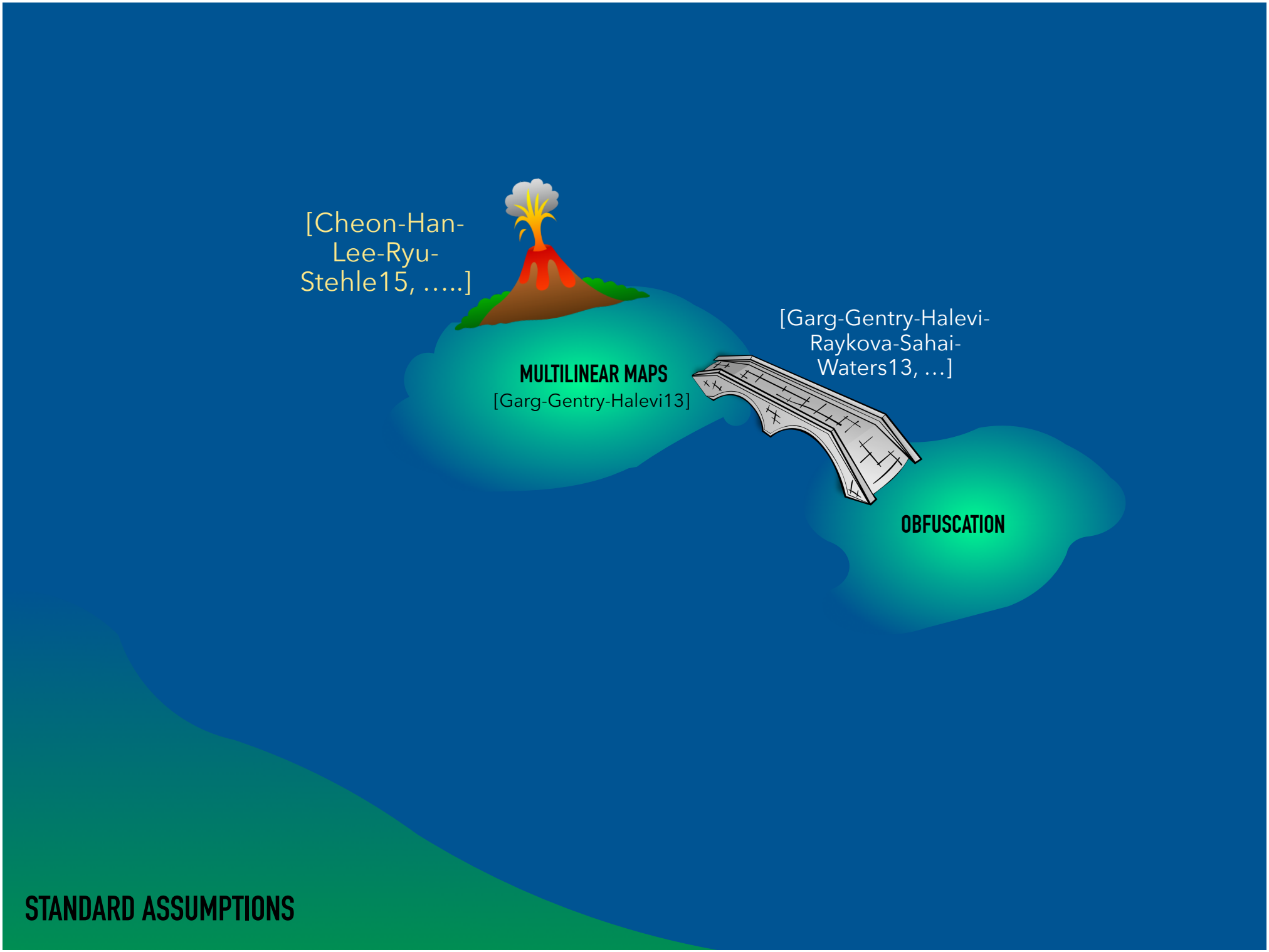
Lockable Obfuscation

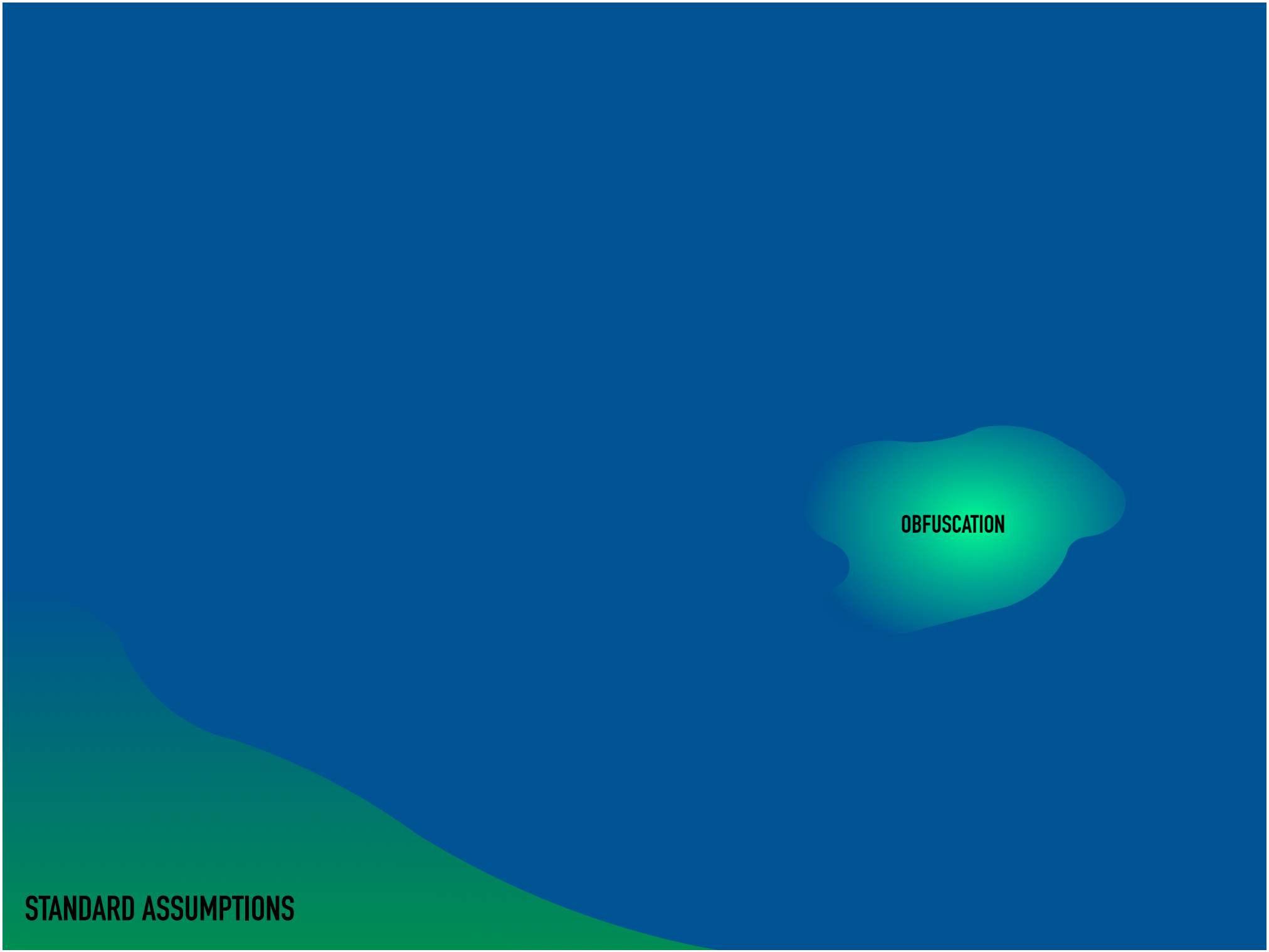
Rishab Goyal Venkata Koppula

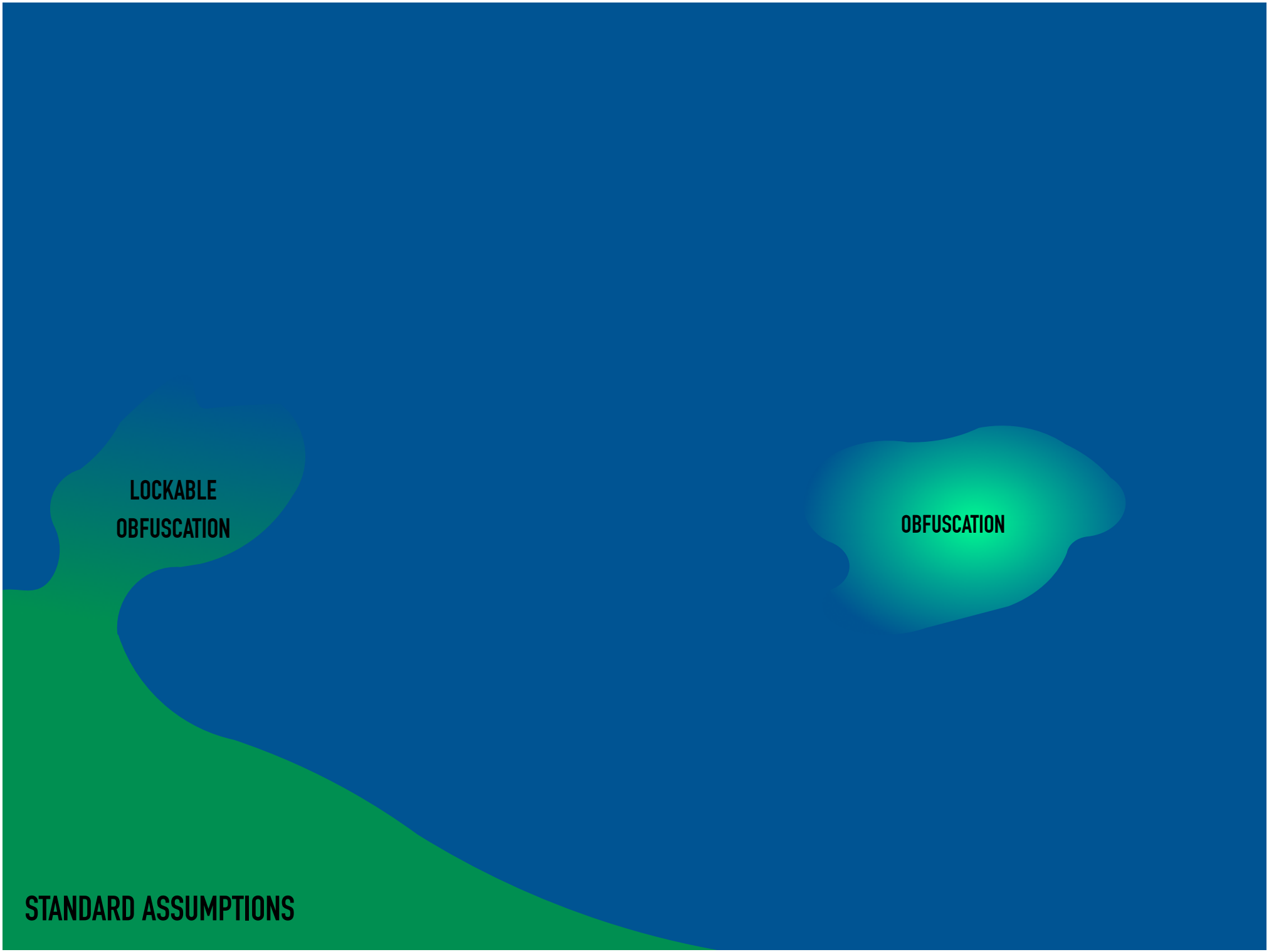
Brent Waters

THE UNIVERSITY OF TEXAS AT AUSTIN


Universal Samplers
Traitor Tracing
Multiparty Key Exchange
Private Constrained PRFs
Deniable Encryption
Functional Encryption
RO uninstantiability
Trapdoor Permutations
INDISTINGUISHABILITY OBFUSCATION
[Garg-Gentry-Halevi-Raykova-Sahai-Waters13]
Garbled RAM
Circular Security separations
Witness Encryption
2-round MPC
PPAD hardness
FHE


