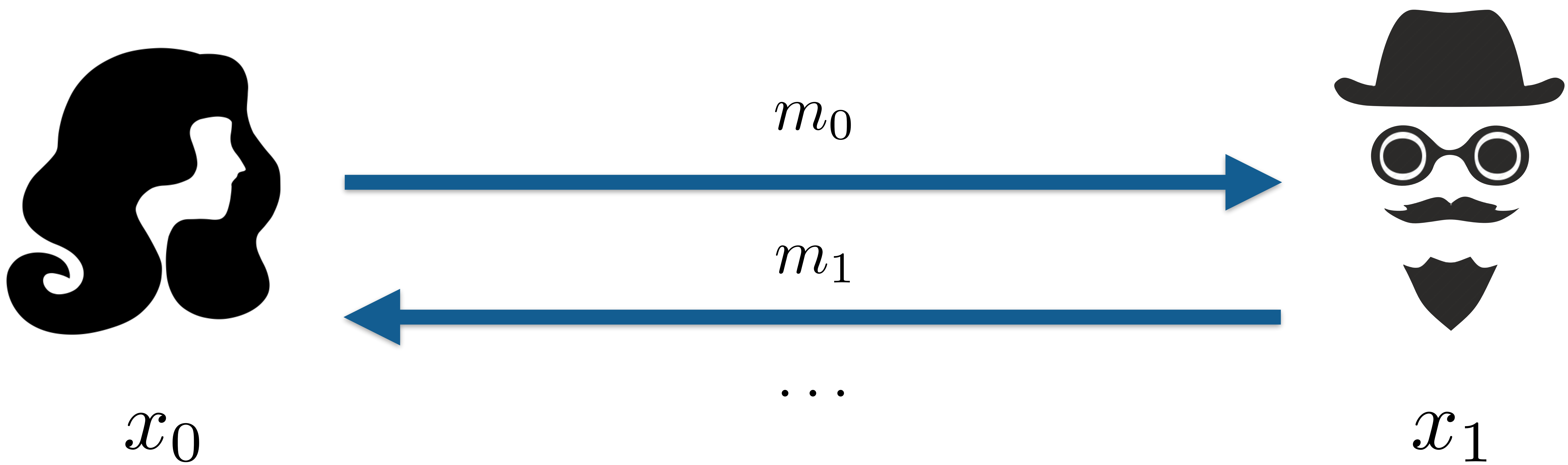
# A Note on the Communication Complexity of Multiparty Computation in the Correlated Randomness Model

*Geoffroy Couteau*

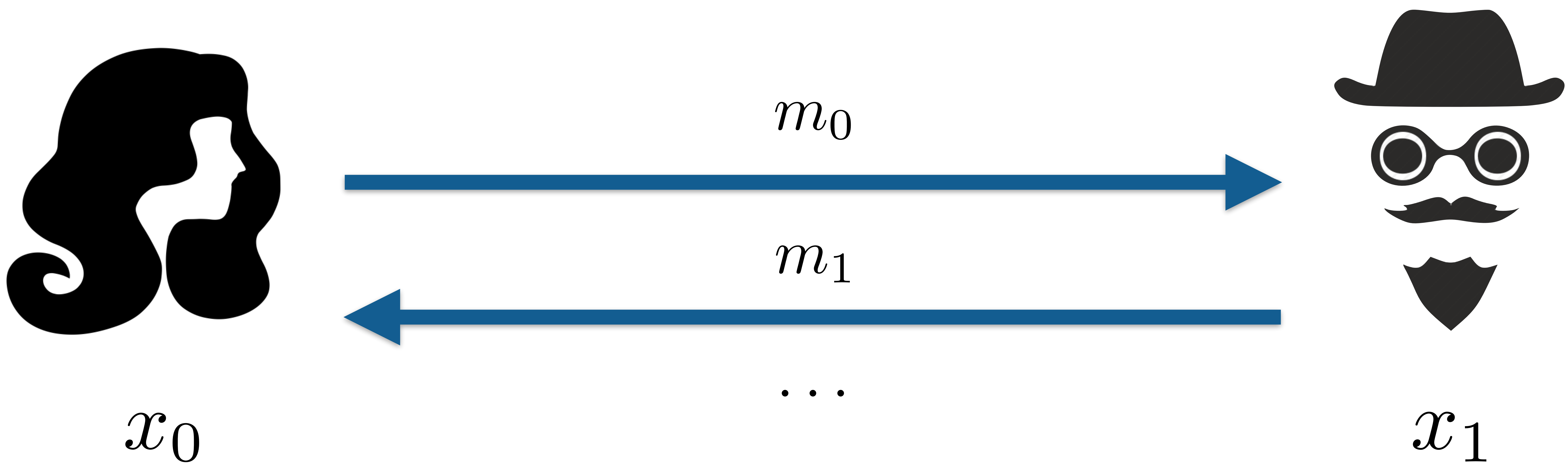
Karlsruher Institut für Technologie

# The Quest for MPC with Low Communication




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

# The Quest for MPC with Low Communication

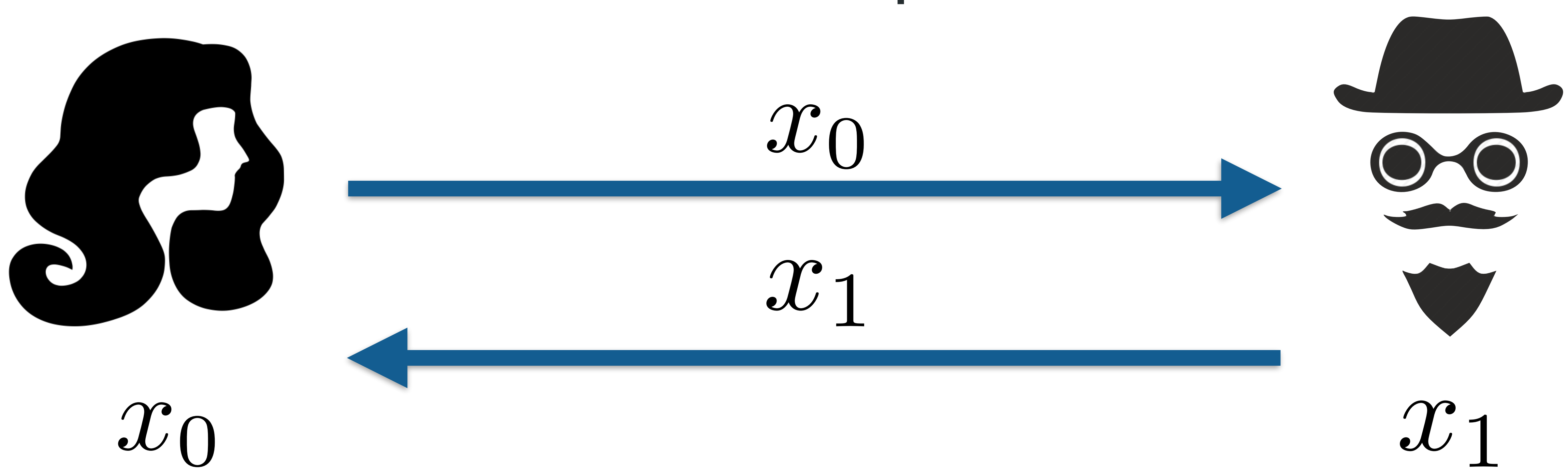



$$y = f(x_0, x_1)$$

- Correctness: the parties learn the correct output
- Privacy: the parties learn nothing more than the output

# The Quest for MPC with Low Communication

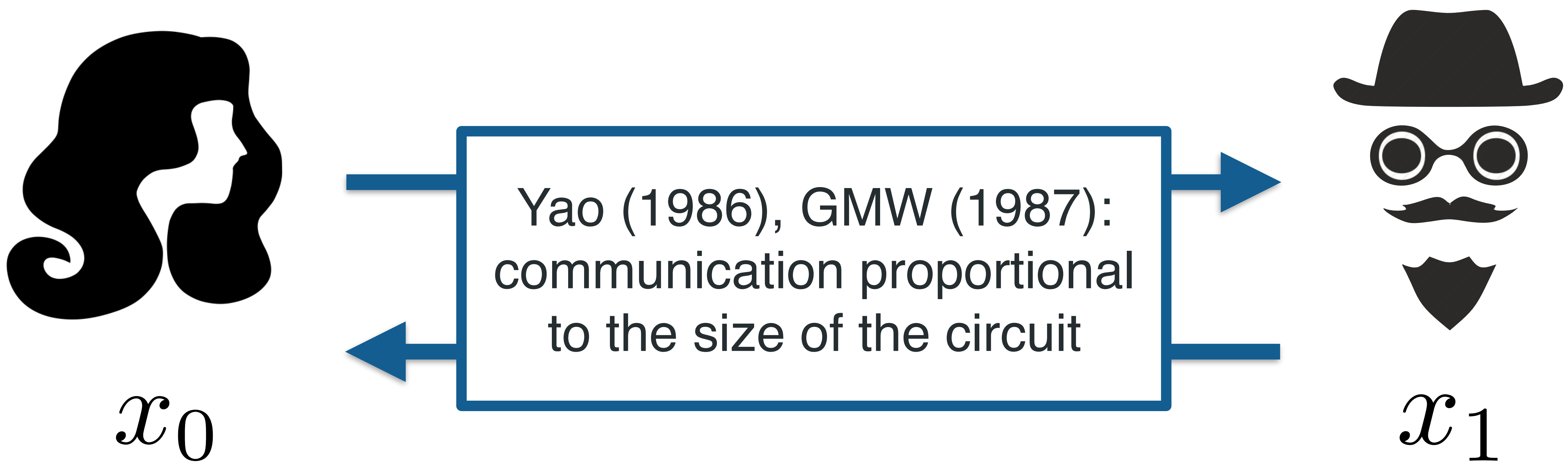
Insecure computation:




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

# The Quest for MPC with Low Communication

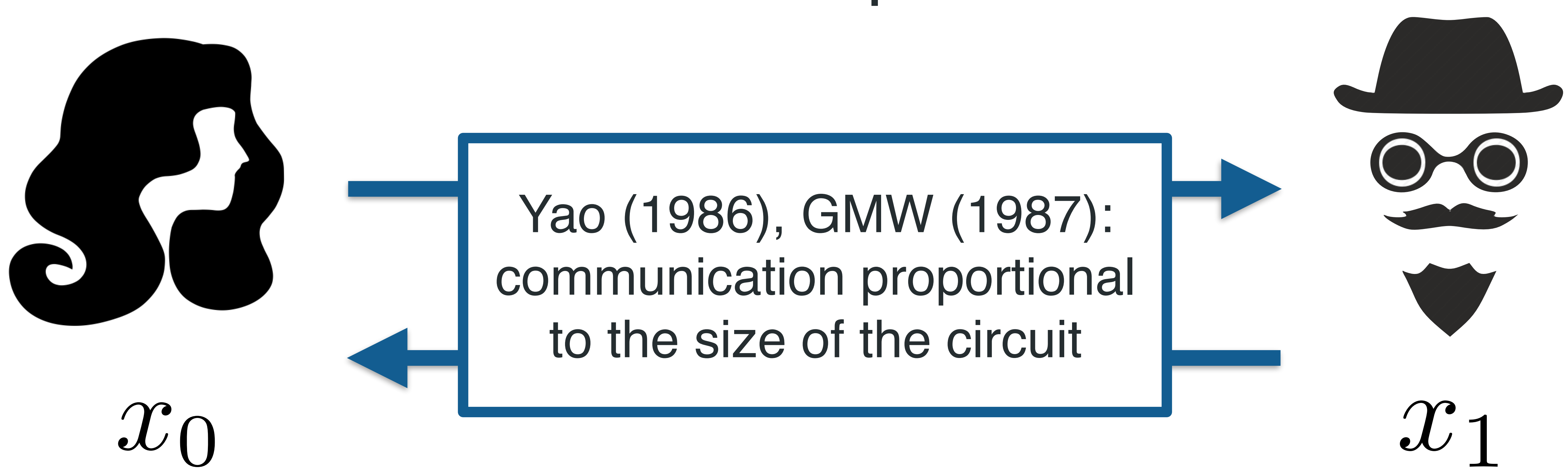
Secure computation:




