Quasilinear Indistinguishability Obfuscation

via

Propositional Proofs of Equivalence

Elaine Shi

Joint work with Yaohua Ma, Chenxin Dai

Constant round concurrent ZK

Succinct garbled RAM

SNARG

Functional encryption

2-round MPC

Witness encryption

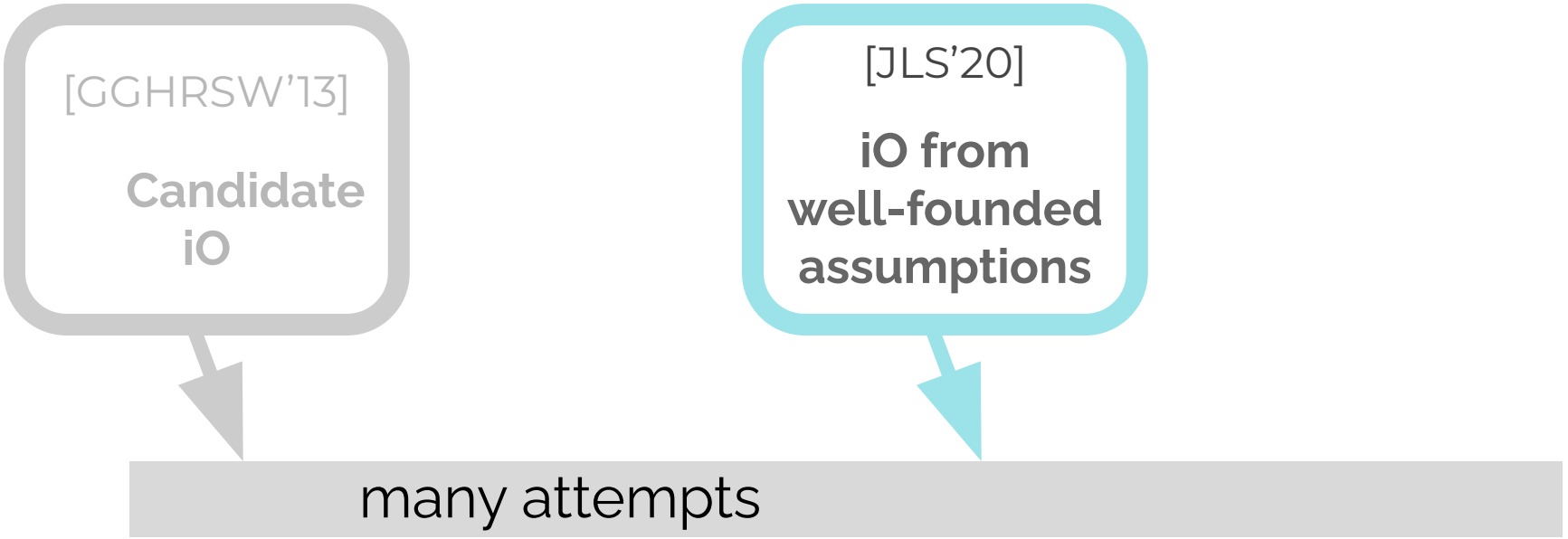
Deniable encryption

Universal samplers

Indistinguishability Obfuscation

[GGHRSW'13]

Candidate iO



Feasibility of provably secure iO?

Efficiency of iO ?

Provably secure iO

- Polynomial blowup
- Input-length barrier?

Efficiency of iO ?

Provably secure iO

- Polynomial blowup
- Input-length barrier?

Heuristic iO

- Quasilinear efficiency
  - VBB-obf for PRF ⇒ full obf [Applebaum]
  - VBB-obf + SNARG ⇒ full obf [Boneh's talk]

Efficiency of iO ?

Provably secure iO

- Polynomial blowup
- Input-length barrier?

Heuristic iO

- Quasilinear efficiency
  - VBB-obf for PRF ⇒ full obf [Applebaum]
  - VBB-obf + SNARG ⇒ full obf [Boneh's talk]

Can we have the best of both worlds?

Provably secure $i\mathcal{O}^{\mathcal{EF}}$


Quasilinear efficiency

Our Result

Provably secure $iO^{\mathcal{EF}}$

Quasilinear efficiency

via

Propositional proof of equivalence

Our Result

- FHE, poly (or subexp) secure
- OWF, subexp secure
- iO for $\widetilde{O}_\lambda(1)$-size circuits

