 bits per AND gate (and 4 bits of storage*), and 0 bit per XOR gates

This work: can get *1 bit* per AND gate in total (amortized) and 0 per XOR gates, at a cost of 8x more storage and 4x more computation

best candidates for concrete efficiency so far?

- There is some cool paradigm shift going on there!



What to Concretely Take out of that?

- MPC from truth-table correlations gives great concrete numbers

TinyTable: only 2 bits per AND gate (and 4 bits of storage*), and 0 bit per XOR gates

This work: can get *1 bit* per AND gate in total (amortized) and 0 per XOR gates, at a cost of 8x more storage and 4x more computation

best candidates for concrete efficiency so far?

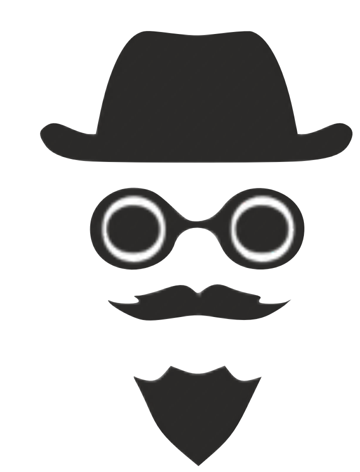
- There is some cool paradigm shift going on there!

public input

$$u = x + r$$

M'_0

"shares of" $f(x+r)$



M'_1

secret (shared) function

Open Questions

Open Questions

- Where is the real barrier?

Open Questions

- Where is the real barrier?
- Can we get sublinear communication *and* linear computation?

Open Questions

- Where is the real barrier?
- Can we get sublinear communication *and* linear computation?
- Can we extend the result to all circuits?

Thanks for your attention

Questions?