[Cheon-Han-Lee-Ryu-Stehle15, ...]
MULTILINEAR MAPS
[Garg-Gentry-Halevi13]
[Garg-Gentry-Halevi-Raykova-Sahai-Waters13, …]
OBFUSCATION
STANDARD ASSUMPTIONS


OBFUSCATION
STANDARD ASSUMPTIONS

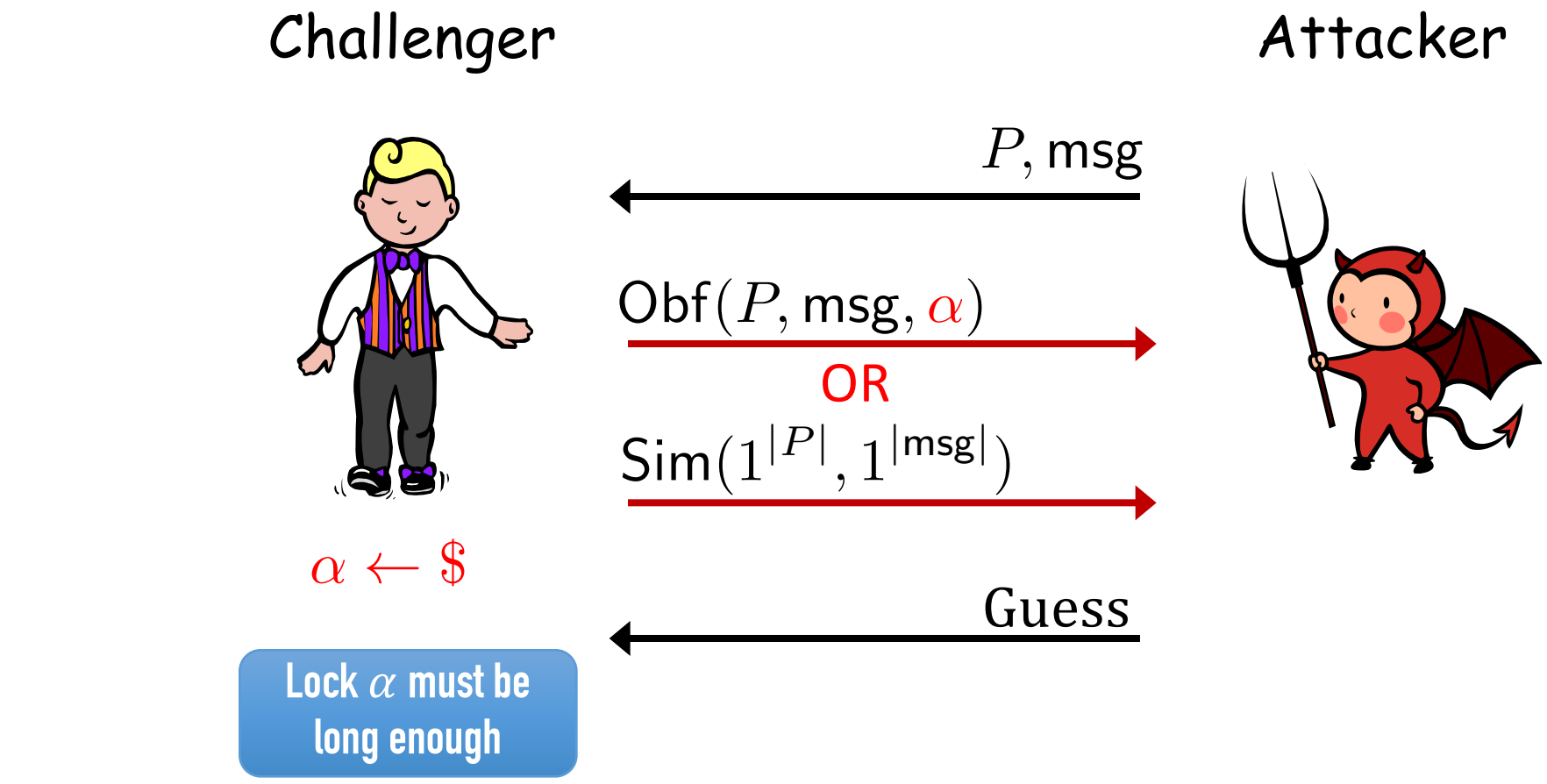

LOCKABLE
OBFUSCATION
OBFUSCATION
STANDARD ASSUMPTIONS

Lockable Obfuscation

- $\mathsf{Obf}(1^\lambda, P, \mathsf{msg}, \alpha) \rightarrow \widetilde{P}$
- $\mathsf{Eval}(\widetilde{P}, x) \rightarrow \mathsf{msg}$ or $\perp$

- Correctness:
  - If $P(x) = \alpha$, $\mathsf{Eval}(\widetilde{P}, x) = \mathsf{msg}$.
  - Else $\mathsf{Eval}(\widetilde{P}, x) = \perp$.

Lockable Obfuscation: Security



Our Result

α could be from high entropy dist

Concurrent Work
[WichsZirdelis17]

- Lockable Obfuscation
  - All poly sized circuits*
  - Secure under LWE

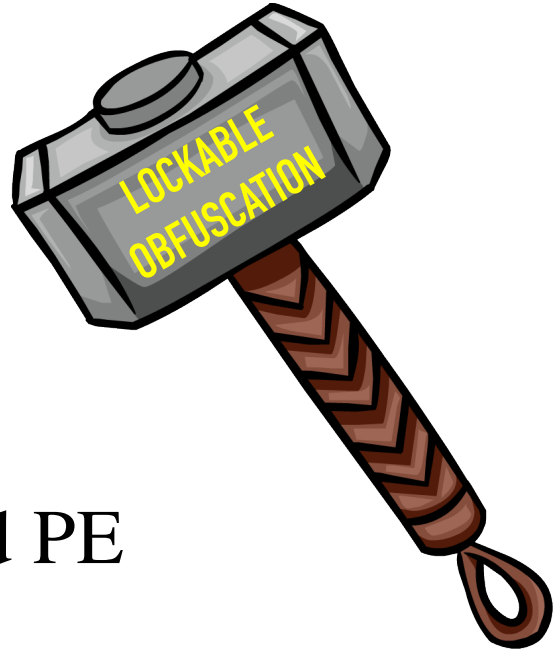
- Applications
  - Attribute-Based Encryption → 1-sided Predicate Encryption
  - Broadcast Encryption → Anonymous Broadcast Encryption
  - Witness Encryption → Reject Indistinguishability Obfuscator (riO)
  - Circular Security Separations (Bit Encryption, Unbounded, …)
  - Random Oracle Uninstantiability (Fujisaki-Okamoto, …)
  - ……

This Talk

Part I: Applications

Part II: Building Lockable Obfuscation from LWE

This Talk



Part I: Applications

- Upgrading ABE to 1-sided PE

Part II: Building Lockable Obfuscation from LWE

Attribute Based Encryption (ABE)

[SahaiWaters05]

Predicate Encryption (PE)

[KatzSahaiWaters08,BonehWaters07]

$\mathsf{ct} \leftarrow \mathsf{Enc}(\mathsf{mpk}, m, x)$ $\qquad$ $\mathsf{sk}_{f_1}, \ldots, \mathsf{sk}_{f_q}$

Lockable Obfuscation

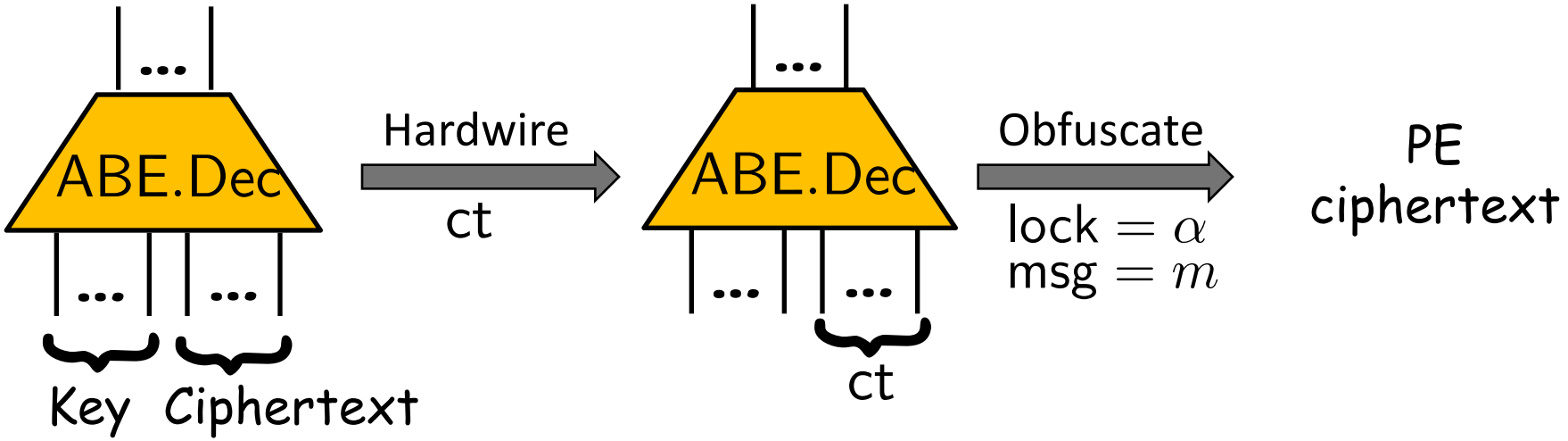
	Attribute Based Encryption	*1-sided* Predicate Encryption	Predicate Encryption
$\forall i,\ f_i(x) = 0$	m hidden x may not be	m hidden x hidden	m hidden x hidden
$\exists i,\ f_i(x) = 1$	x may not be	x may not be	x hidden