$\Rightarrow$

$i\mathcal{O}^{\mathcal{FF}}$, poly (or sub-exp) secure, with

$\widetilde{O}_\lambda(N_{\text{circ}} + N_{\text{proof}})$ obf, and eval time

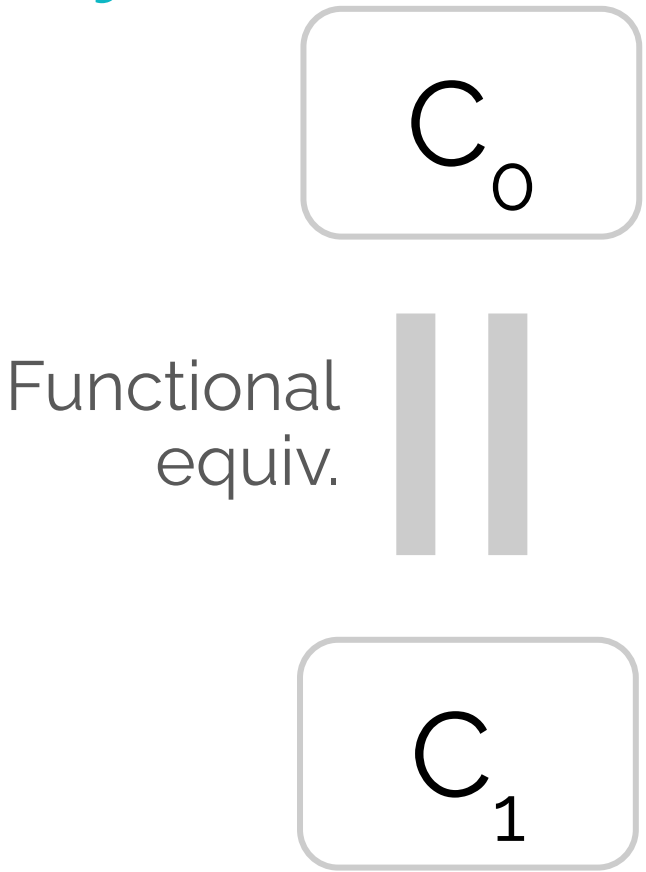
Our Result

What is

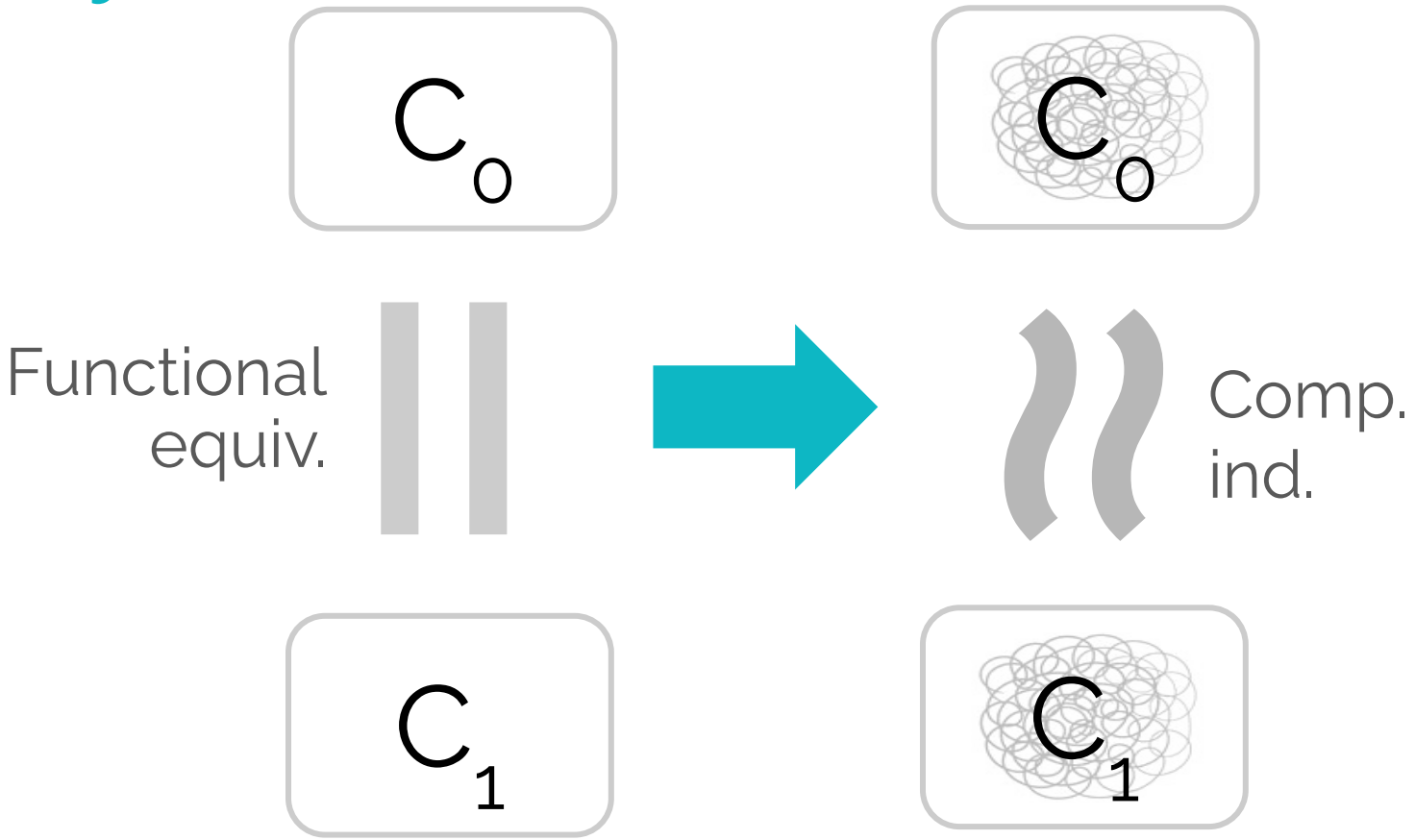
Propositional proof of equivalence

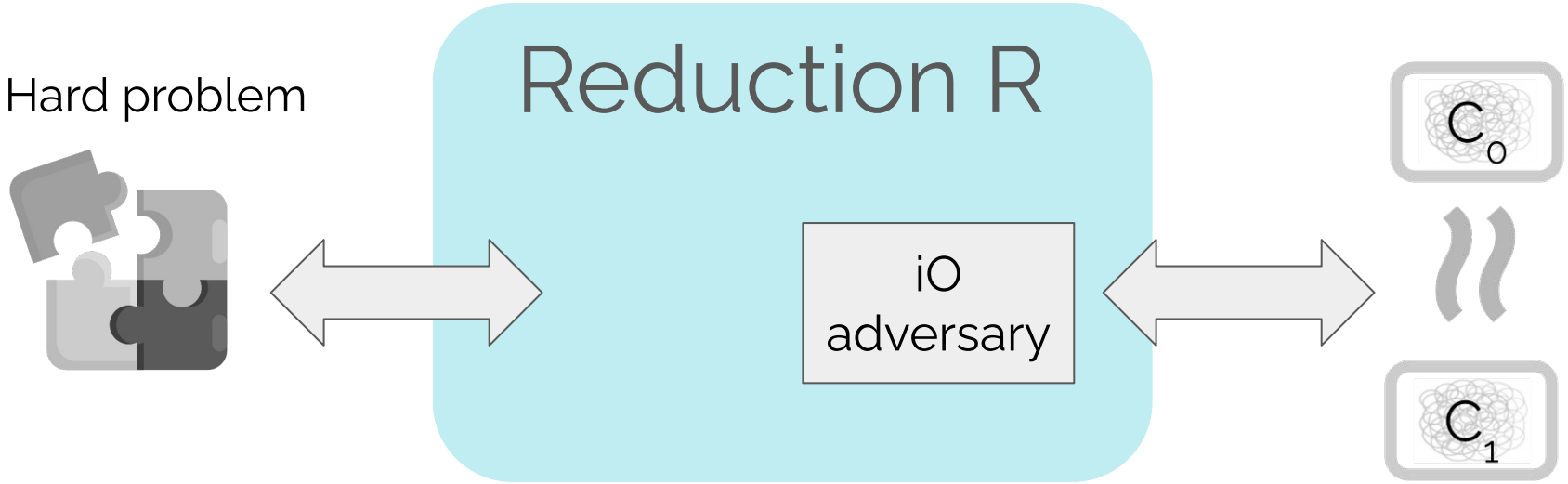
and why?

iO Security



iO Security




Hard problem
Reduction R
iO
adversary
C_0
C_1

Conjecture: R “knows” proof of equiv

Conjecture: R “knows” proof of equiv

Conjecture: R "knows" proof of equiv

$\Longrightarrow$ R must enumerate all inputs

Conjecture: R “knows” proof of equiv

$\Longrightarrow$ R must enumerate all inputs

$\Longrightarrow$ # hybrids exponential in input len

Not a formal proof

Conjecture: R "knows" proof of equiv

$\Longrightarrow$ R must enumerate all inputs

$\Longrightarrow$ # hybrids exponential in input len

$\Longrightarrow$ sec param ≥ poly(input len)

Not a formal proof

Conjecture: R "knows" proof of equiv

$\Longrightarrow$ R must enumerate all inputs

$\Longrightarrow$ # hybrids exponential in input len

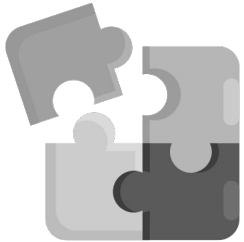
$\Longrightarrow$ sec param ≥ poly(input len)

"folklore:
The input length barrier"