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

# The Quest for MPC with Low Communication

Secure computation:

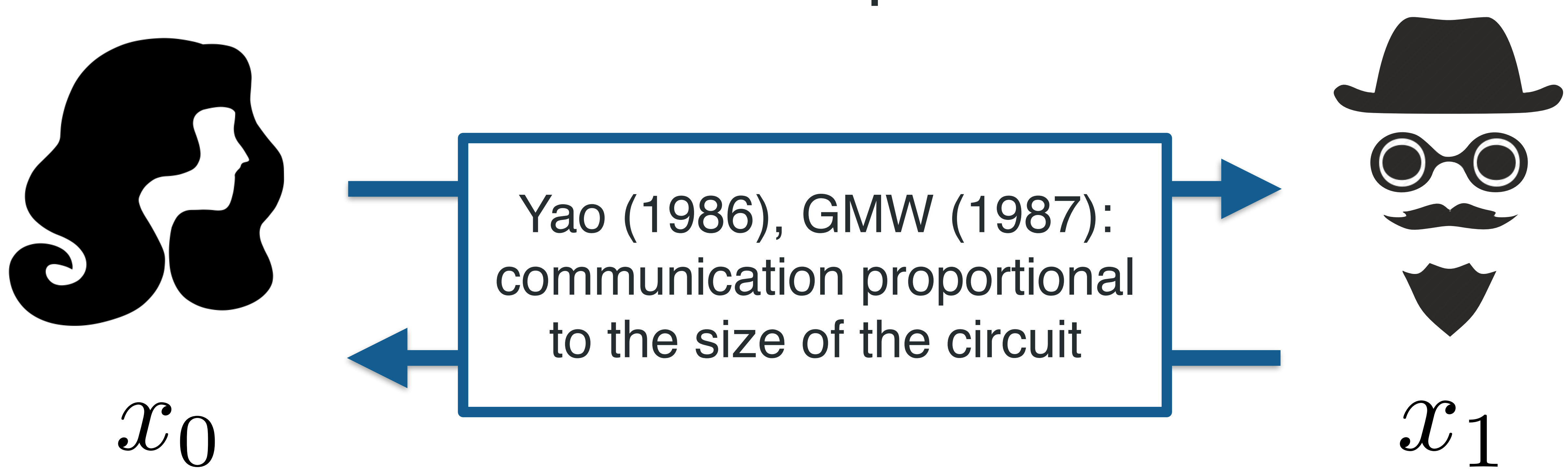



$$y = f(x_0, x_1)$$

Does secure computation inherently require so much communication?

# The Quest for MPC with Low Communication

Secure computation:



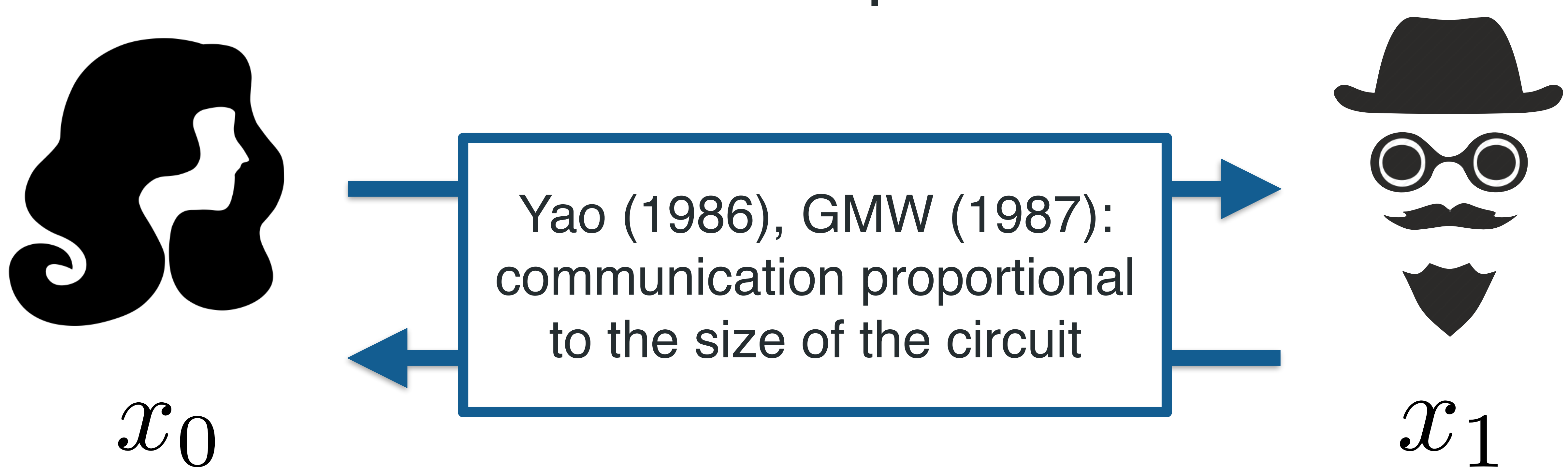

$$y = f(x_0, x_1)$$

**Does secure computation inherently require so much communication?**

Gentry (2009): MPC with optimal communication from (variants of) LWE

# The Quest for MPC with Low Communication

Secure computation:




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

**Does secure computation inherently require so much communication?**

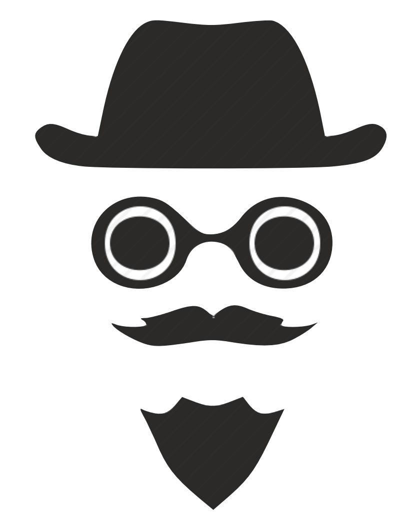
Gentry (2009): MPC with optimal communication from (variants of) LWE

This work: revisiting this question for MPC with correlated randomness

# MPC with Correlated Randomness

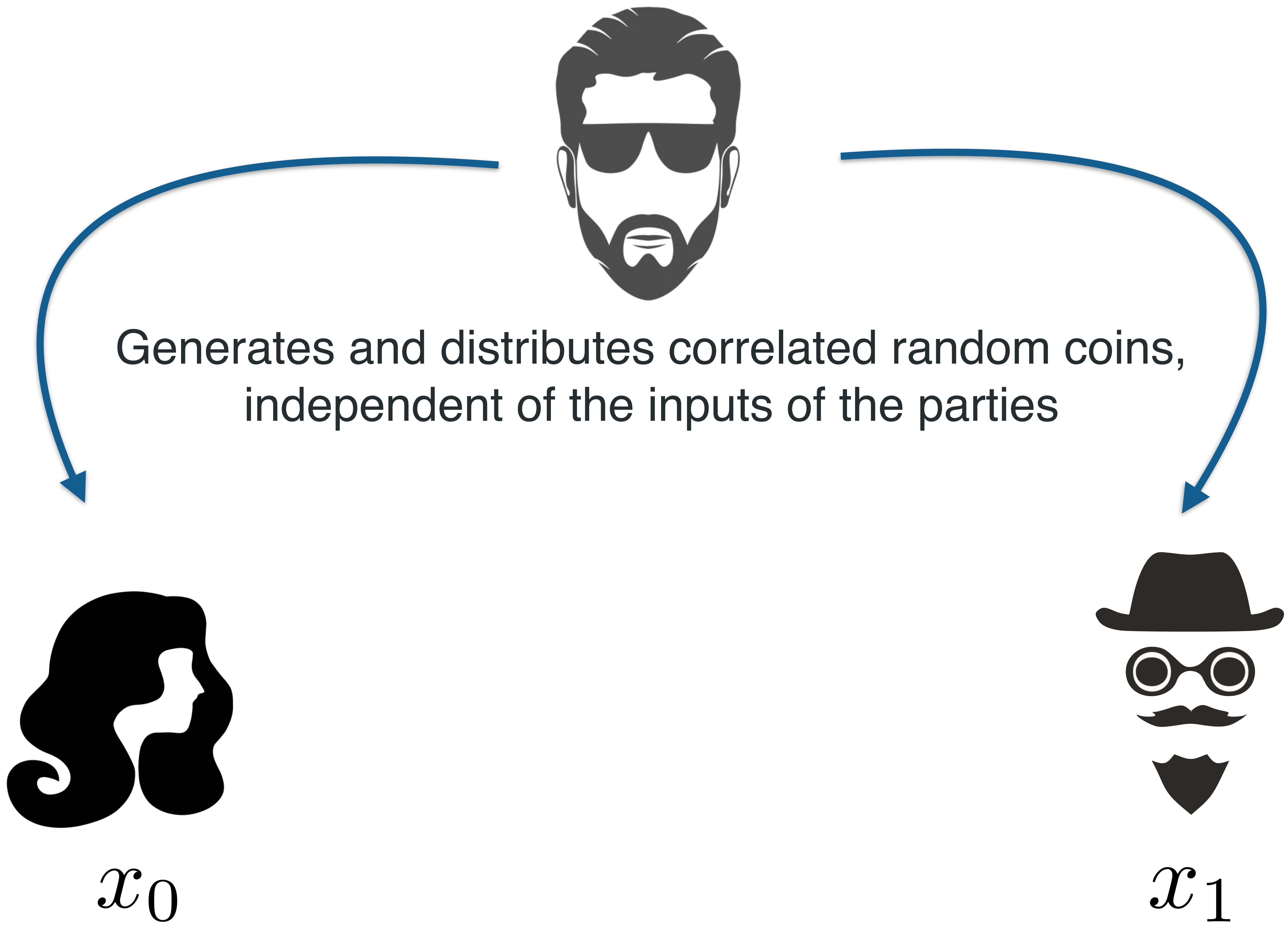
Generates and distributes correlated random coins,
independent of the inputs of the parties