Upgrading ABE to “1-sided” PE

- Setup, KeyGen *unchanged*

Upgrading ABE to "1-sided" PE

- $Enc(\text{message } m, \text{attribute } x)$
  - Choose random lock α
  - Compute ct $\leftarrow$ ABE.Enc(α, x)

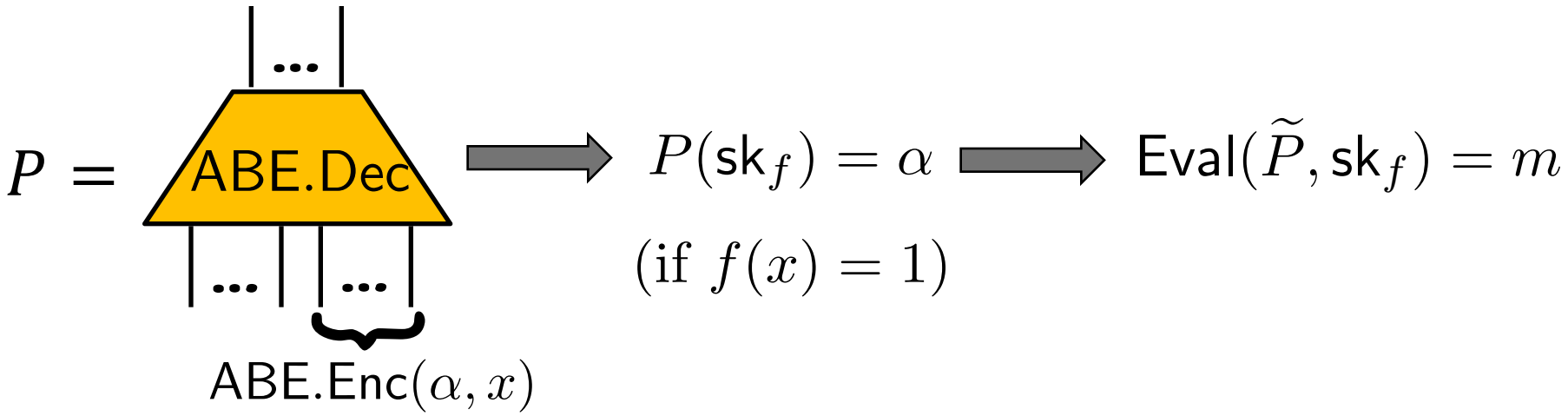


Upgrading ABE to "1-sided" PE

- *Enc*(message m, attribute x)
  - Choose random lock α
  - Compute $\mathsf{ct} \leftarrow \mathsf{ABE.Enc}(\alpha, x)$
  - Let $P = \mathsf{ABE.Dec}(\cdot, \mathsf{ct})$
  - Compute $\widetilde{P} \leftarrow \mathsf{Obf}(P, m, \alpha)$
  - Output $\widetilde{P}$

Upgrading ABE to "1-sided" PE

- $Dec(\text{key } sk_f, \text{ciphertext } \tilde{P})$
  - Output $\mathsf{Eval}(\widetilde{P}, \mathsf{sk}_f)$



Proof Overview

- Choose random lock α
- Compute $\mathsf{ct} \leftarrow \mathsf{ABE.Enc}(\alpha, x)$
- Let $P = \mathsf{ABE.Dec}(\cdot, \mathsf{ct})$
- Compute $\widetilde{P} \leftarrow \mathsf{Obf}(P, m, \alpha)$
- Output $\widetilde{P}$

$\approx$ (Using ABE security)

$\vdots$

- Compute $\mathsf{ct} \leftarrow \mathsf{ABE.Enc}(0, x)$

$\vdots$

Proof Overview

- Choose random lock α
- Compute $\mathsf{ct} \leftarrow \mathsf{ABE.Enc}(0, x)$
- Let $P = \mathsf{ABE.Dec}(\cdot, \mathsf{ct})$
- Compute $\widetilde{P} \leftarrow \mathsf{Obf}(P, m, \alpha)$
- Output $\widetilde{P}$

$\approx$ (Using Lockable Obfuscator)

- Output $\widetilde{P} \leftarrow \mathsf{Sim}(\cdots)$

1-Cycle Tester

- Setup : Choose key pair $(\mathsf{pk}, \mathsf{sk})$, string s
  Compute aux = obfuscation of

iO

Constants: Key sk, String s
Input: Ciphertext ct

– Compute $(m_1, m_2) = \mathsf{Dec}(\mathsf{sk}, \mathsf{ct})$.
– If $\mathrm{PRG}(m_2) = \mathrm{PRG}(s)$, output 1.

Lockable

$\alpha = s, \mathsf{msg} = 1$

Constants: Key sk
Input: Ciphertext ct

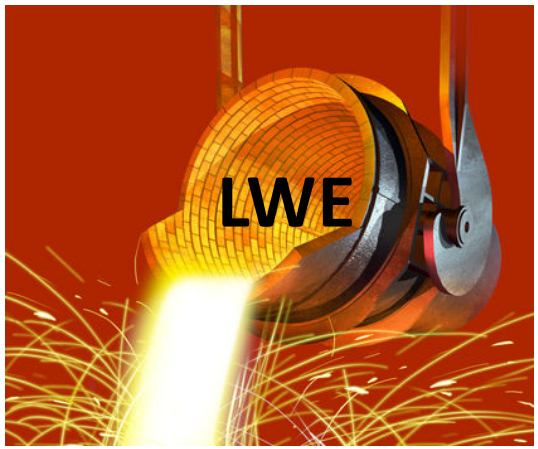
– Compute $(m_1, m_2) = \mathsf{Dec}(\mathsf{sk}, \mathsf{ct})$.
– Output m_2.

Output $\mathsf{pk}' = (\mathsf{pk}, \mathrm{aux}), \mathsf{sk}' = (\mathsf{sk}, s)$.

This Talk

Part I: Applications

Part II: Building Lockable Obfuscation from LWE



Lattice Trapdoors [GentryPeikertVaikuntanathan08,…]

$\texttt{TrapGen}(1^n, 1^m, q) \rightarrow (\mathbf{A}, T_{\mathbf{A}})$ $\quad\quad$ $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$

$\texttt{SamplePre}(\mathbf{A}, T_{\mathbf{A}}, \sigma, \mathbf{Z}) \rightarrow \mathbf{U}$ $\quad\quad$ $\mathbf{A} \cdot \mathbf{U} = \mathbf{Z}$ and $||\mathbf{U}||$ is small

Construction: Breaking It Down

- Step I: Log-depth circuits
  1-bit messages
- Step II: General circuits

Why log-depth?

Barrington's Theorem

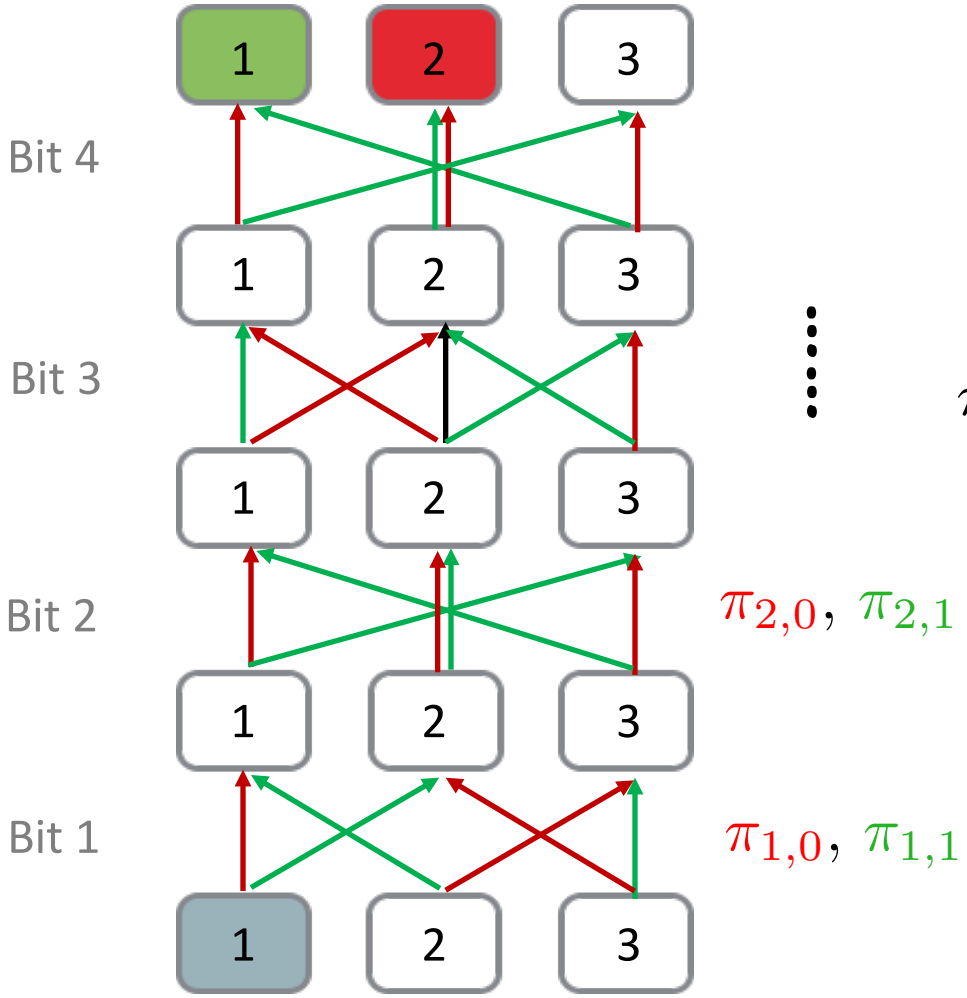
Log-depth circuits can be expressed as poly-length *permutation* branching programs of width 5.