Overcoming the input-len barrier

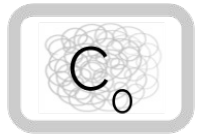
Proof: $C_0 = C_1$

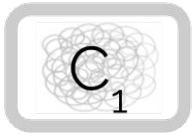
Hard problem



Reduction R

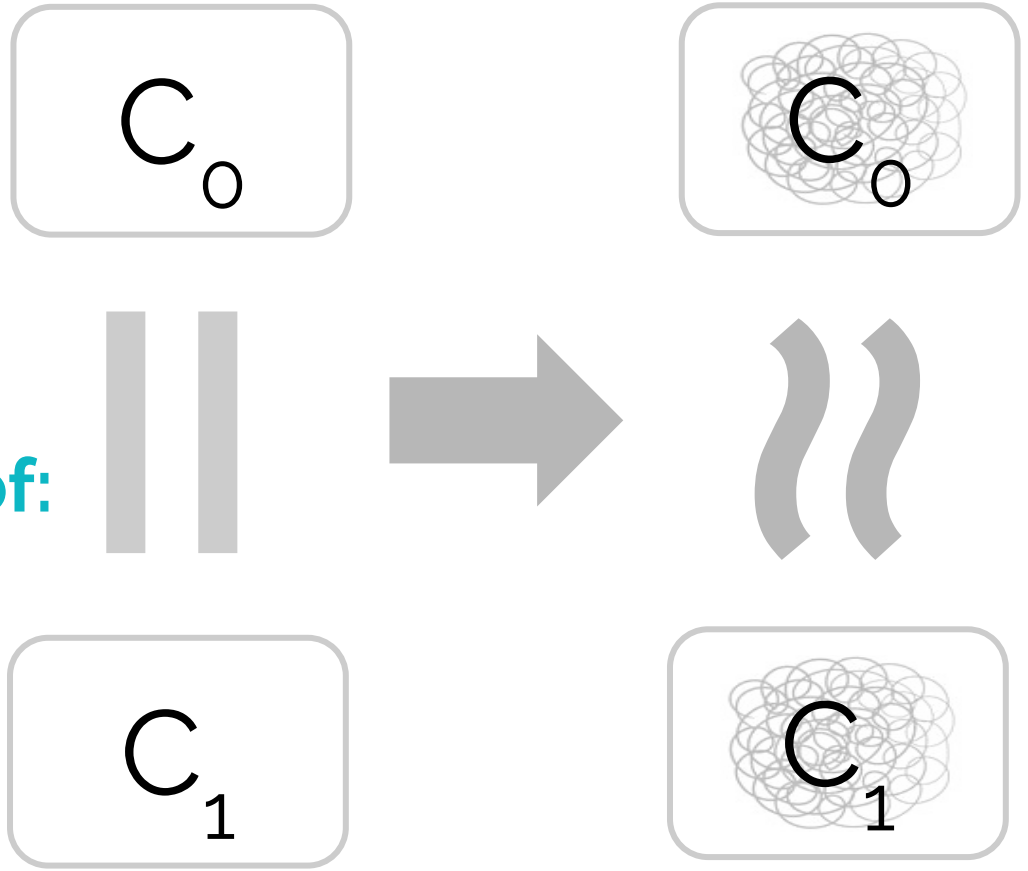
iO adversary





Relaxed iO security

Short proof:
$C_0 = C_1$



Expressive power of relaxed notion ?

> “ 
>
> Sufficient for almost all known applications of iO ”