$x_0$

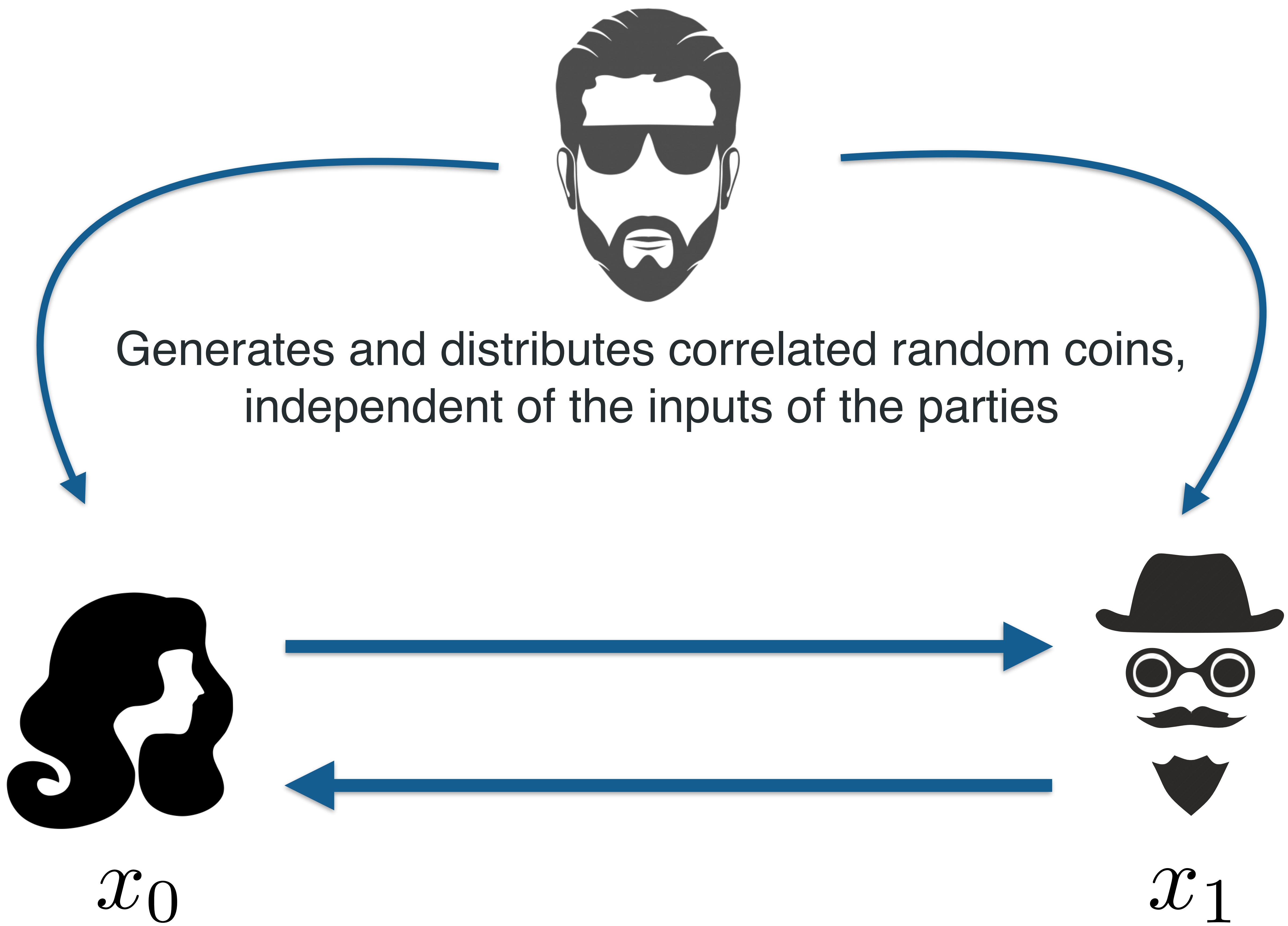


$x_1$

# MPC with Correlated Randomness



# MPC with Correlated Randomness

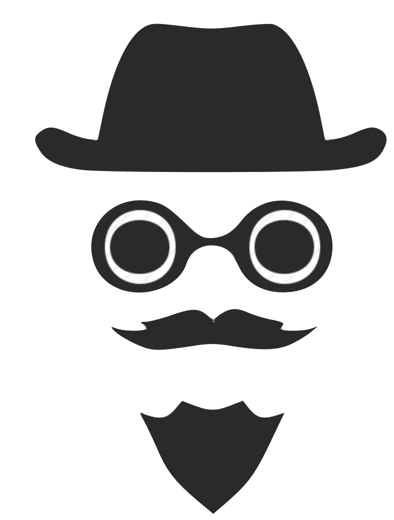


# MPC with Correlated Randomness

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



$x_1$

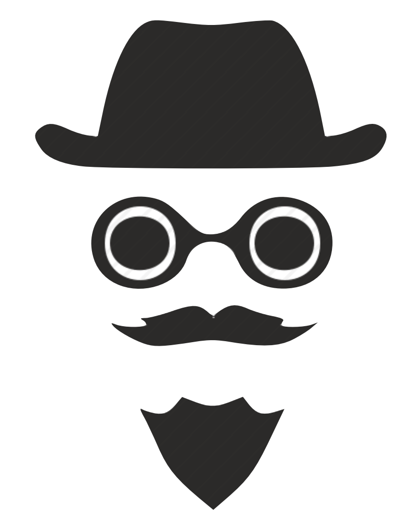
# MPC with Correlated Randomness

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



$x_1$

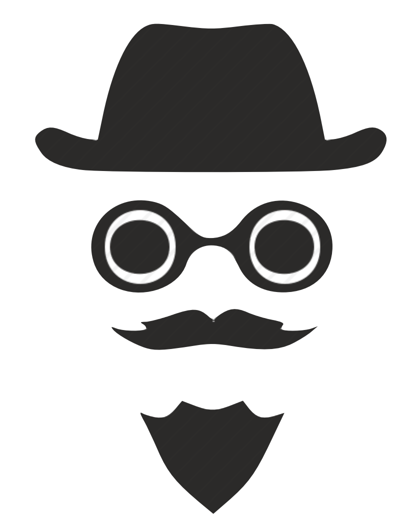
# MPC with Correlated Randomness

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



$x_1$

All known protocols in the correlated randomness model have communication proportional to the circuit size

# 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

$x_1$

All known protocols in the correlated randomness model have communication proportional to the circuit size

DNPR16: this is inherent for gate-by-gate protocols

# 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 and function-independent preprocessing

+ Concrete efficiency improvements for TinyTable

# 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 and function-independent preprocessing

+ Concrete efficiency improvements for TinyTable

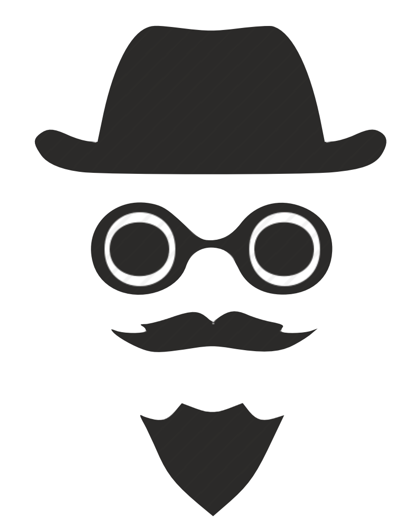
We'll focus on 2 parties & semi-honest security here

# 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

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

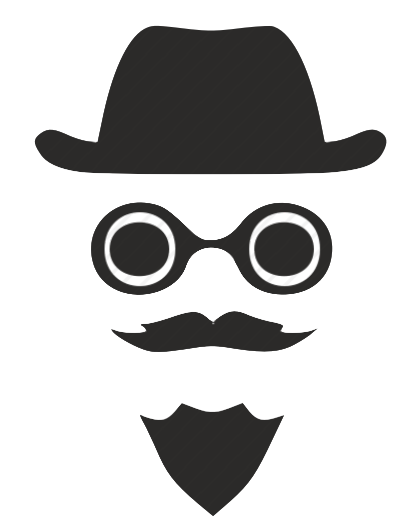
$M =$

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 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

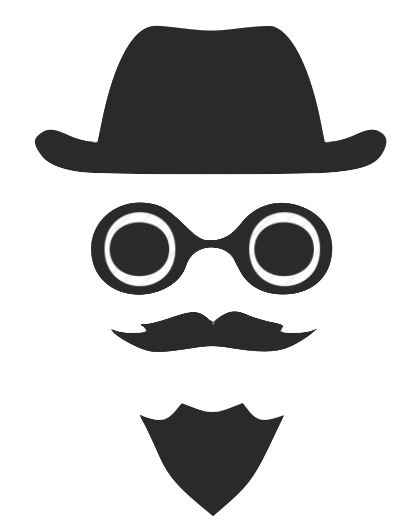
$$f(x + r) = f((x_0 + r_0) + (x_1 + r_1))$$

$M' =$

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ... | 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

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

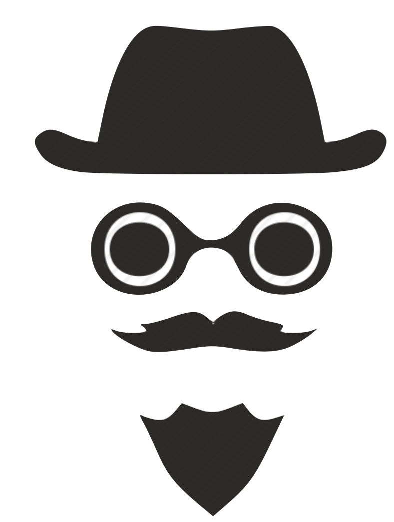
$M' =$

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ... | 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
$M' = M'_0 + M'_1$