Techniques from works on circular security

CIRCULAR SECURITY SEPARATIONS FOR SYMMETRIC-KEY BIT ENCRYPTION FROM LWE

G, KOPPULA, WATERS

Leveled Permutation Branching Programs



Read-once BP on 4-bit inputs

Say Width-3, Length-5

$\pi_{j,b} : [3] \to [3]$ specifies transitions

Simplifications

Read-once Permutation BPs

ℓ-bit input and ℓ-bit output

Length $\ell + 1$, Width 3

Fixed input-bit selector

"small" entries/ error terms → 0

Preview

Obfuscated Program: $\widetilde{P} = \begin{cases} \mathbf{M}_{1,1}^{(\ell)}, & (\mathbf{C}_{1,0}^{(\ell)}, \mathbf{C}_{1,1}^{(\ell)}), \ldots, (\mathbf{C}_{\ell,0}^{(\ell)}, \mathbf{C}_{\ell,1}^{(\ell)}) \\ \vdots & \\ \mathbf{M}_{1,1}^{(1)}, & (\mathbf{C}_{1,0}^{(1)}, \mathbf{C}_{1,1}^{(1)}), \ldots, (\mathbf{C}_{\ell,0}^{(1)}, \mathbf{C}_{\ell,1}^{(1)}) \end{cases}$

Encode BP computing i^{th} output bit

Starting state

State Transitions

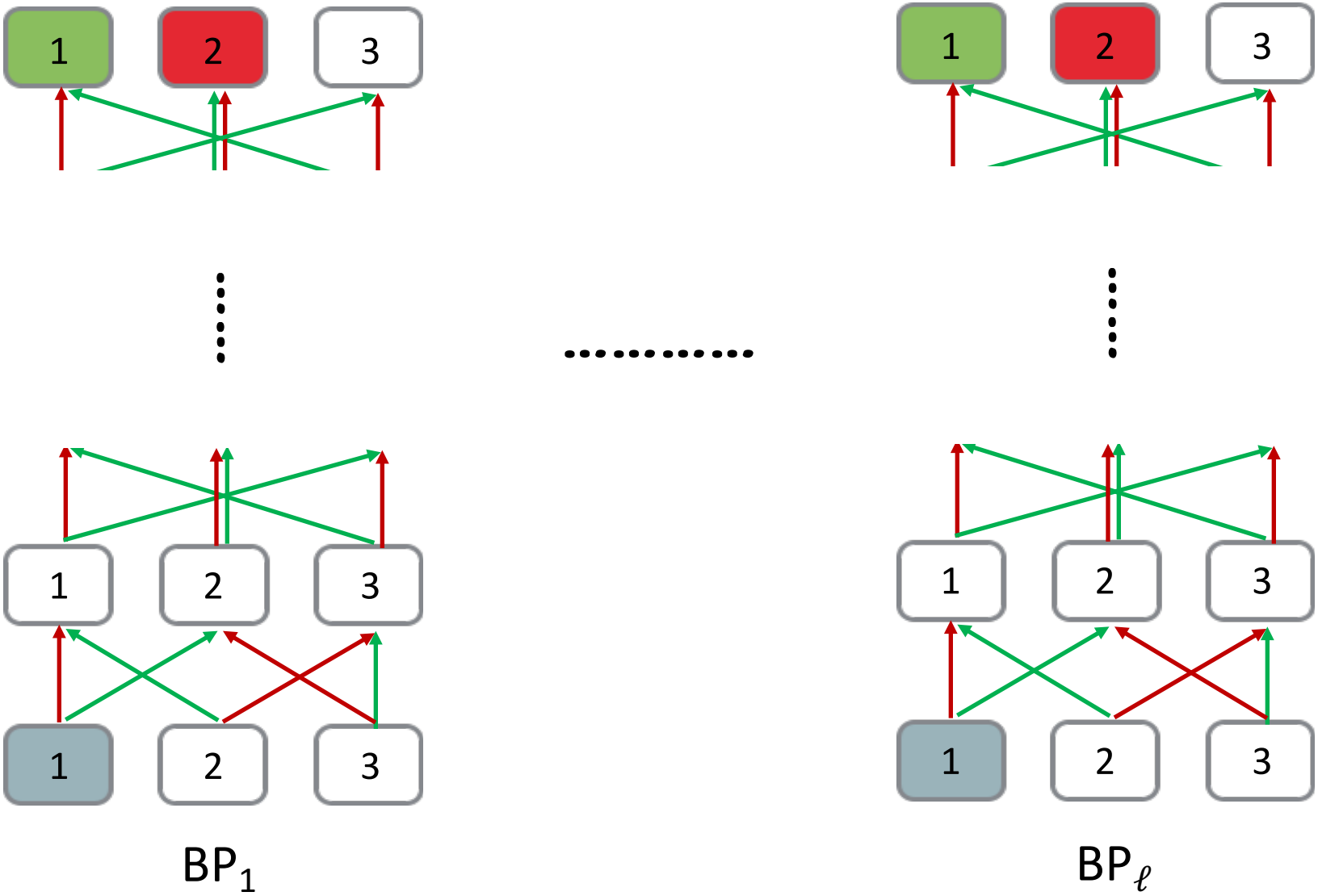
$$\mathbf{M}_{1,1}^{(i)}, \quad (\mathbf{C}_{1,0}^{(i)}, \mathbf{C}_{1,1}^{(i)}), \ldots, (\mathbf{C}_{\ell,0}^{(i)}, \mathbf{C}_{\ell,1}^{(i)})$$

$$\boxed{\mathbf{M}_{1,1}^{(i)}} \times \boxed{\mathbf{C}_{1,x_1}^{(i)}} \times \boxed{\mathbf{C}_{2,x_2}^{(i)}} \cdots\cdots\cdots \boxed{\mathbf{C}_{\ell,x_\ell}^{(i)}}$$

Sum & Test

Obf(P,α,msg)

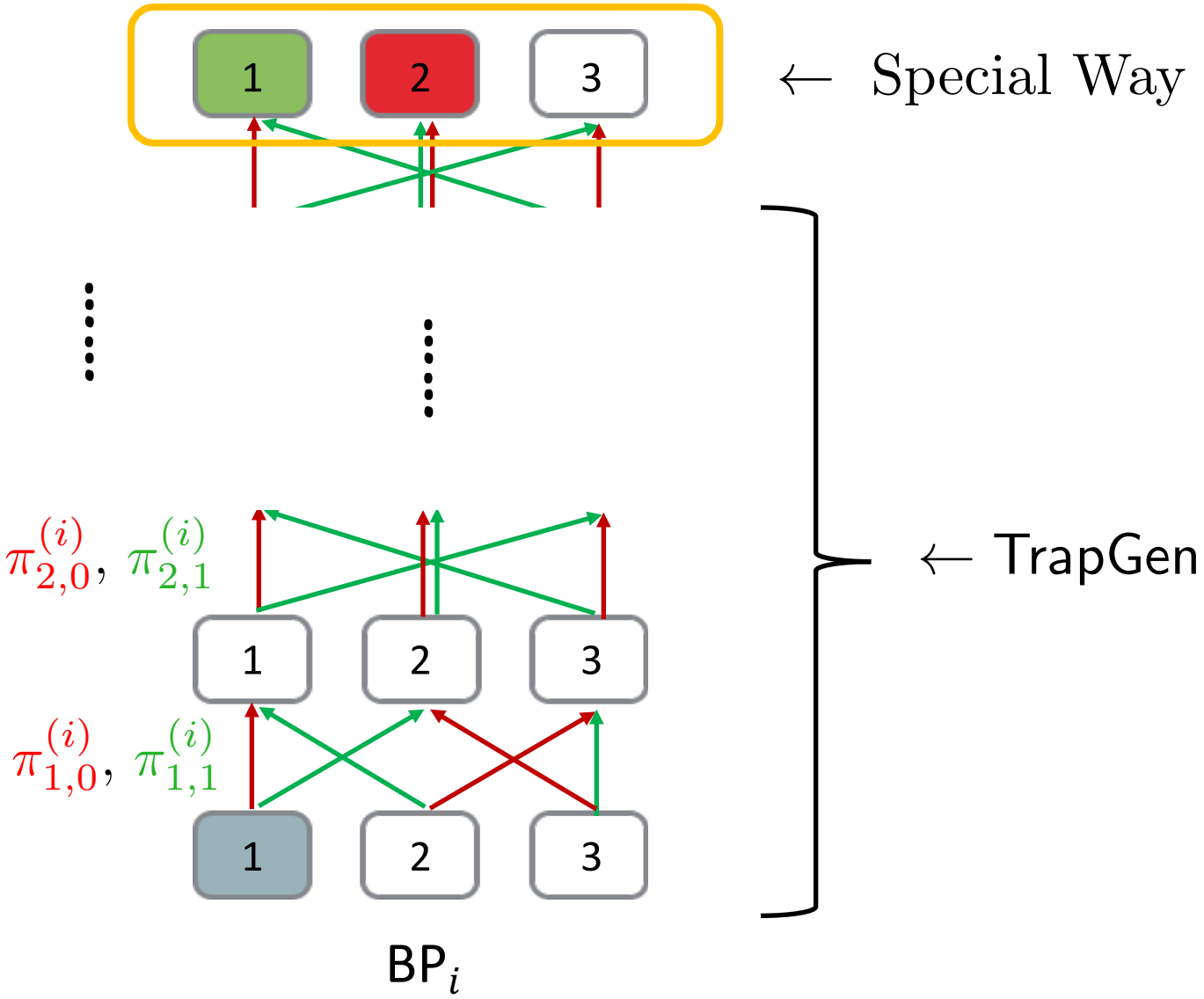
Let $P(x) = \mathsf{BP}_1(x)||\cdots||\mathsf{BP}_\ell(x)$



Obf(P,α,msg)