's blueprint

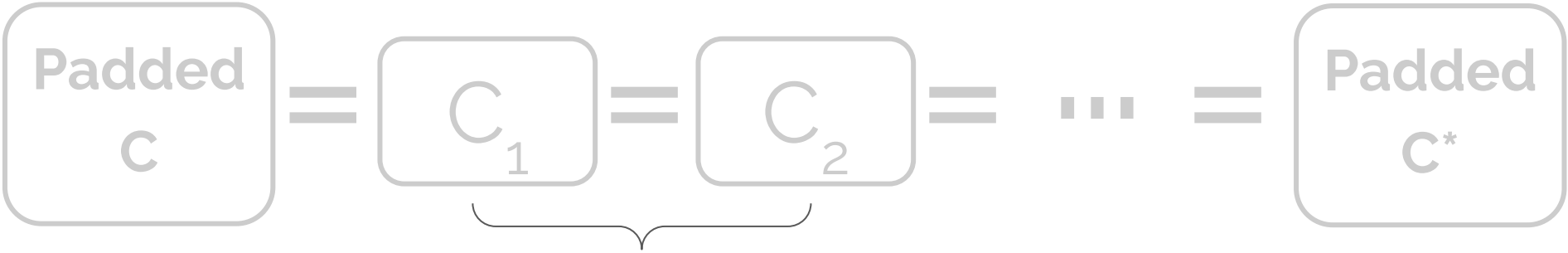
Short proof: **C** = **C***

Padded C = C_1 = C_2 = ... = **Padded C***

Identical except an O(log n)-sized subcircuit

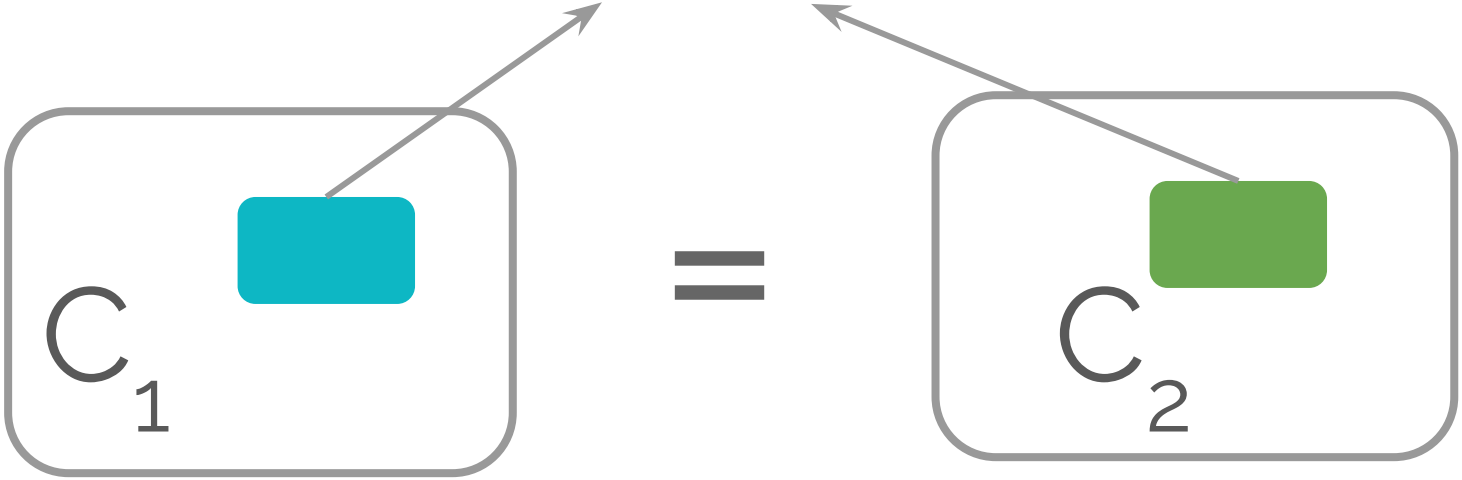
's blueprint

Short proof: **C** = **C***

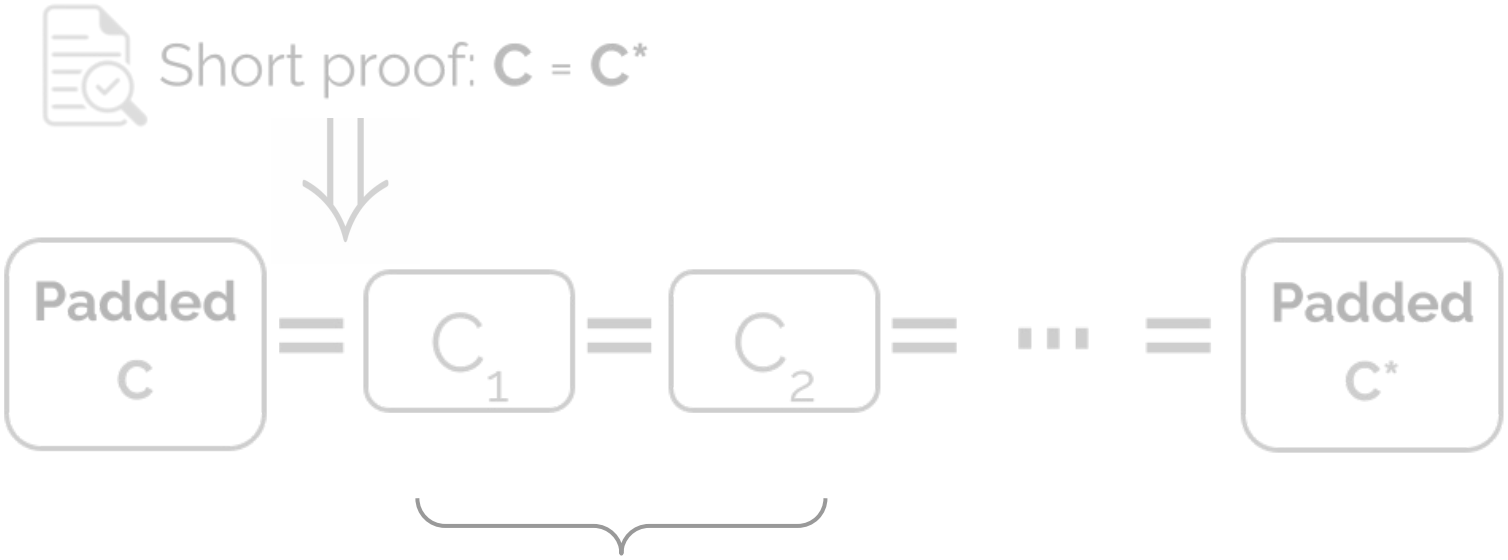


O(log n)-equivalent

Functionally equiv, differing in impl
O(log n) size

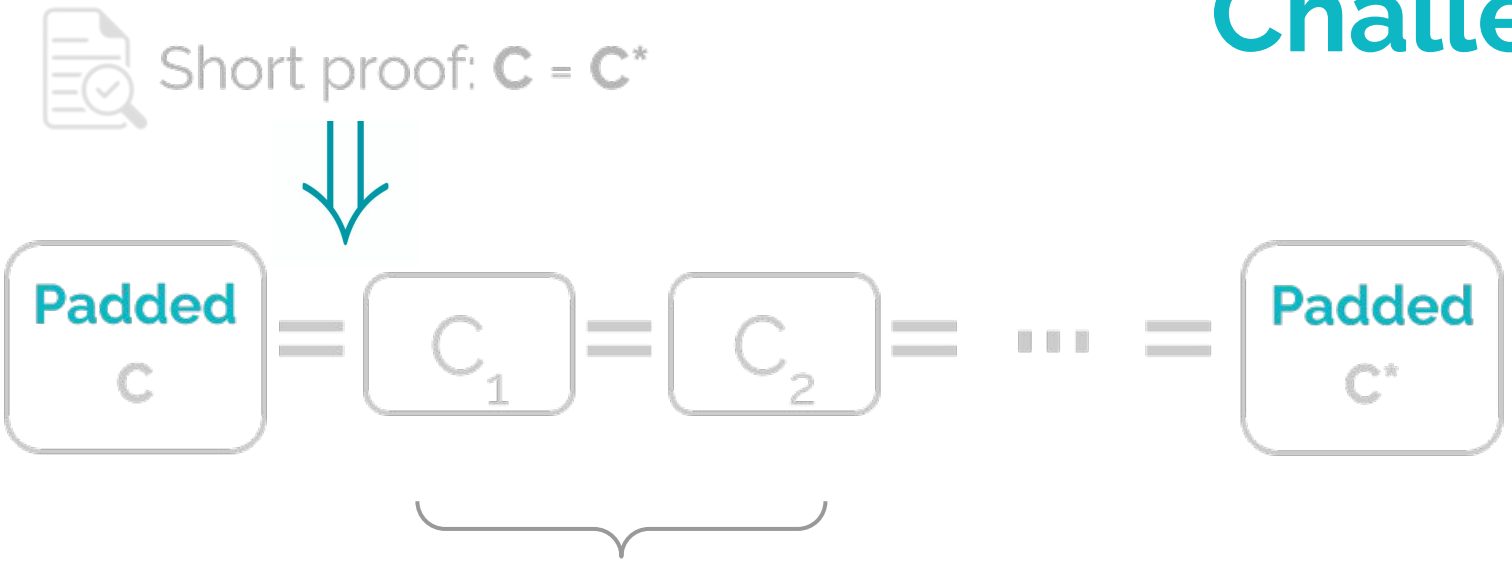


O(log n)-equivalent circuits



iO for O(log n)-equivalent circuits suffices

Challenge 1



iO for O(log n)-equivalent circuits

Challenge 2

's approach

n gates

padding

n^2 gates

iO

$\geq n^4$ eval time

assume: BARG with
ideal efficiency (not known)