$x_0$

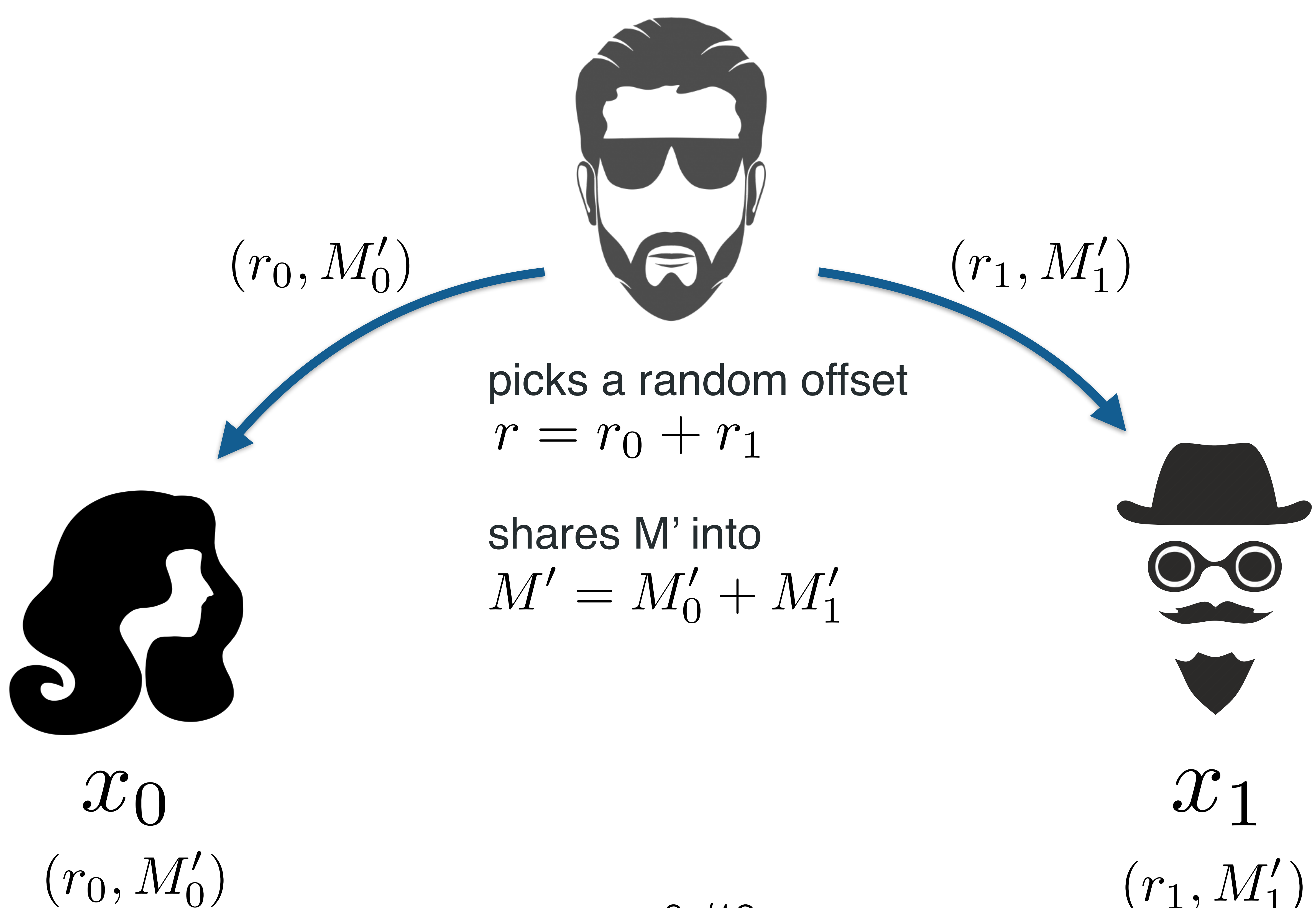


$x_1$

# Sharing Truth-Table Correlations

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

$$M' = \begin{array}{|c|c|c|c|c|c|c|c|c|c|c|c|c|c|c|c|}\hline \ldots & f(N\text{-}5) & f(N\text{-}4) & f(N\text{-}3) & f(N\text{-}2) & f(N\text{-}1) & f(N) & f(0) & f(1) & f(2) & f(3) & f(4) & f(5) & \ldots & \ldots & \ldots \\ \hline \end{array}$$



# Sharing Truth-Table Correlations

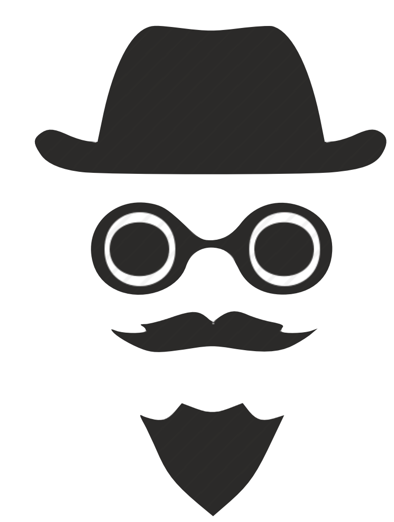
$$f(x + r) = f((x_0 + r_0) + (x_1 + r_1))$$

$M' =$

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| … | 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

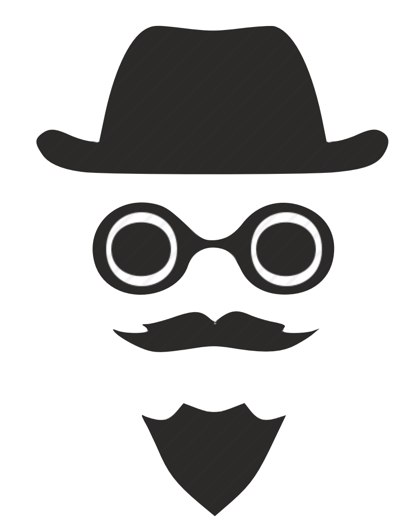
$$f(x+r) = f((x_0+r_0)+(x_1+r_1))$$

$M' =$

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ... | 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) | ... | ... | ... |

$u_0 = x_0 + r_0$

$u_1 = x_1 + r_1$



$x_0$

$(r_0, M'_0)$

$x_1$

$(r_1, M'_1)$

# Sharing Truth-Table Correlations

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

$M' =$

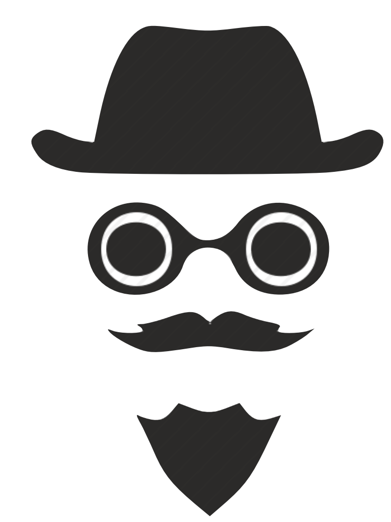
| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ... | f(N-5) | f(N-4) | f(N-3) | f(N-2) | f(N-1) | f(N) | f(0) | f(1) | f(2) | f(3) | f(4) | f(5) | ... | ... | ... |

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

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

$u_0 = x_0 + r_0$

$u_1 = x_1 + r_1$



$x_0$

$(r_0, M'_0)$

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

$x_1$

$(r_1, M'_1)$

# 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: $2n$

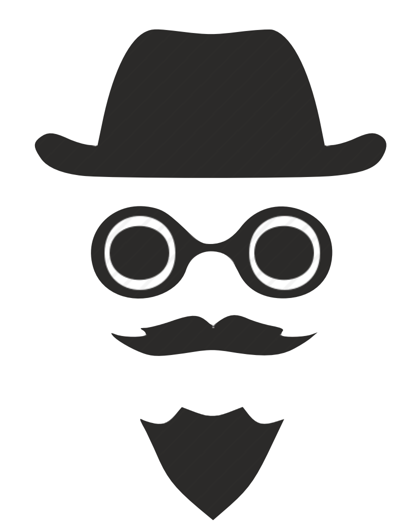
storage: $m \cdot 2^n + n$

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

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

$u_0 = x_0 + r_0$

$u_1 = x_1 + r_1$



$x_0$

$(r_0, M'_0)$

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

$x_1$

$(r_1, M'_1)$

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

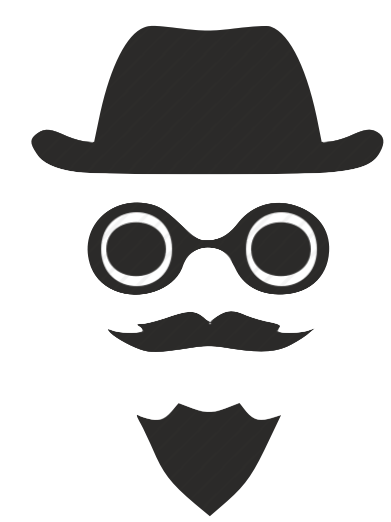
that's bad

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

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

$u_0 = x_0 + r_0$

$u_1 = x_1 + r_1$



$x_0$

$(r_0, M'_0)$

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

$x_1$

$(r_1, M'_1)$

# 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

communication: $2n$

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

that's bad

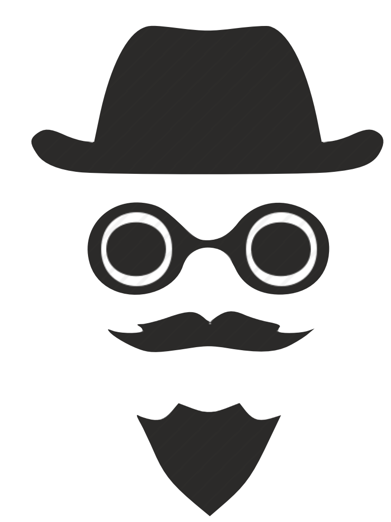
IKMOP (2013): a polynomial storage for all functions would imply a breakthrough in information-theoretic PIR

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

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

$u_0 = x_0 + r_0$

$u_1 = x_1 + r_1$



$x_0$

$(r_0, M'_0)$

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

$x_1$

$(r_1, M'_1)$

# The Core Lemma

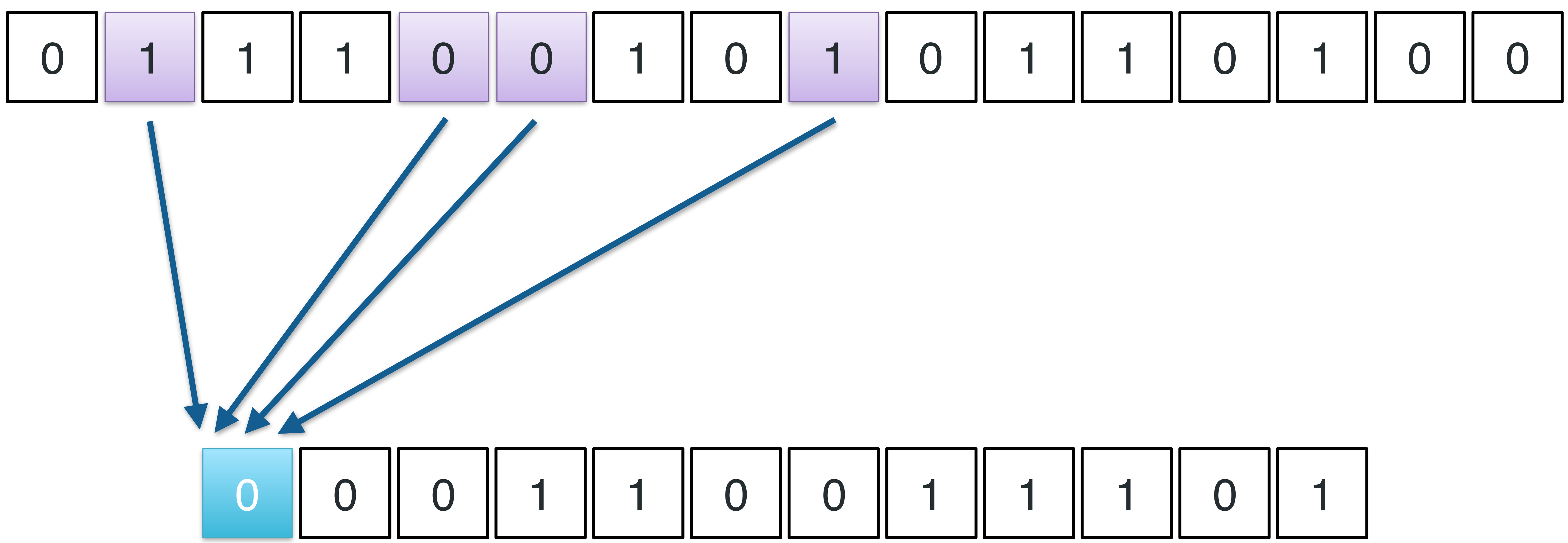
Let $f$ be a $c$-local function, with input of size $n$ and output of size $m$. Then there exists a protocol $\Pi$ which securely computes shares of $f$ in the correlated randomness model, with optimal communication $O(n)$ and storage $m \cdot 2^c + n$.

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 0 |

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 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 $\Pi$ which securely computes shares of $f$ in the correlated randomness model, with optimal communication $O(n)$ and storage $m \cdot 2^c + n$.