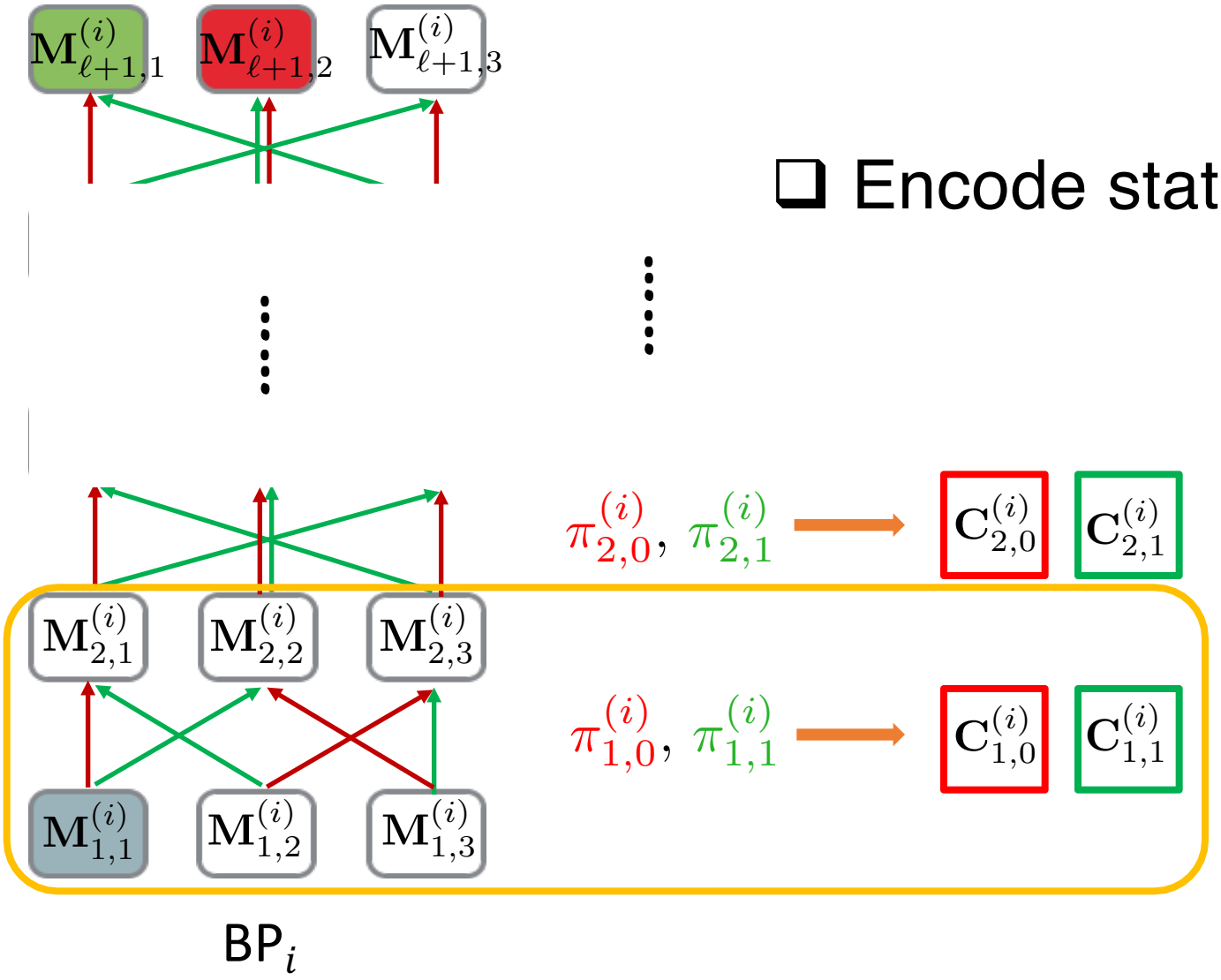
$$\text{Let } P(x) = \mathsf{BP}_1(x)||\cdots||\mathsf{BP}_\ell(x)$$

$$\mathsf{BP}_i(x) = (\pi^{(i)}_{1,0}, \pi^{(i)}_{1,1}), \ldots, (\pi^{(i)}_{\ell,0}, \pi^{(i)}_{\ell,1})$$



Obf(P,α,msg)

$$\mathsf{BP}_i(x) = (\pi^{(i)}_{1,0}, \pi^{(i)}_{1,1}), \ldots, (\pi^{(i)}_{\ell,0}, \pi^{(i)}_{\ell,1})$$

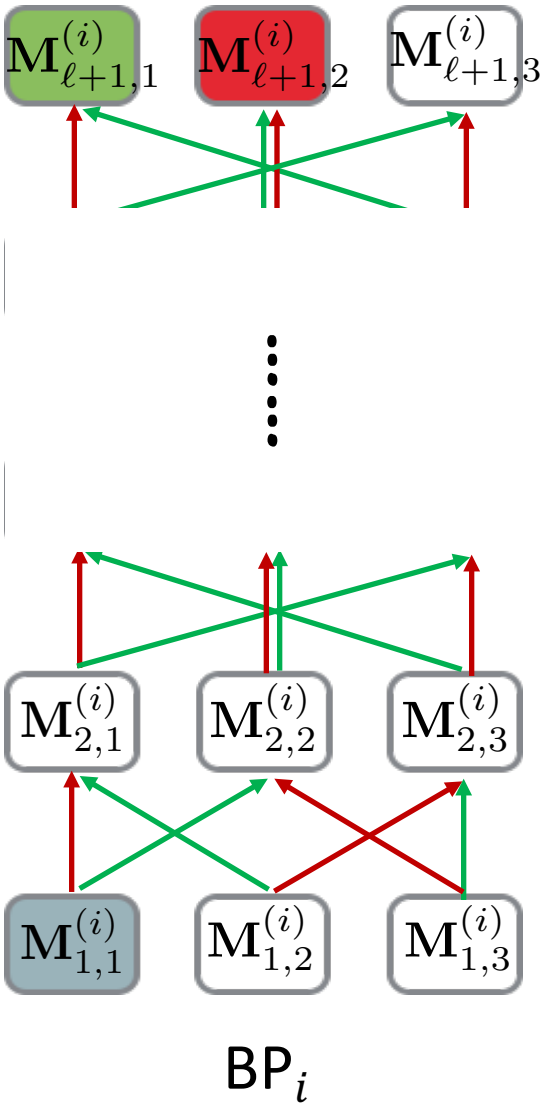


BP_i

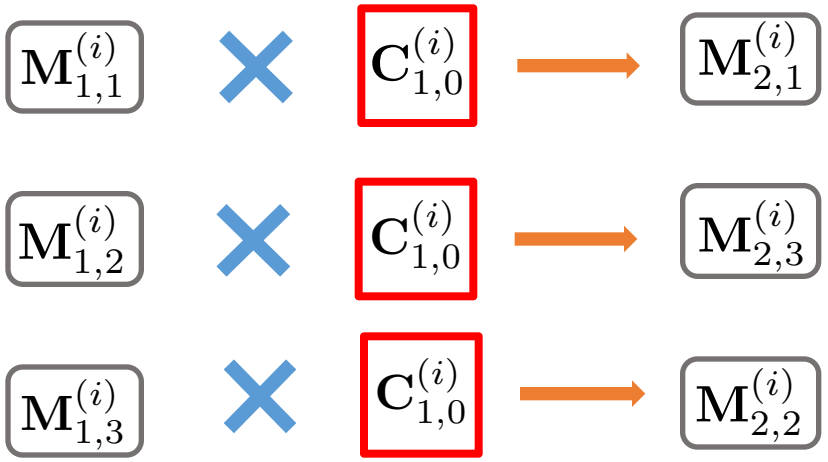
- Encode state transitions

Obf(P,α,msg)

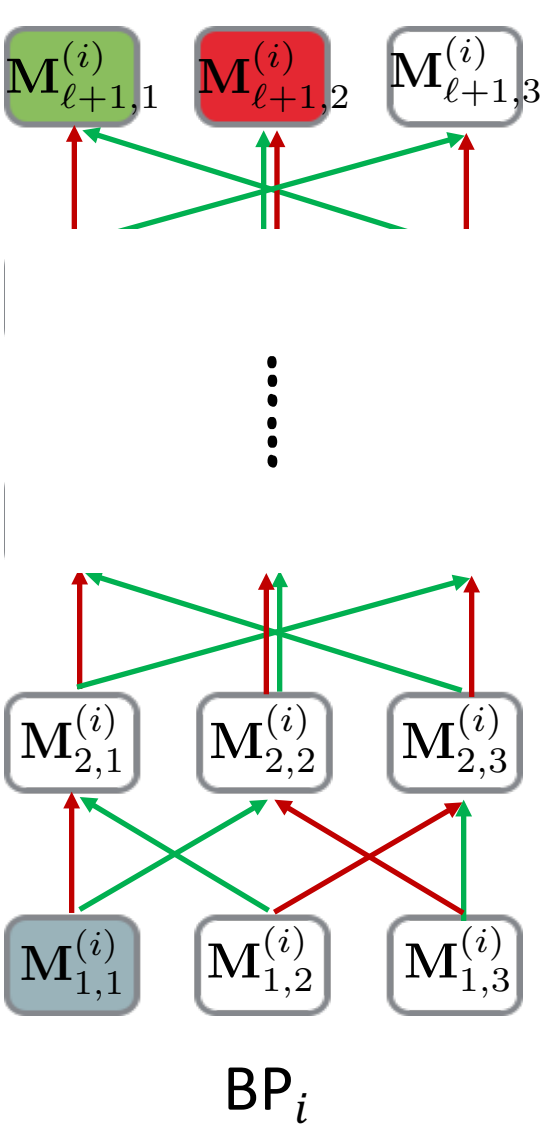
$$\mathsf{BP}_i(x) = (\pi_{1,0}^{(i)}, \pi_{1,1}^{(i)}), \ldots, (\pi_{\ell,0}^{(i)}, \pi_{\ell,1}^{(i)})$$