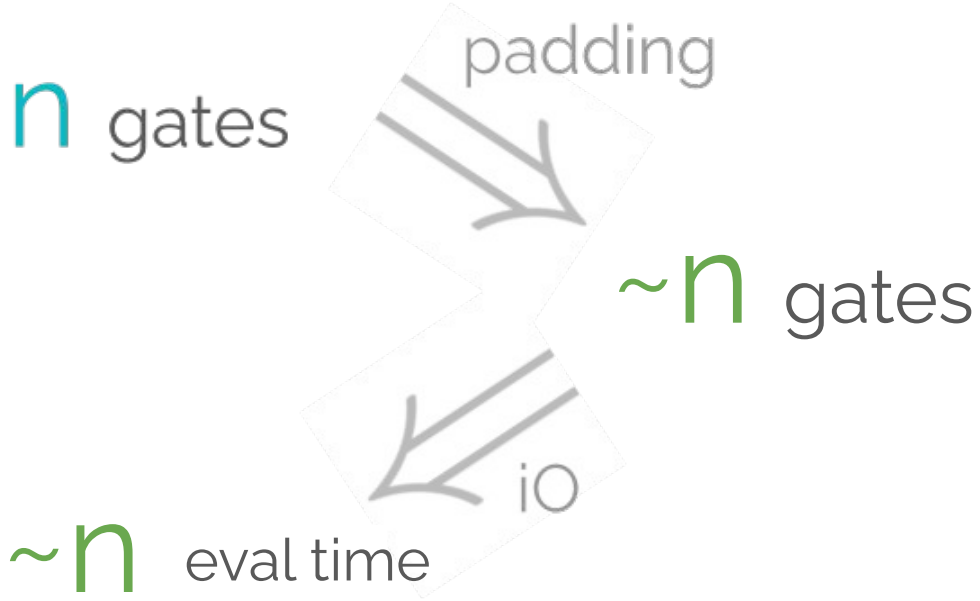
Challenge 1

padding

iO for log-equiv circ

Challenge 2

Our approach



assume: proof size ~ circuit size

Challenge 1

padding

iO for log-equiv circ

Challenge 2

Challenge 1

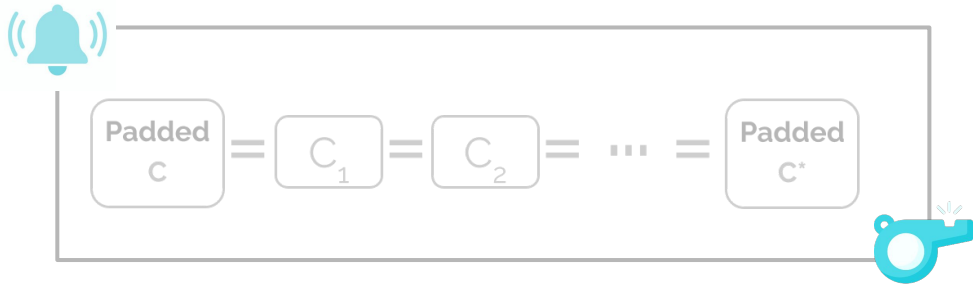
padding

Challenge 1

padding

Padded circuit is a **universal circuit**

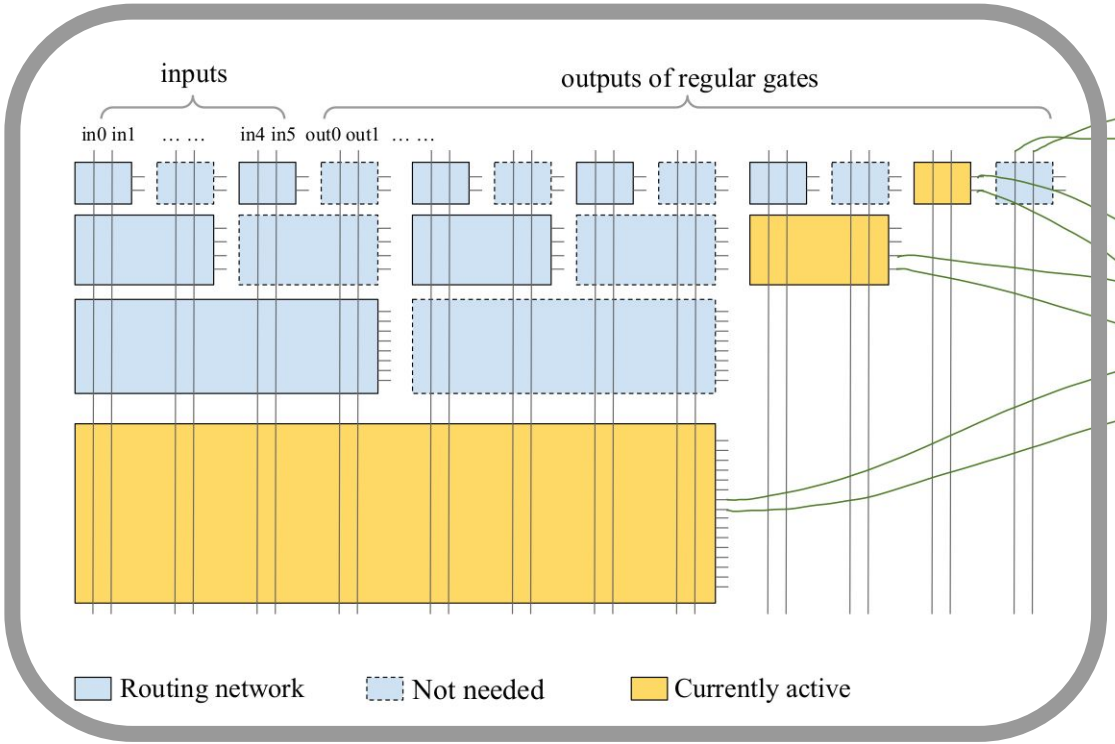
But with

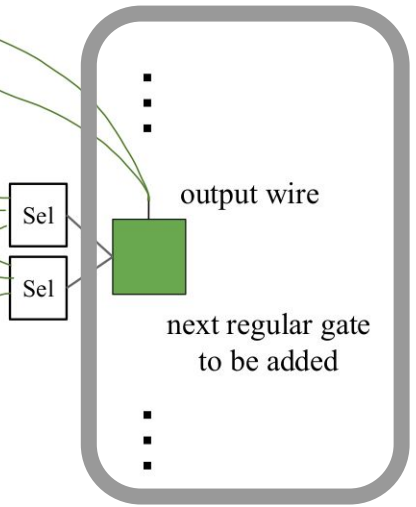


Challenge 1

padding

1 Routing gates





2 Regular gates

Challenge 2

iO for log-equiv circ

Gate by gate obfuscation

Mix-and-match attack

e.g., use values from a different eval

Challenge 2 iO for log-equiv circ

each wire carries a BARG provenance proof, **O(n)** time per wire

Ours

achieve the same but w/o BARG, **~O(1)** time per wire

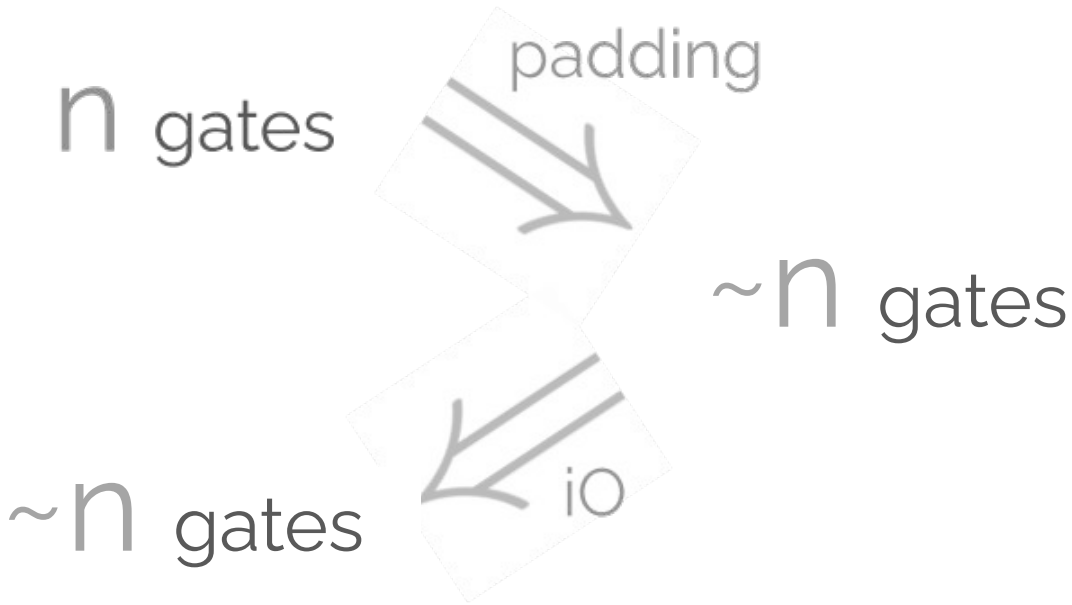
Applications of our quasilinear iO

- **Multi-input functional encryption** with quasilinear efficiency
- **iO for TM** with quasilinear efficiency