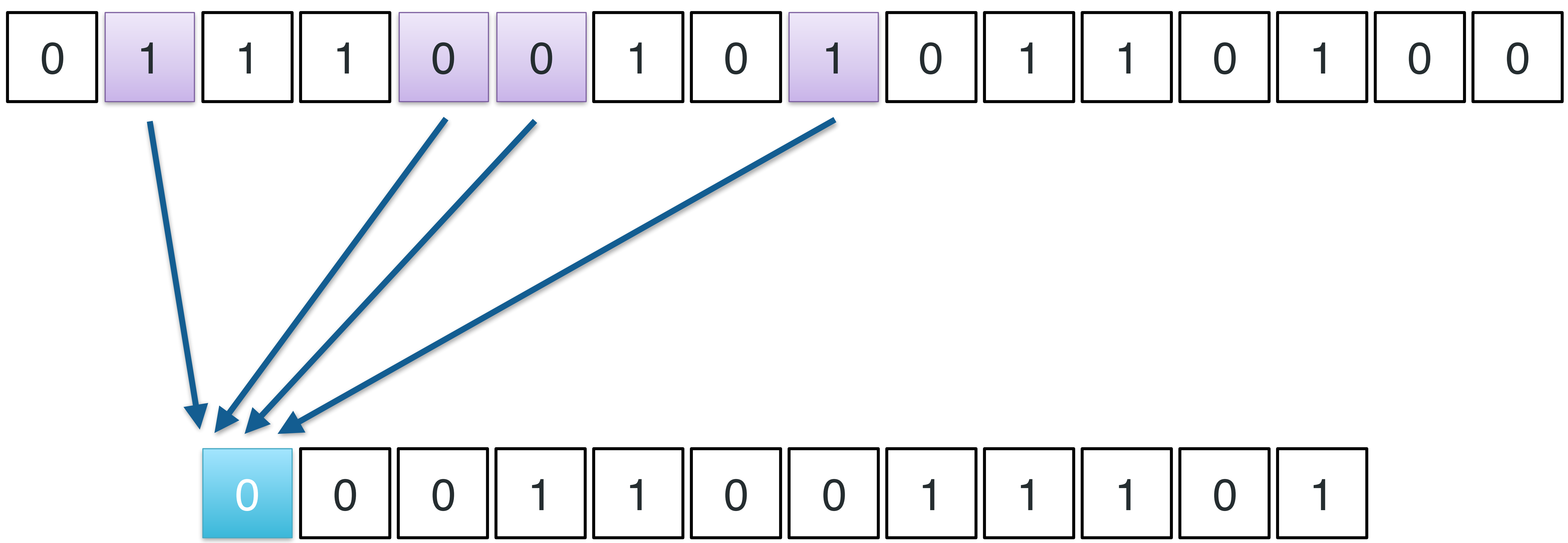



$$f(x) = (f_1(x[S_1]), f_2(x[S_2]), \cdots, f_m(x[S_m]))$$
$$\forall i, \ |S_i| = c$$

# 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 $\Pi$ which securely computes shares of $f$ in the correlated randomness model, with optimal communication $O(n)$ and storage $m \cdot 2^c + n$.

instead of $m \cdot 2^n + n$



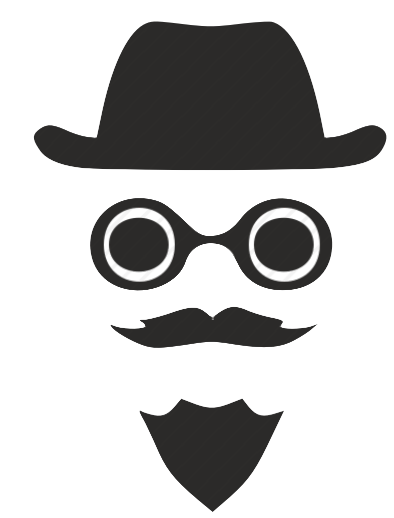

$$f(x) = (f_1(x[S_1]), f_2(x[S_2]), \cdots, f_m(x[S_m]))$$

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

# 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 $\Pi$ which securely computes shares of $f$ in the correlated randomness model, with optimal communication $O(n)$ and storage $m \cdot 2^c + n$.

$x_0$



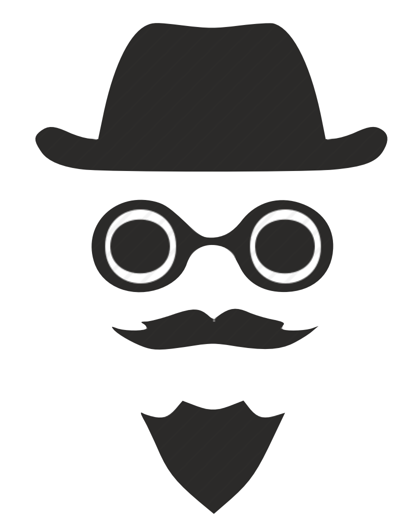
$x_1$

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

# 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 $\Pi$ which securely computes shares of $f$ in the correlated randomness model, with optimal communication $O(n)$ and storage $m \cdot 2^c + n$.

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

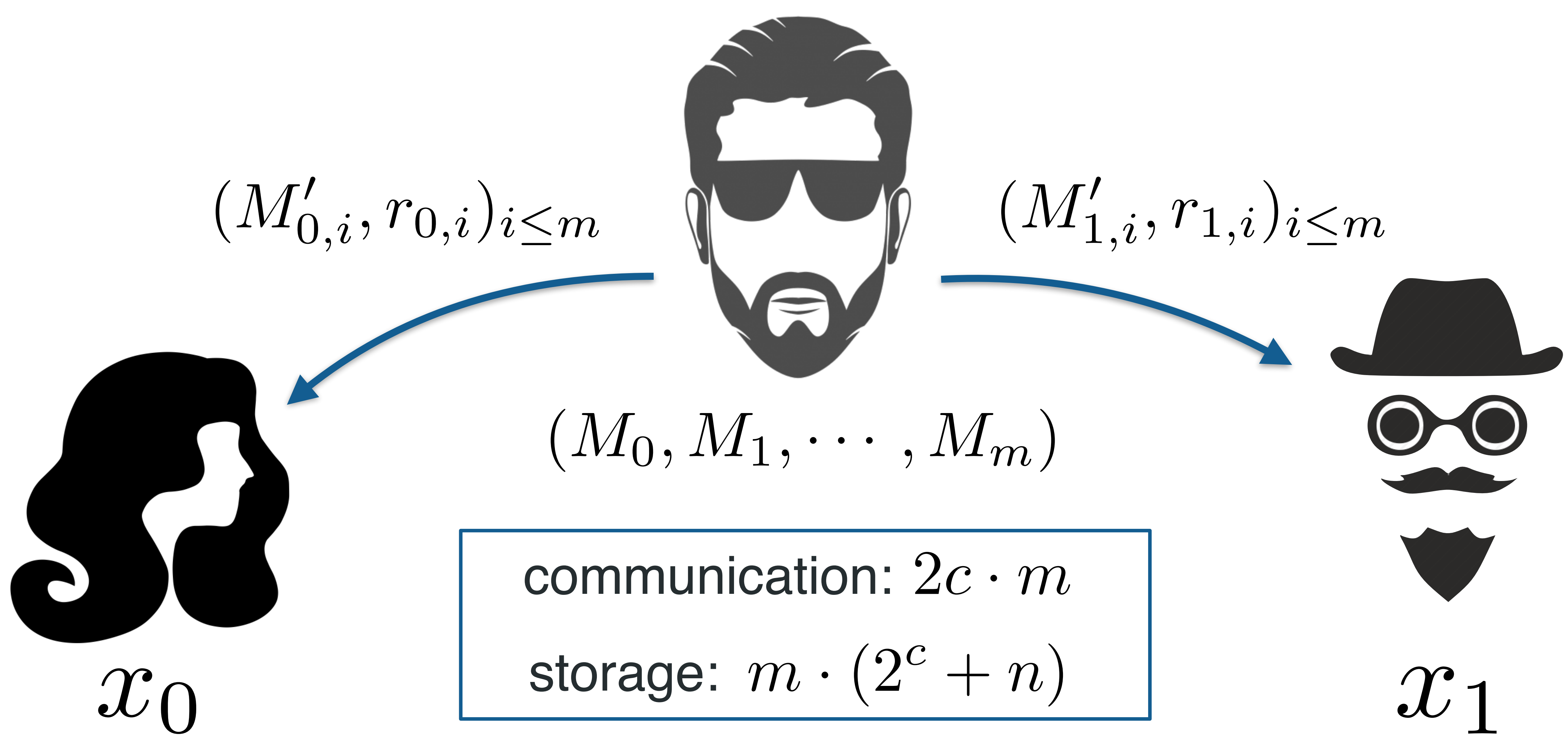


$x_0$ $x_1$

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

# 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 $\Pi$ 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 $\Pi$ which securely computes shares of $f$ in the correlated randomness model, with optimal communication $O(n)$ and storage $m \cdot 2^c + n$.

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

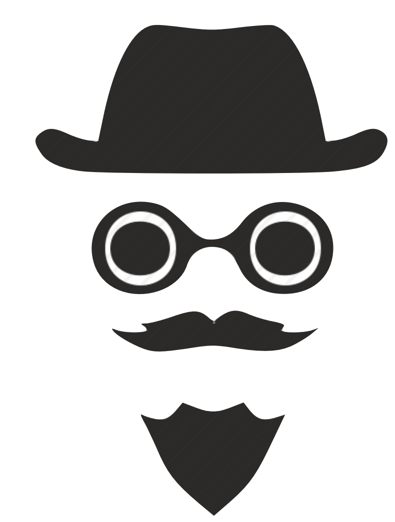
$M_1$ , $M_2$ ... $M_m$

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

$\xrightarrow{\quad r_1 \quad}$ $\xrightarrow{\ r_2\ }$ $\xrightarrow{\quad r_m \quad}$

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

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



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

# 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 $\Pi$ which securely computes shares of $f$ in the correlated randomness model, with optimal communication $O(n)$ and storage $m \cdot 2^c + n$.

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

$M_1$ , $M_2$ ... $M_m$

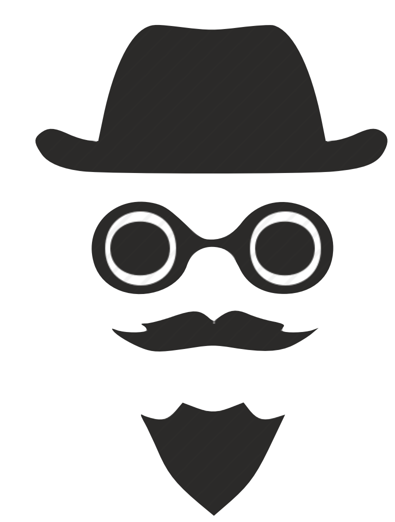
| $f_1(1)$ | $f_1(2)$ | ... | $f_1(2^c)$ | , | $f_2(1)$ | $f_2(2)$ | ... | $f_2(2^c)$ | ... | $f_m(1)$ | $f_m(2)$ | ... | $f_m(2^c)$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|

$\xrightarrow{\hspace{4em}}$ $r_1$ $\quad$ $\xrightarrow{\hspace{1em}}$ $r_2$ $\quad$ $\xrightarrow{\hspace{3em}}$ $r_m$

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

$x_0 + r_0$

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 $\Pi$ which securely computes shares of $f$ in the correlated randomness model, with optimal communication $O(n)$ and storage $m \cdot 2^c + n$.