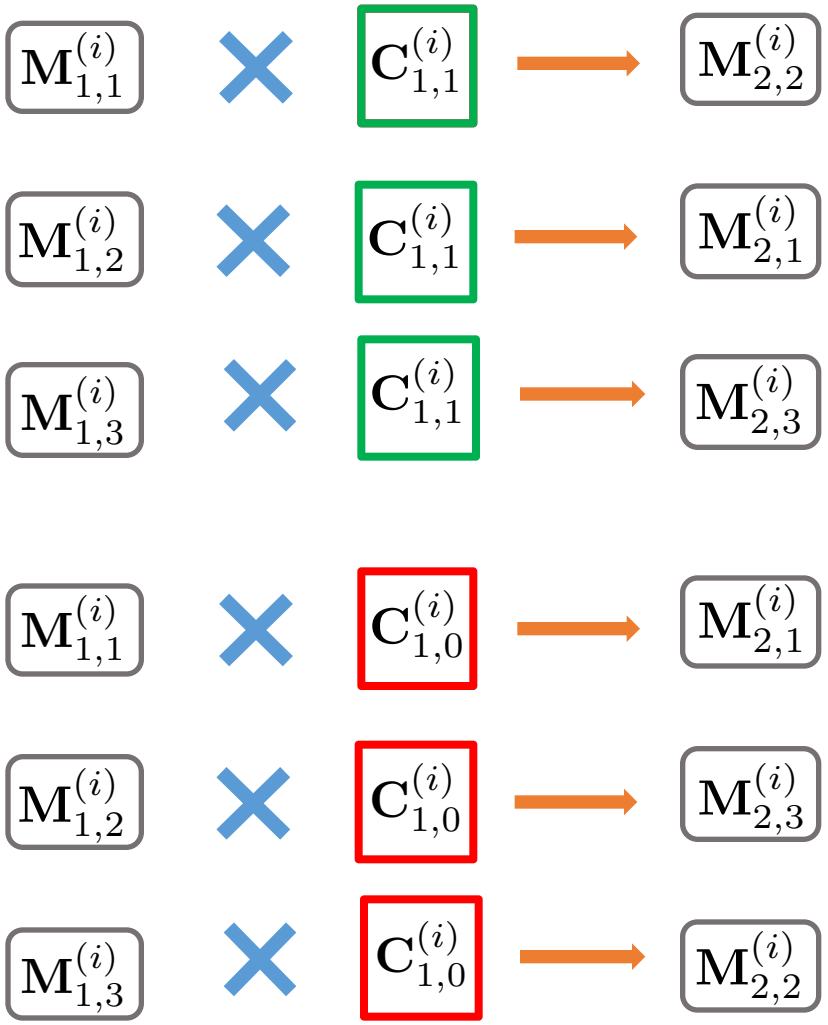
BP_i



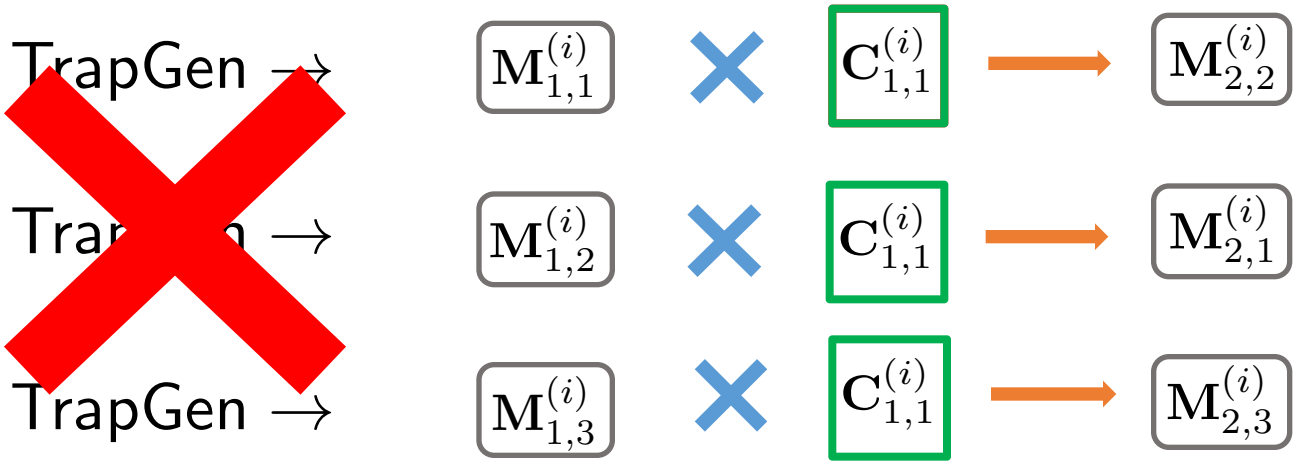
Obf(P,α,msg)



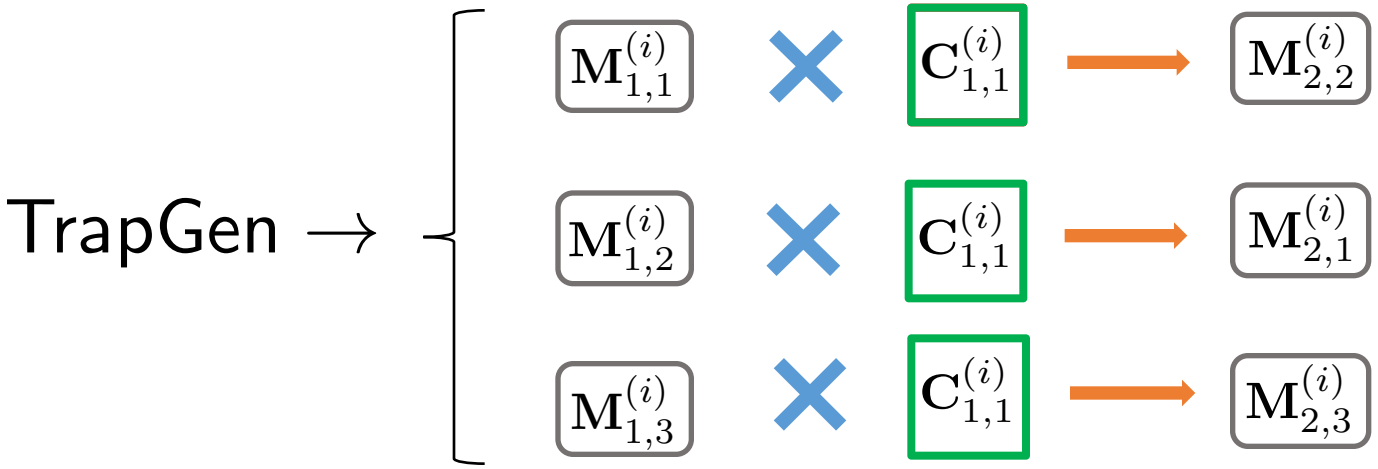
BP_i



How to SamplePre?



How to SamplePre?



✓ Joint trapdoor generation at *each level* for every BP_i

NOTE: Cannot jointly sample for all BP_i together (Parameter issue)

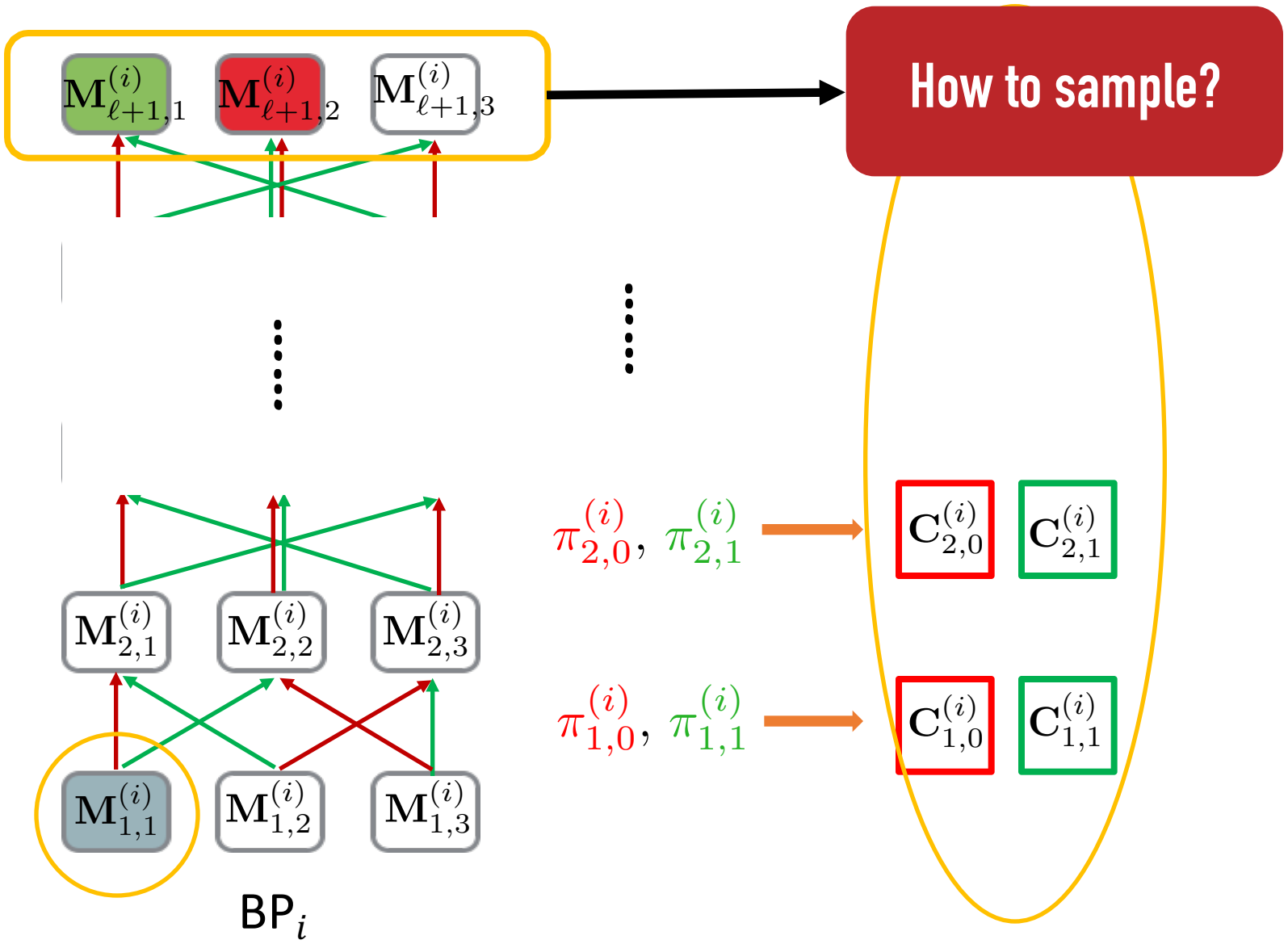
For Security

- Choose short secrets $\mathbf{S}_{j,0}, \mathbf{S}_{j,1}$ Per level j BUT shared across BP_i

$$\begin{bmatrix} \mathbf{M}^{(i)}_{1,1} \\ \mathbf{M}^{(i)}_{1,2} \\ \mathbf{M}^{(i)}_{1,3} \end{bmatrix} \xrightarrow{\mathbf{C}^{(i)}_{1,0}} \begin{bmatrix} \mathbf{M}^{(i)}_{2,\pi^{(i)}_{1,0}(1)} \\ \mathbf{M}^{(i)}_{2,\pi^{(i)}_{1,0}(2)} \\ \mathbf{M}^{(i)}_{2,\pi^{(i)}_{1,0}(3)} \end{bmatrix} \quad \begin{bmatrix} \mathbf{S}_{1,0} \cdot \mathbf{M}^{(i)}_{2,\pi^{(i)}_{1,0}(1)} \\ \mathbf{S}_{1,0} \cdot \mathbf{M}^{(i)}_{2,\pi^{(i)}_{1,0}(2)} \\ \mathbf{S}_{1,0} \cdot \mathbf{M}^{(i)}_{2,\pi^{(i)}_{1,0}(3)} \end{bmatrix} + \text{noise}$$