assume: short proofs of equivalence

Challenge 1

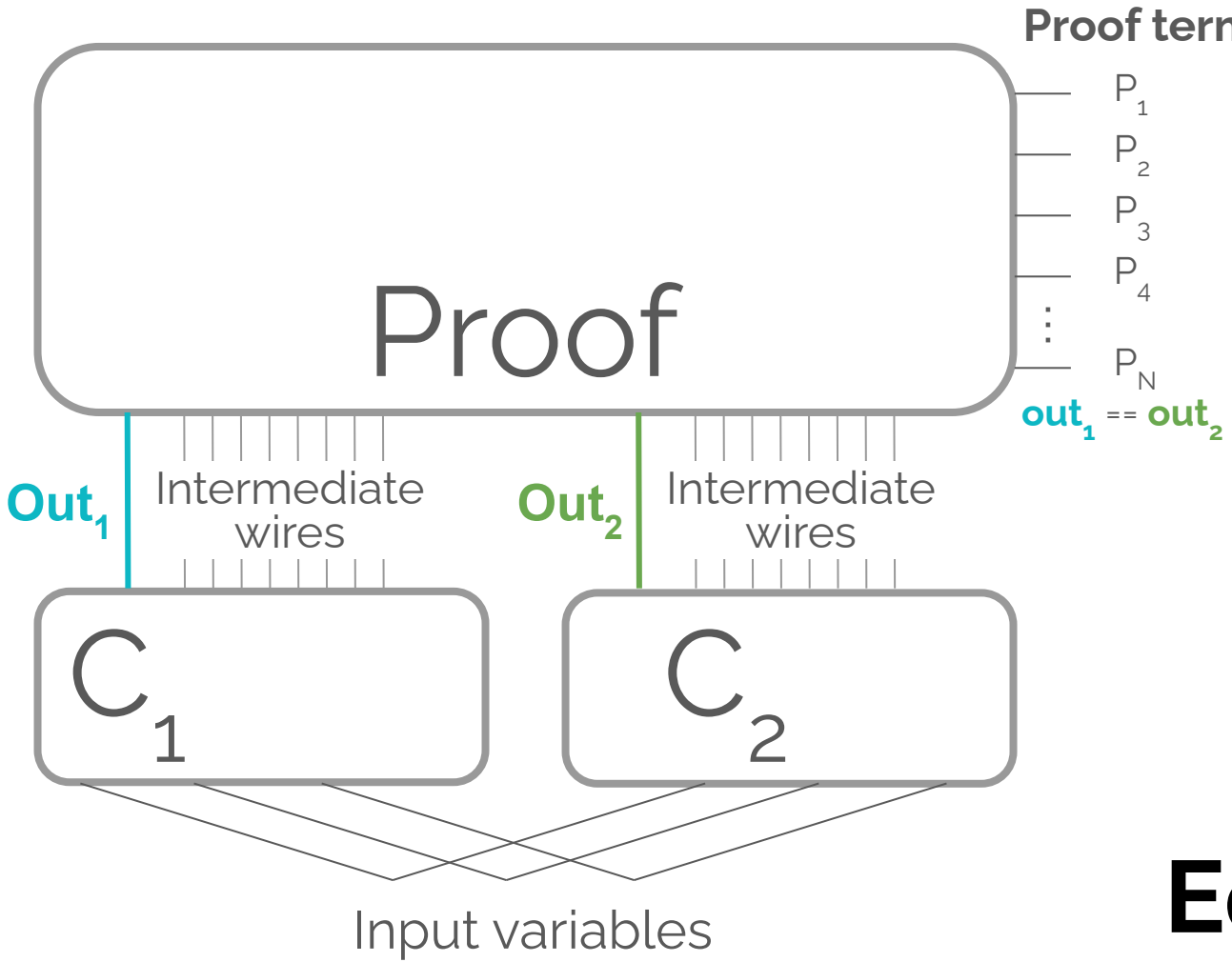
padding



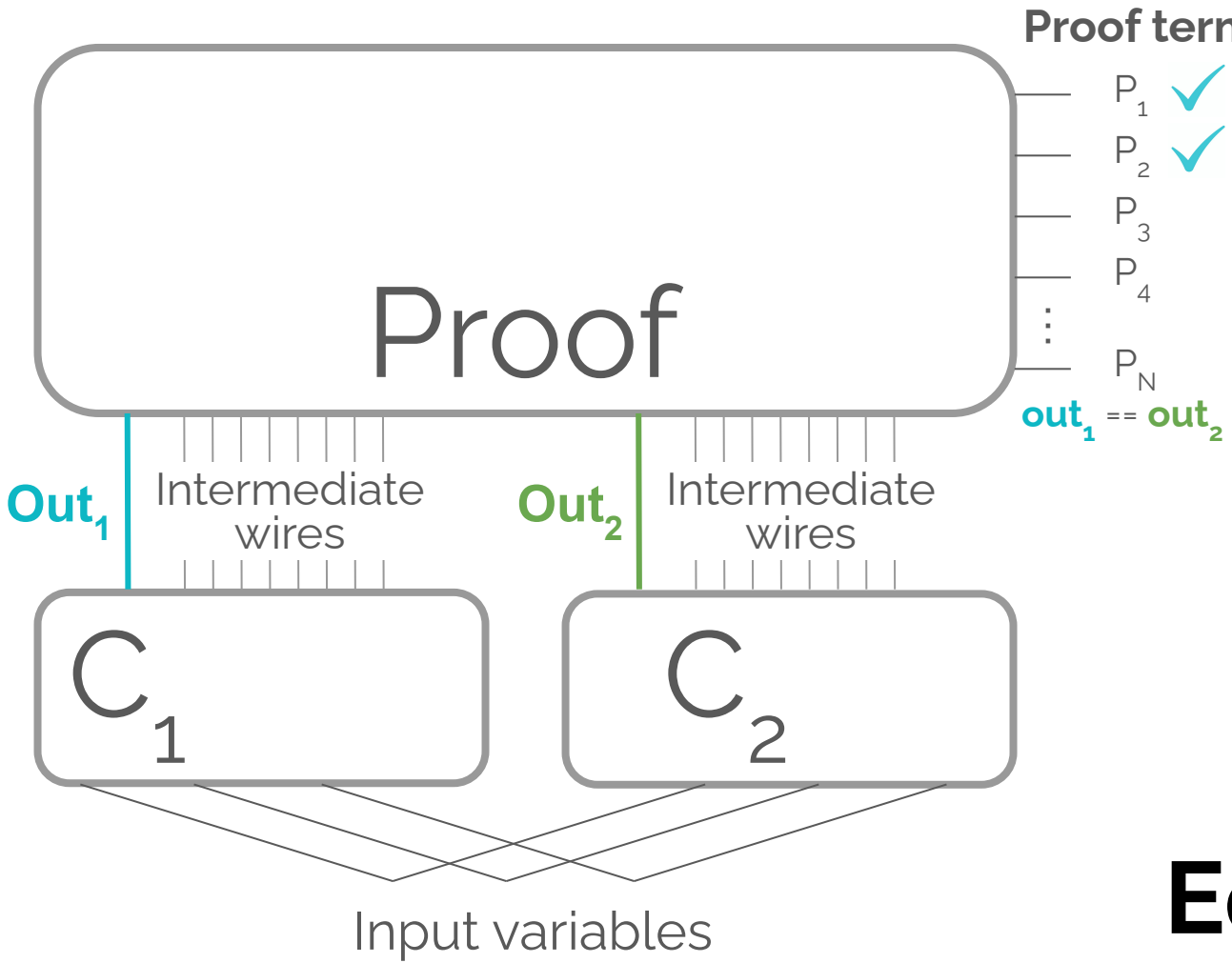
iO for log-equiv circ

assume: proof size ~ circuit size

Challenge 2