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

$M_1$ , $M_2$ ... $M_m$

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

$r_1$ $r_2$ $r_m$

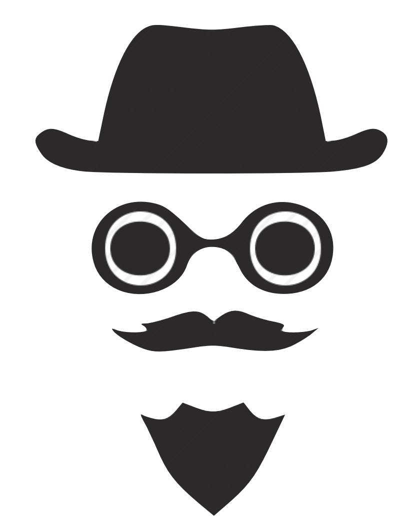
$$\forall i, \ |r_i| = c$$

$x_0 + r_0$

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

communication: $2n$

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

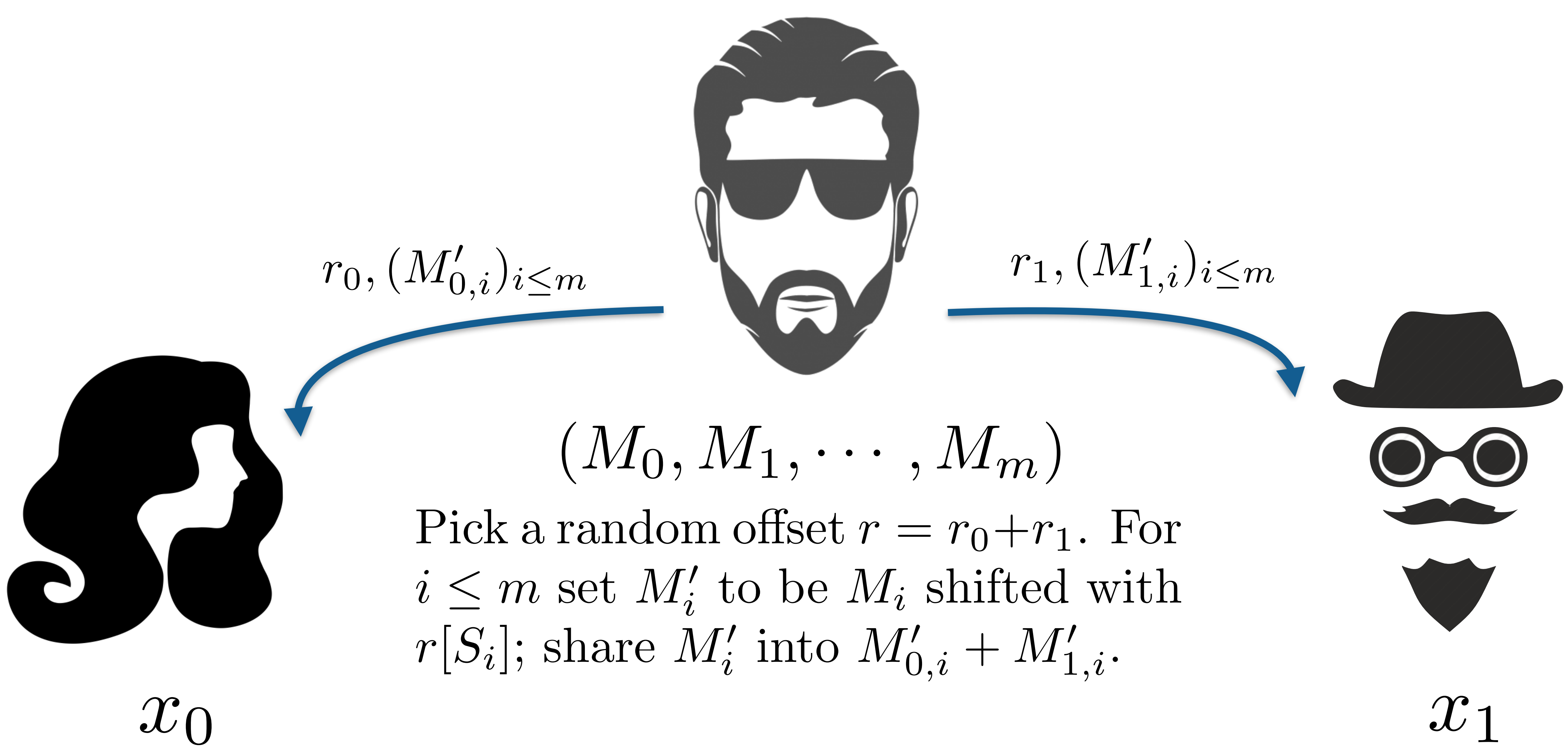


$x_1 + r_1$

# 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 $\Pi$ 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 $\Pi$ which securely computes shares of $f$ in the correlated randomness model, with optimal communication $O(n)$ and storage $m \cdot 2^c + n$.

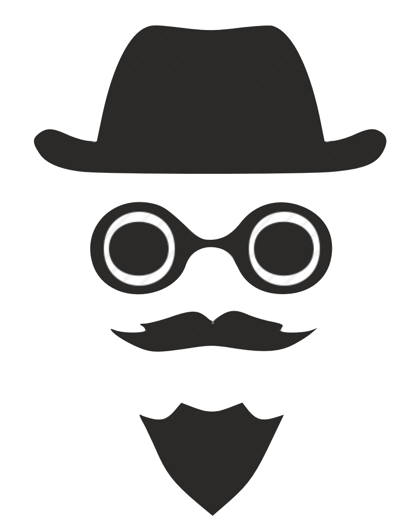
$$f(x) = (f_1(x[S_1]), f_2(x[S_2]), \cdots, f_m(x[S_m]))$$