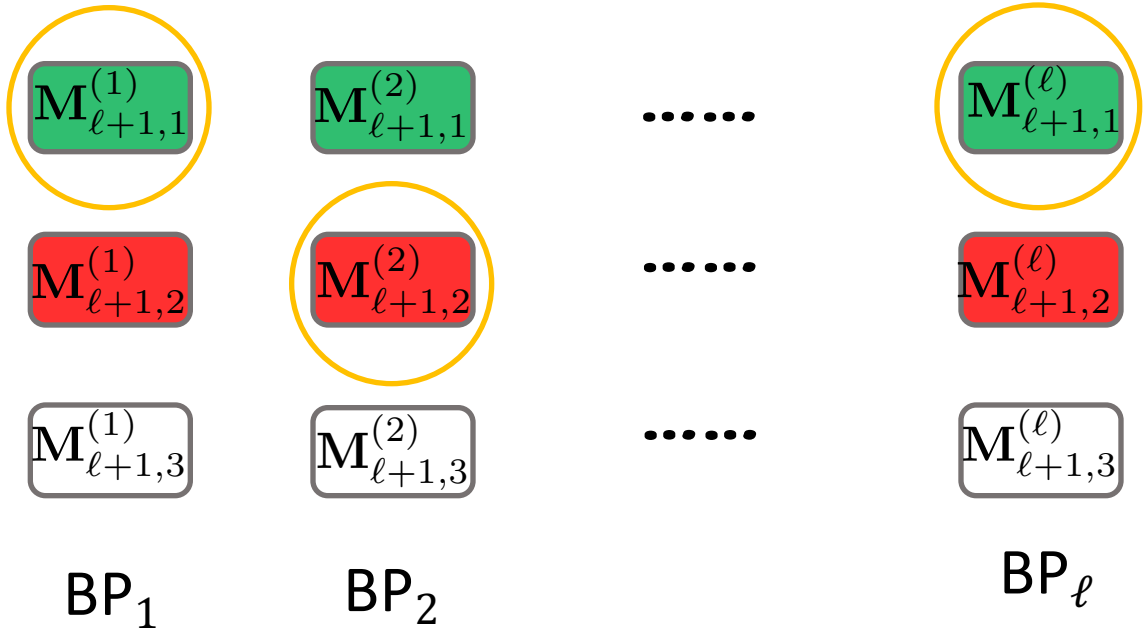
$$\begin{bmatrix} \mathbf{M}^{(i)}_{1,1} \\ \mathbf{M}^{(i)}_{1,2} \\ \mathbf{M}^{(i)}_{1,3} \end{bmatrix} \xrightarrow{\mathbf{C}^{(i)}_{1,1}} \begin{bmatrix} \mathbf{M}^{(i)}_{2,\pi^{(i)}_{1,1}(1)} \\ \mathbf{M}^{(i)}_{2,\pi^{(i)}_{1,1}(2)} \\ \mathbf{M}^{(i)}_{2,\pi^{(i)}_{1,1}(3)} \end{bmatrix} \quad \begin{bmatrix} \mathbf{S}_{1,1} \cdot \mathbf{M}^{(i)}_{2,\pi^{(i)}_{1,1}(1)} \\ \mathbf{S}_{1,1} \cdot \mathbf{M}^{(i)}_{2,\pi^{(i)}_{1,1}(2)} \\ \mathbf{S}_{1,1} \cdot \mathbf{M}^{(i)}_{2,\pi^{(i)}_{1,1}(3)} \end{bmatrix} + \text{noise}$$

Obf(P,α,msg)



Encoding Lock α and msg

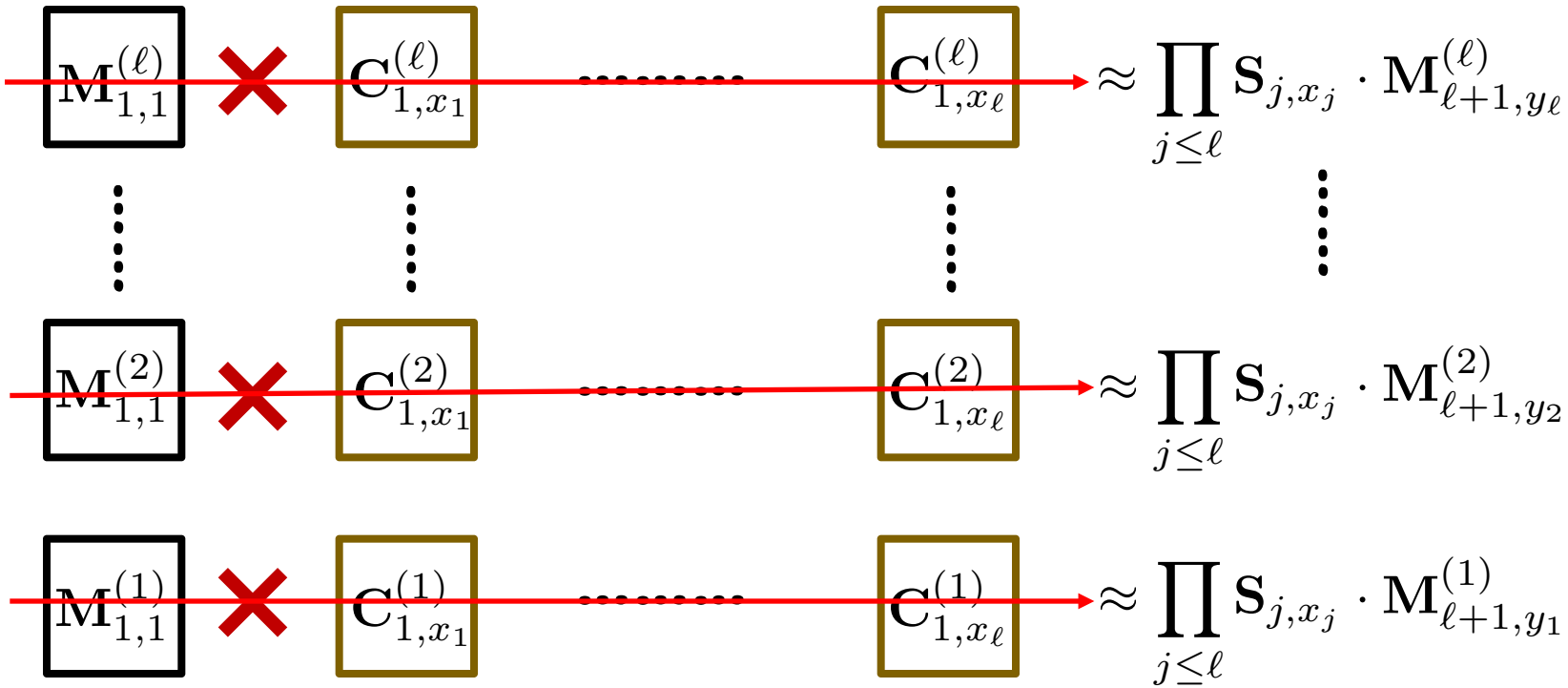
Say $\alpha = 10 \cdots 1$



$$\sum_{i \le \ell} \mathbf{M}^{(i)}_{\ell+1,\alpha_i} = \begin{cases} \mathbf{0} & \text{if } \mathsf{msg} = 0 \\ \sqrt{q} \cdot \mathbf{I} & \text{otherwise} \end{cases}$$

$\mathsf{Eval}(\widetilde{\mathrm{P}}, x)$

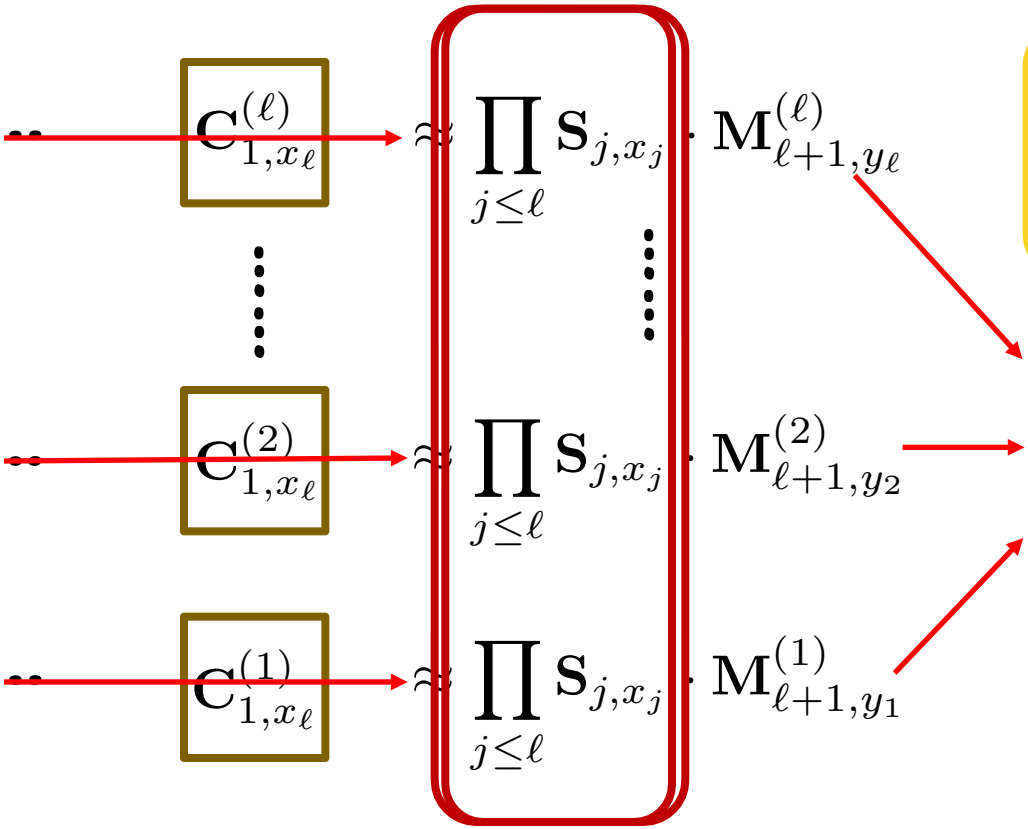
Let $y = P(x)$



If all entries 'small', output 0;
all entries 'medium', output 1;
otherwise output $\perp$

$\mathsf{Eval}(\widetilde{\mathrm{P}}, x)$

Let $y = P(x)$



If all entries 'small', output 0;
all entries 'medium', output 1;
otherwise output $\perp$

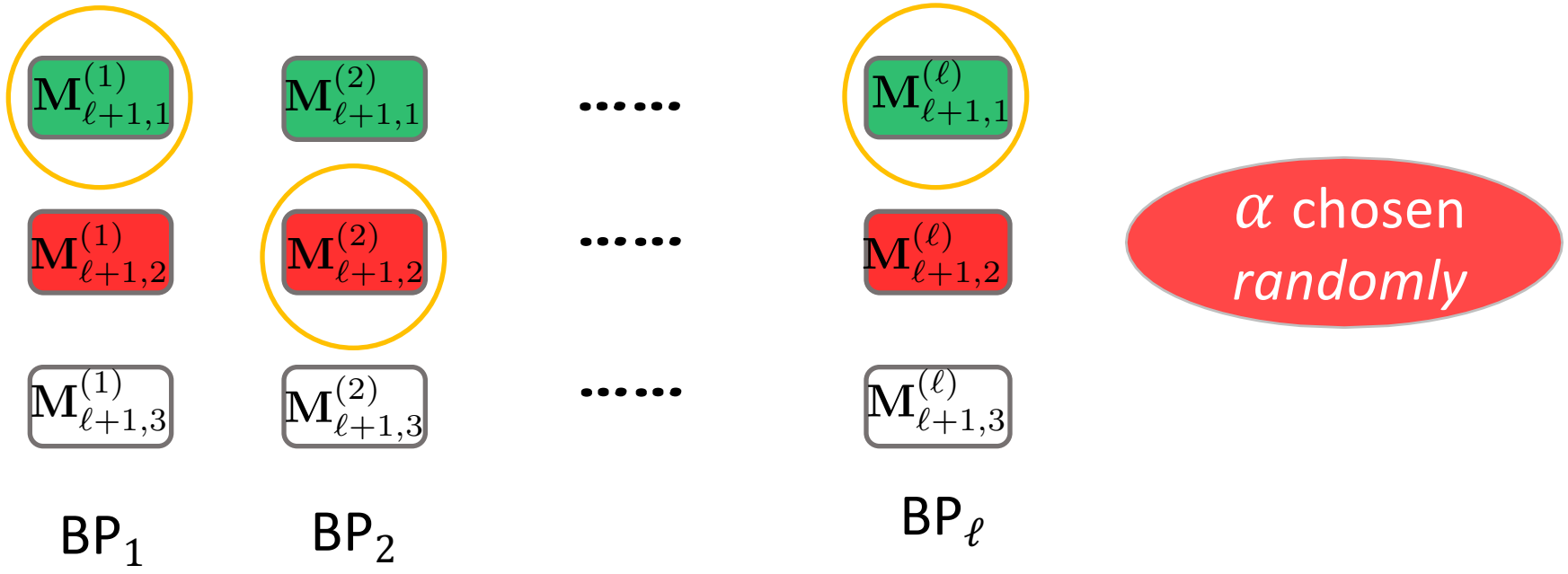
$$+ \approx \prod_{j\le\ell} \mathbf{S}_{j,x_j} \cdot \sum_{i\le\ell} \mathbf{M}^{(i)}_{\ell+1,y_i}$$