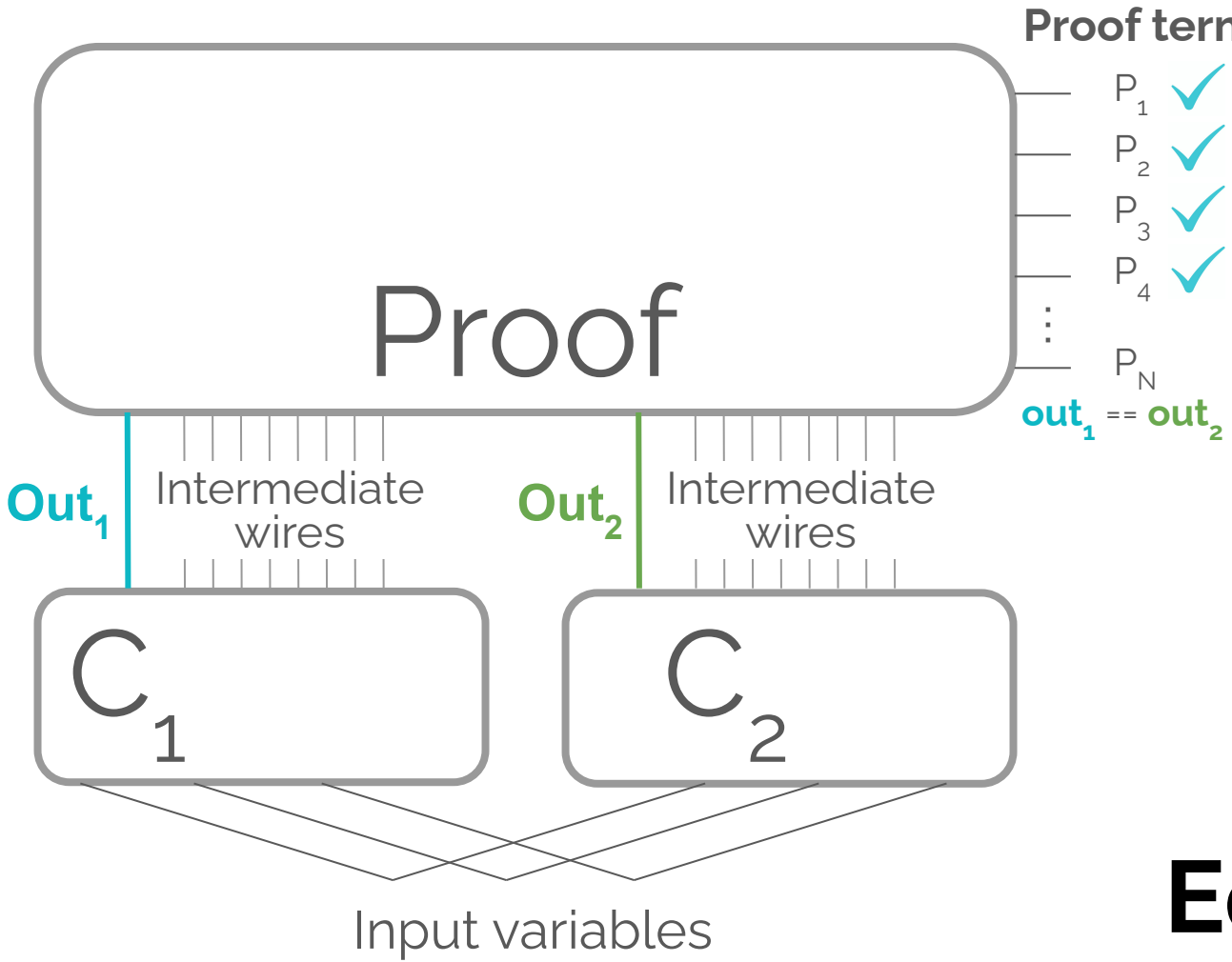
Proof of Equivalence



$\vdash P_1, P_2$

...

Proof of Equivalence

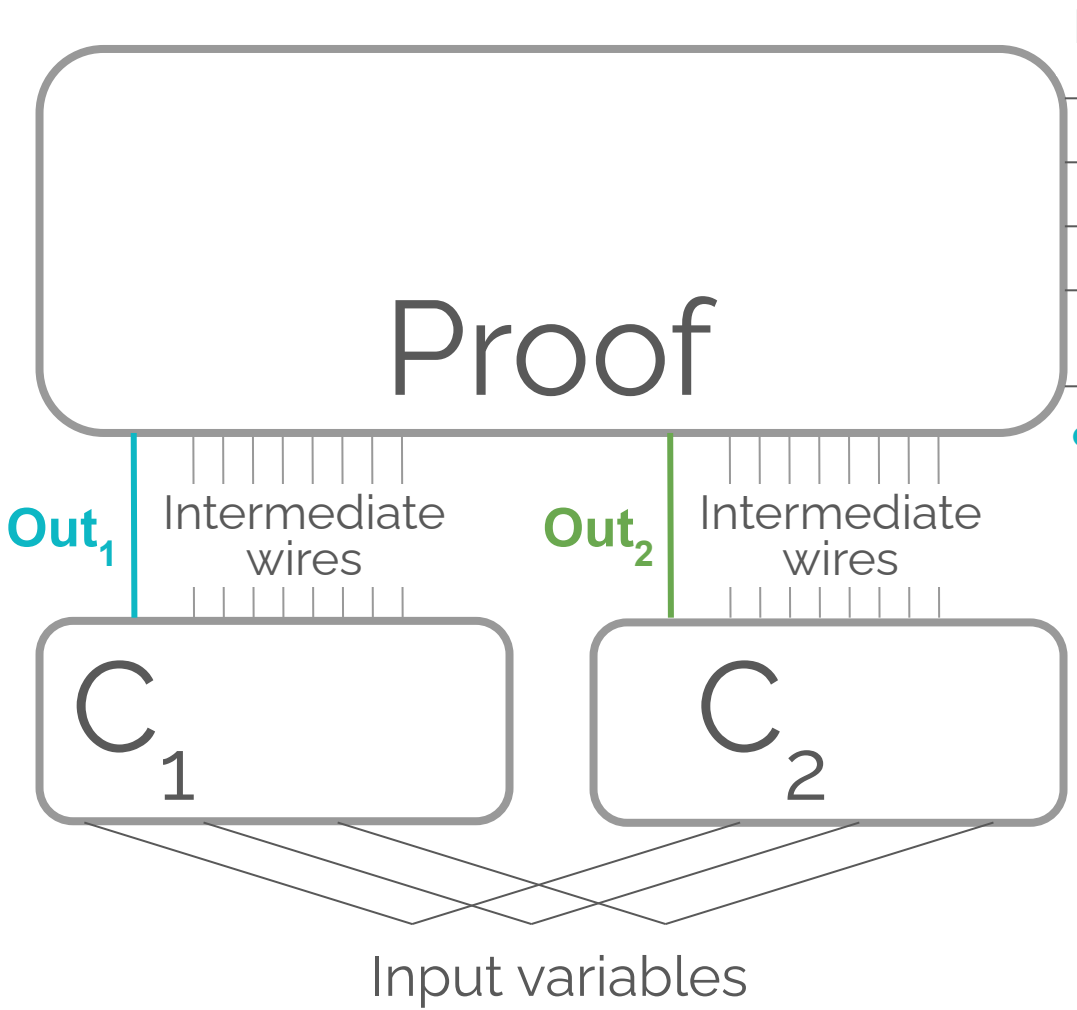


$\vdash P_1, P_2$

$P_1, P_2 \vdash P_3, P_4$

...