$y_{0,i} \leftarrow M'_{0,i}|_{u[S_i]}$

$y_{1,i} \leftarrow M'_{1,i}|_{u[S_i]}$

$u_0 = x_0 + r_0$

$u_1 = x_1 + r_1$



$x_0$

$r_0, (M'_{0,i})_{i \leq m}$

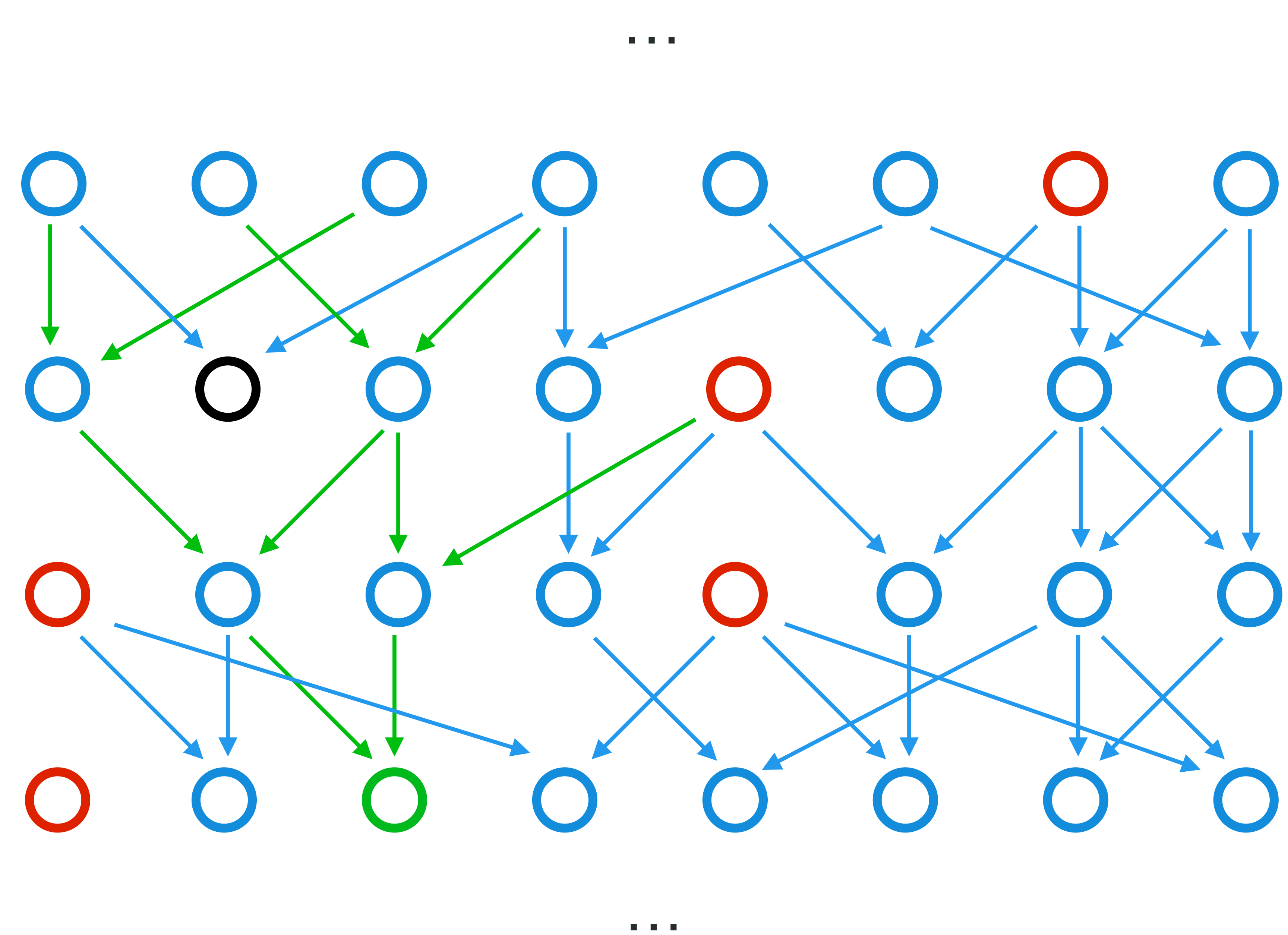
$u \leftarrow u_0 + u_1$

$x_1$

$r_1, (M'_{1,i})_{i \leq m}$

# Construction

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



: node

: input node

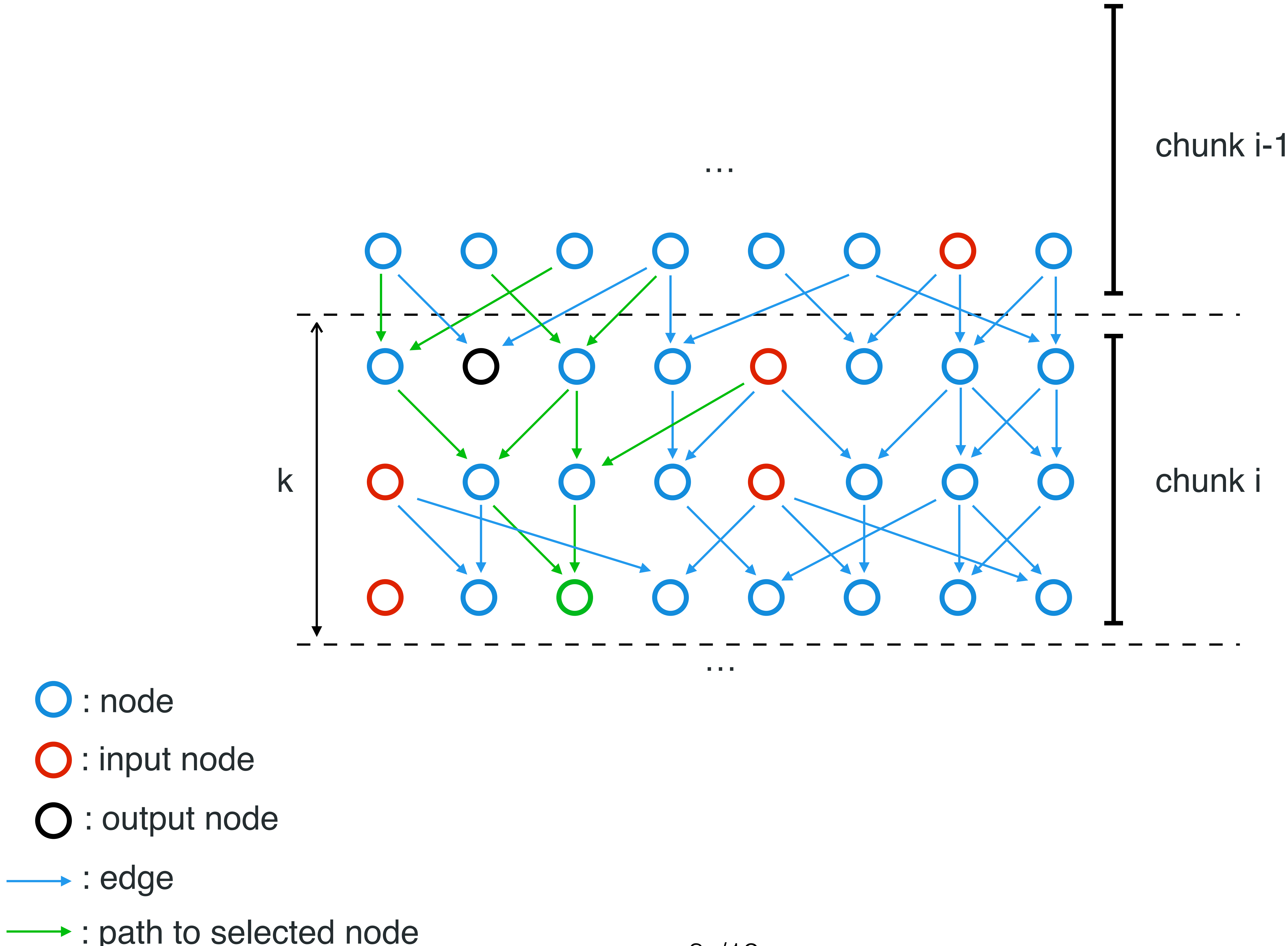
: output node

: edge

: path to selected node

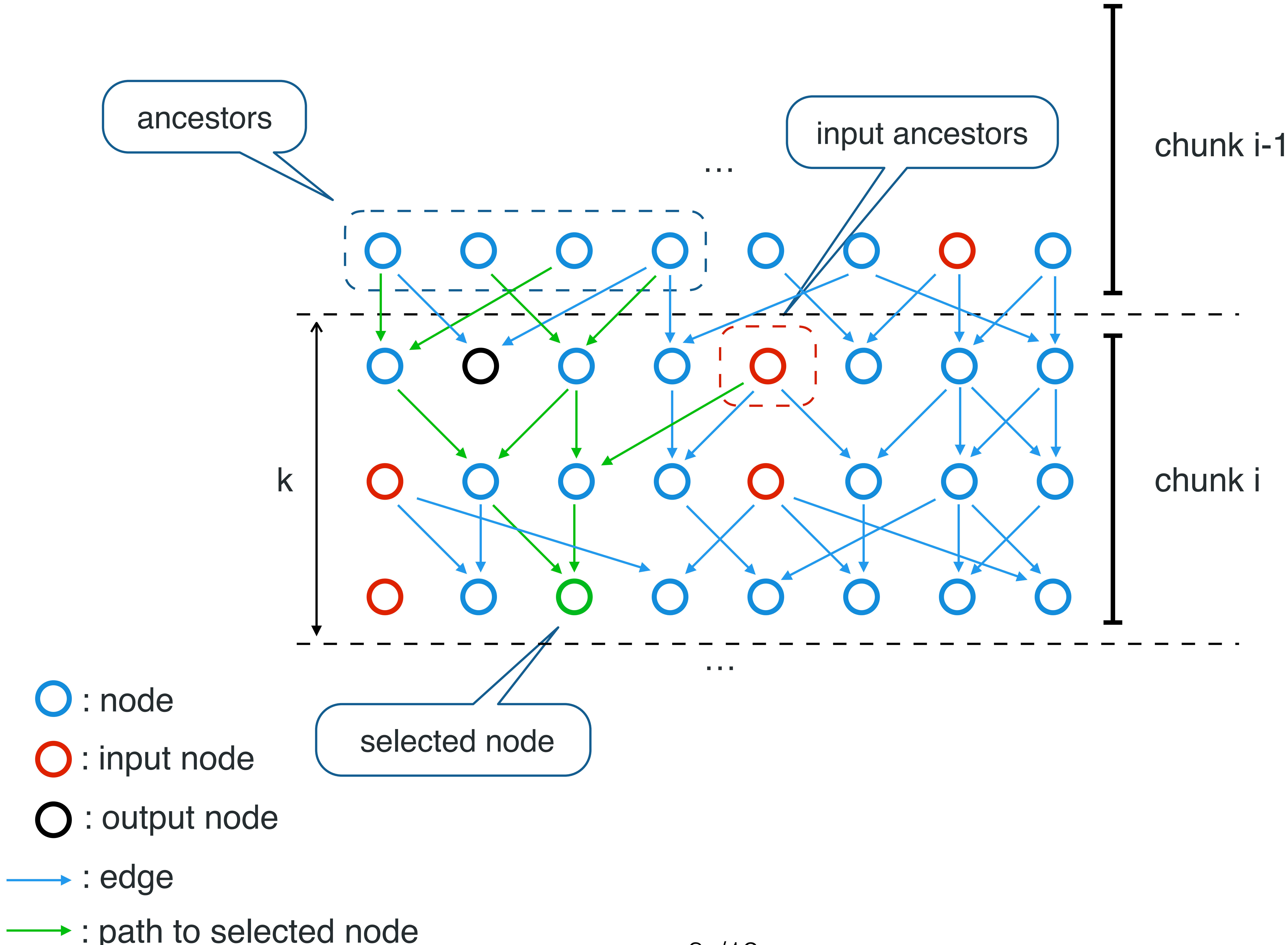
# Construction

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



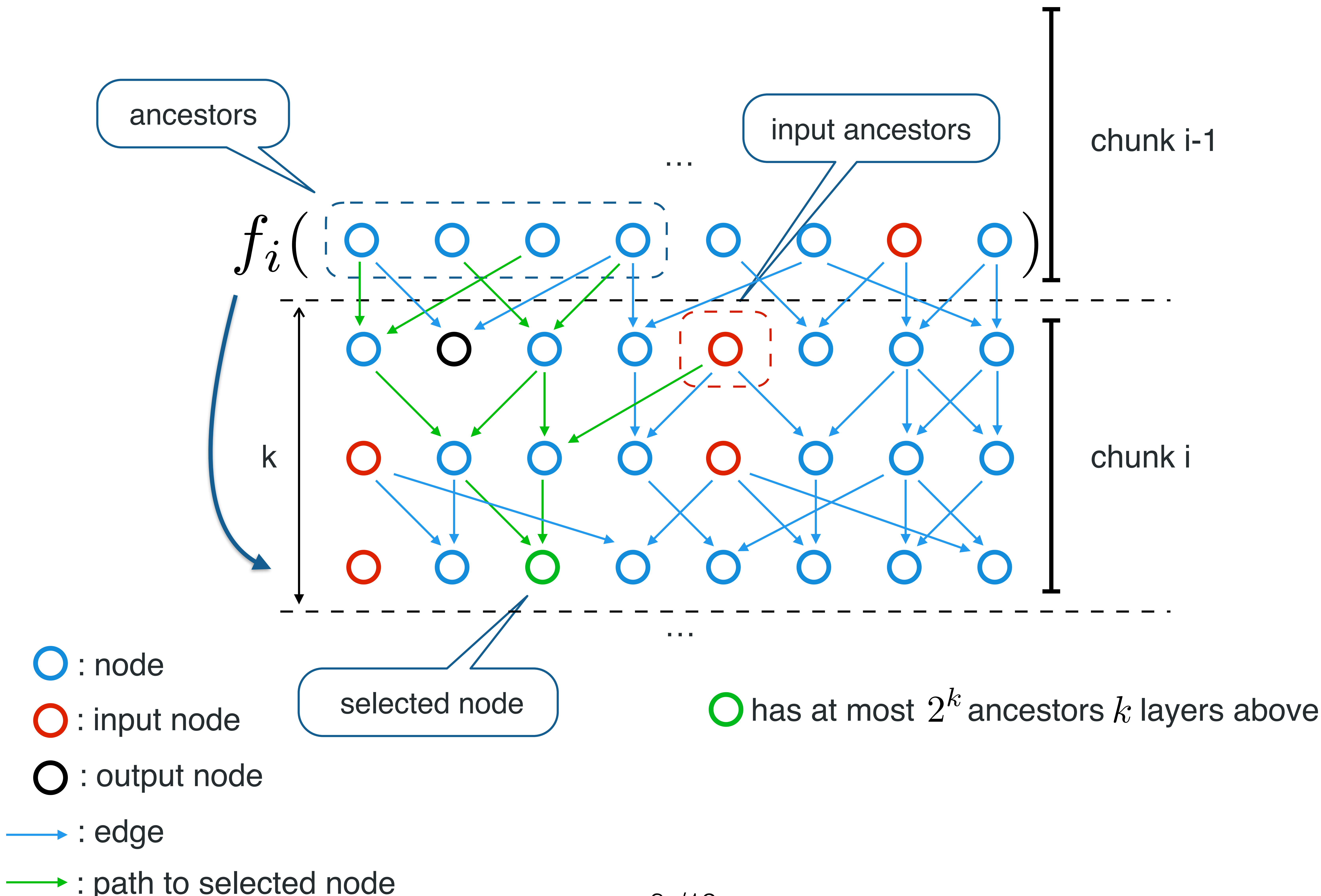
# Construction

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



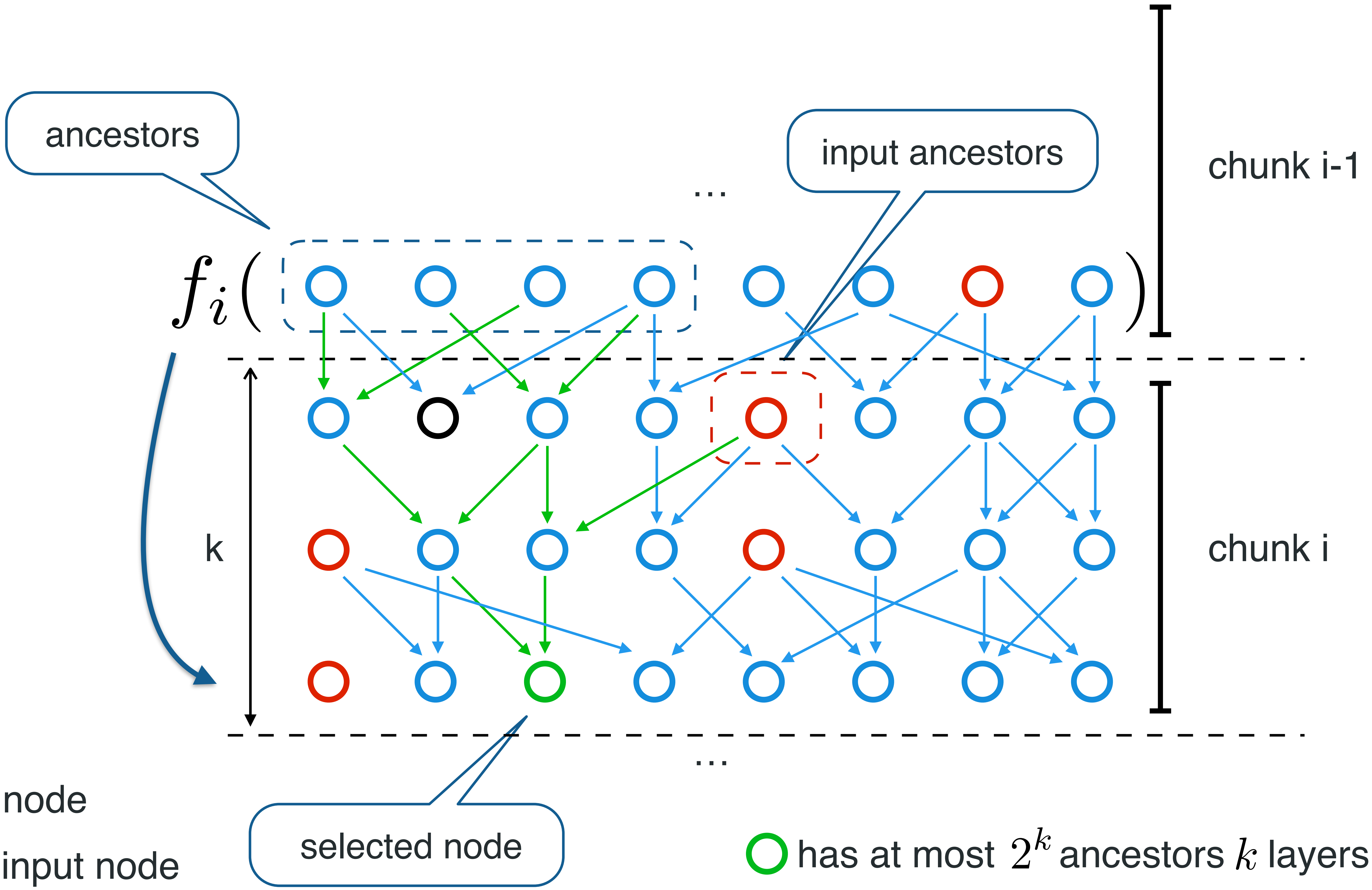
# Construction

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




: node

: input node

: output node

: edge

: path to selected node