If $y = \alpha$

$$= \begin{cases} \mathbf{0} & \text{if msg} = 0 \\ \sqrt{q} \cdot \mathbf{I} & \text{otherwise} \end{cases}$$

BP_i share input-selector
S matrices reused at each level

Proof Sketch



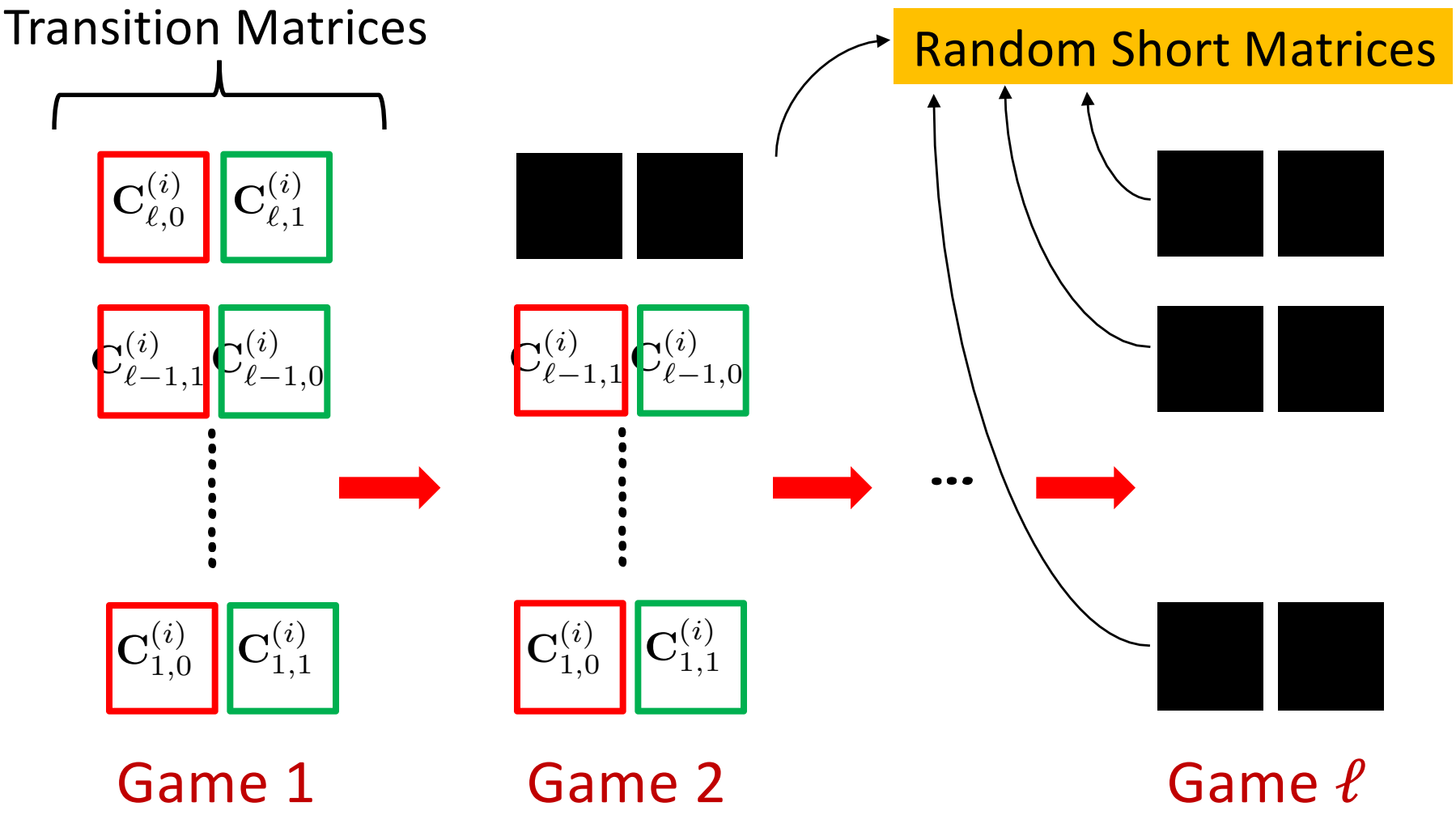
$$\sum_{i\le\ell} \mathbf{M}^{(i)}_{\ell+1,\alpha_i} = \begin{cases} \mathbf{0} & \text{if msg} = 0 \\ \sqrt{q}\cdot\mathbf{I} & \text{otherwise} \end{cases}$$

LHL

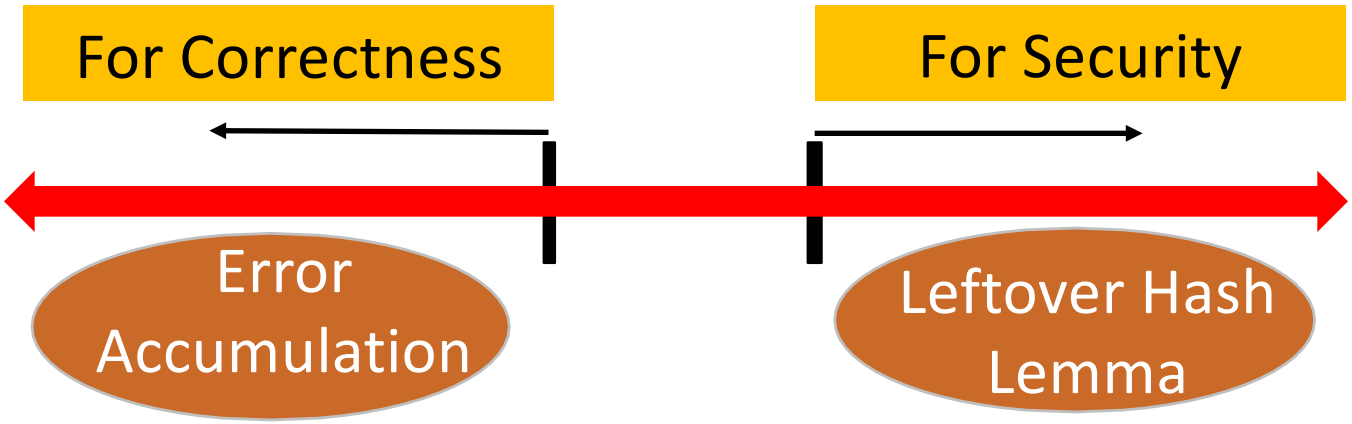
$$\mathbf{M}^{(i)}_{\ell+1,v} \leftarrow \mathbb{Z}_q^{n\times m}$$

Proof Sketch

Trapdoor Properties
LWE



Setting Parameters!?



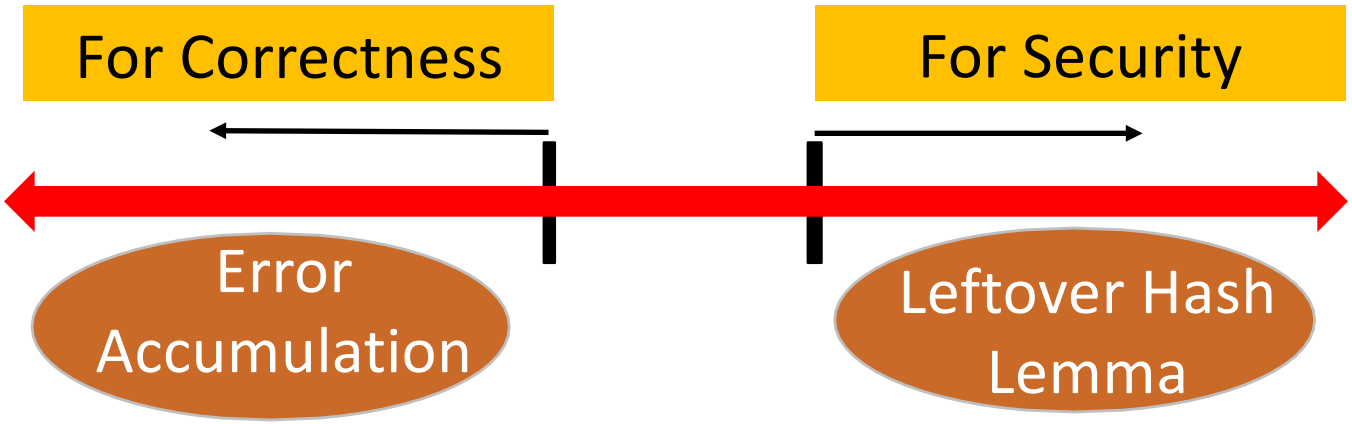
Problem.

For LHL: Output Length **>** $log(q)$.

Error Accumulation: *#Levels* **<** $log(q)$.

Currently: Output Length **=** #Levels.

Setting Parameters!?



Problem.

For LHL: Output Length **>** $log(q)$.

Error Accumulation: *#Levels* **<** $log(q)$.

~~Currently: Output Length = #Levels.~~

Obfuscate PRG(P(·)) instead!

Summing it all up

- Removing fixed-input selector restriction
  - Pad branching programs
- Relieving parameter tension
  - Compose with a NC^1 PRG
- Log-depth => poly-size circuits
  - Use LHE for bootstrapping
- Extending to multi-bit messages
  - See paper

Concluding Remarks

- More applications of lockable obfuscation?
  - [BadrinarayananKhuranaSahaiWaters18]: CCA-secure non-FE compatible scheme
- Stronger notions of obfuscation from standard assumptions?
- Direct construction for circuits and other models of computation?
  - [ChenVaikuntanathanWee18]: Lockable obf. for *non-permutation* BPs

Thanks!