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 $\Pi$ 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 $O(w \cdot 2^{2^k})$

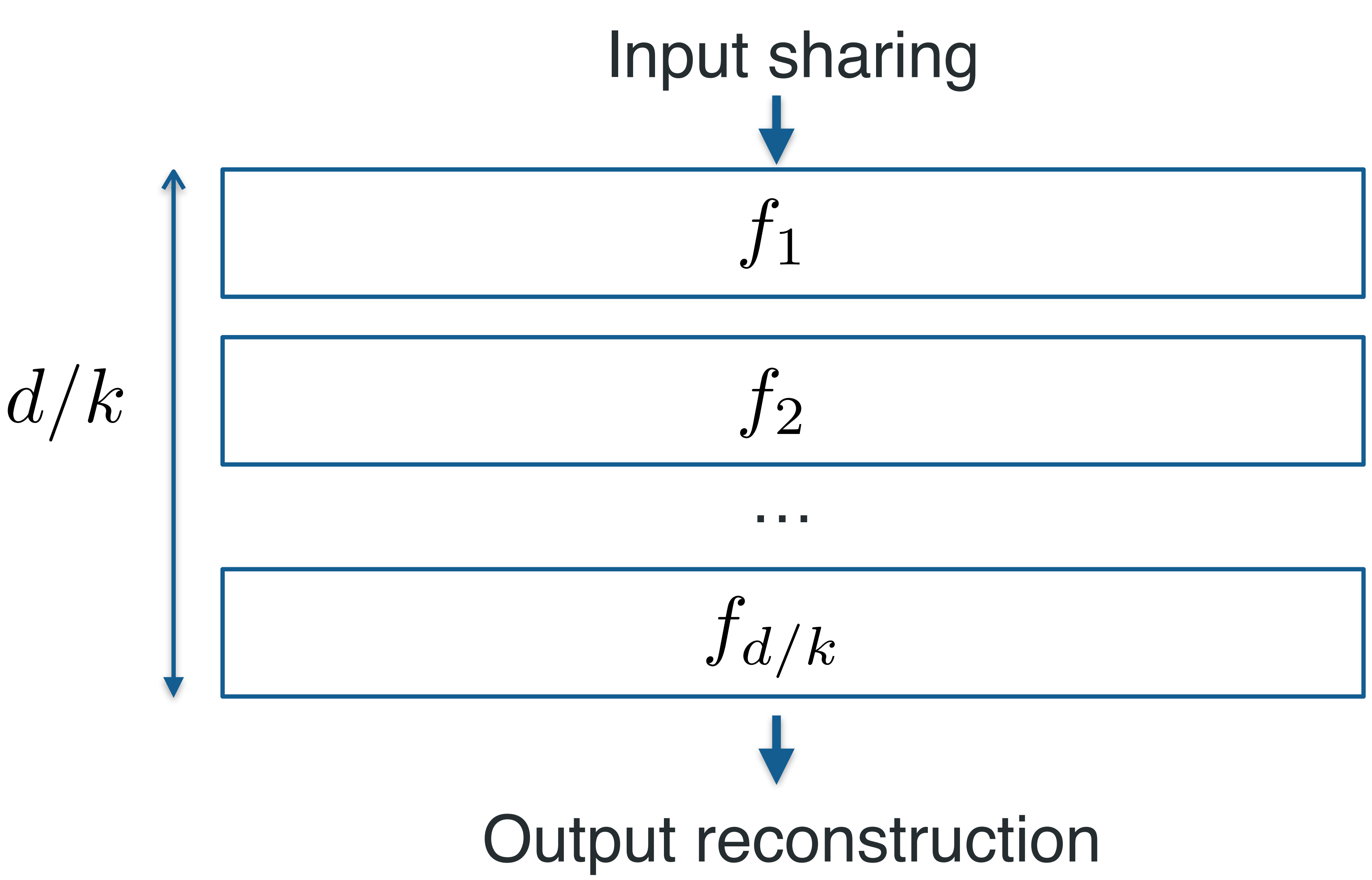
# Construction

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 $\Pi$ 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 $O(w \cdot 2^{2^k})$



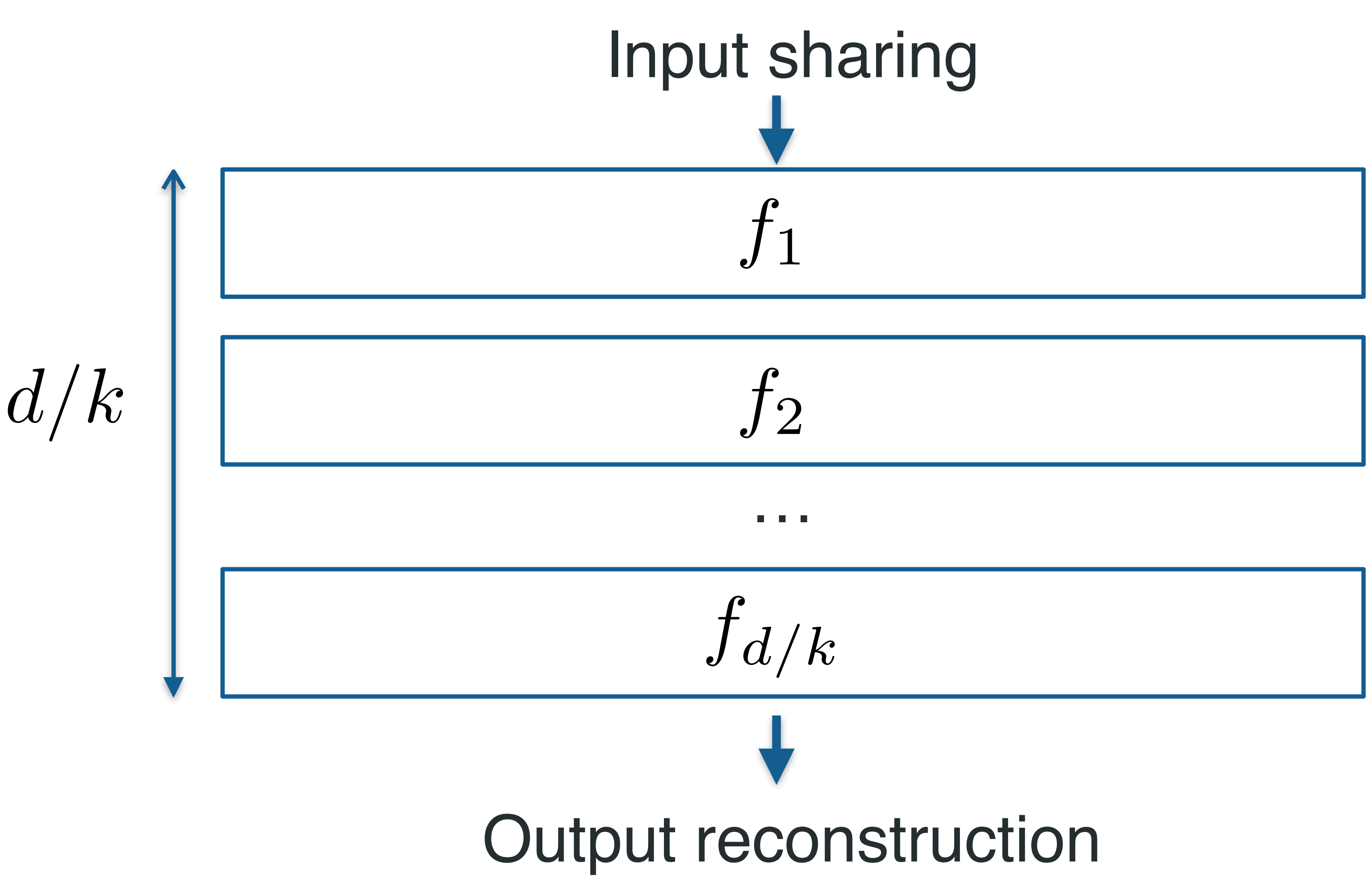
# Construction

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 $\Pi$ 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 $\Pi$ 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 $O(w \cdot 2^{2^k})$

Input sharing

$d/k$

$f_1$

$f_2$

…

$f_{d/k}$

Output reconstruction

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:

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

# 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?

(Paper is online: ia.cr/2018/465)