Proof of Equivalence



$\vdash P_1, P_2$

$P_1, P_2 \vdash P_3, P_4$

$P_3, P_4 \vdash P_5$

...

$P_{10}, P_{N-1} \vdash \mathbf{P_N}$

Proof of Equivalence

Axioms

$\vdash P \rightarrow (Q \rightarrow P)$

$\vdash \neg\neg P \rightarrow P$

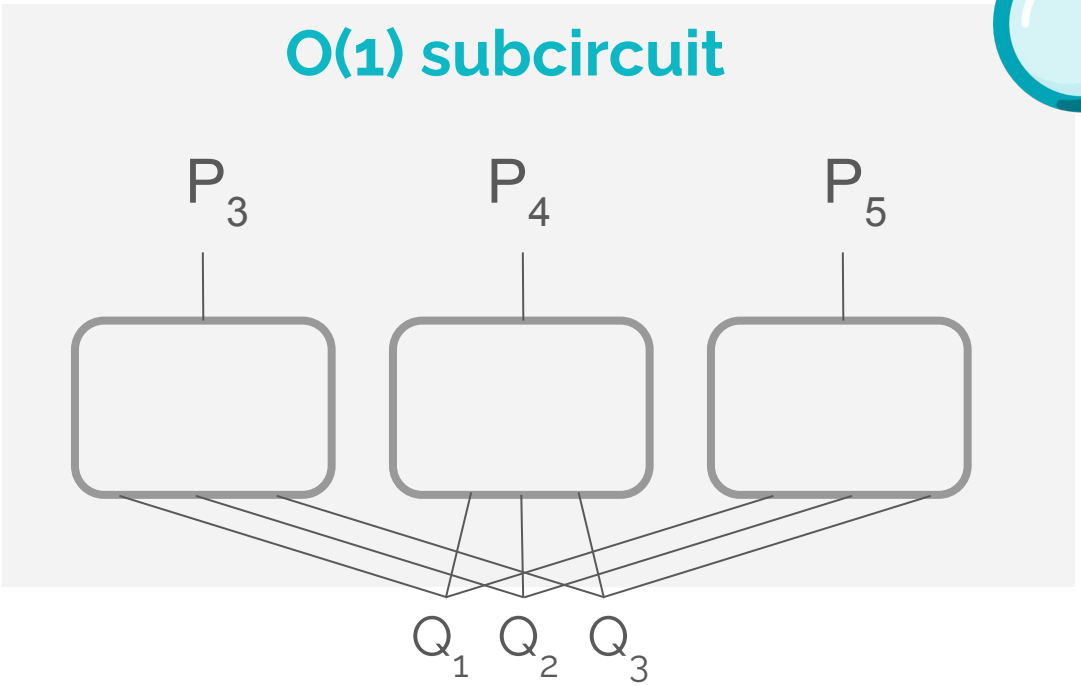
...

Modus Ponens

$P, P \rightarrow Q \vdash Q$

$P_3, P_4 \vdash P_5$

Example of a Proof Line



$P_3, P_4 \vdash P_5$

Each proof line: O(1)-sized subcircuit

Example: ∄ Short Equivalence Proof

$R \notin img(PRG)$

```
// do something

If PRG(inp) = R then
    // do special

Else
    // proceed normally
```

=

```
// do something

~~If PRG(inp) = R then~~
    ~~// do special~~

~~Else~~
    // proceed normally
```

[SW13]

Example: ∃ Short Equivalence Proof

C: encryption of 1

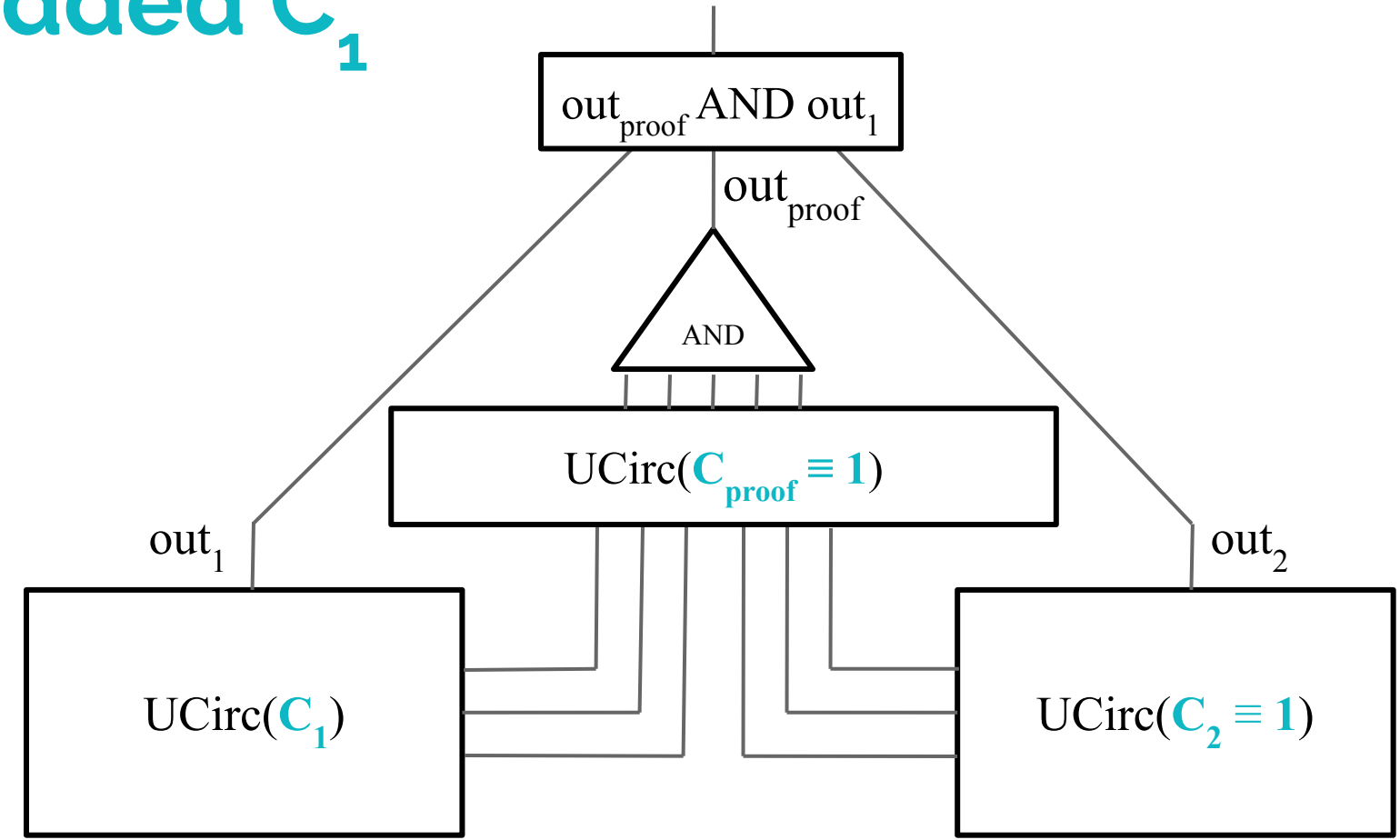
```
// do something

If Enc(o, r) = C then
    // do special

Else
    // proceed normally
```

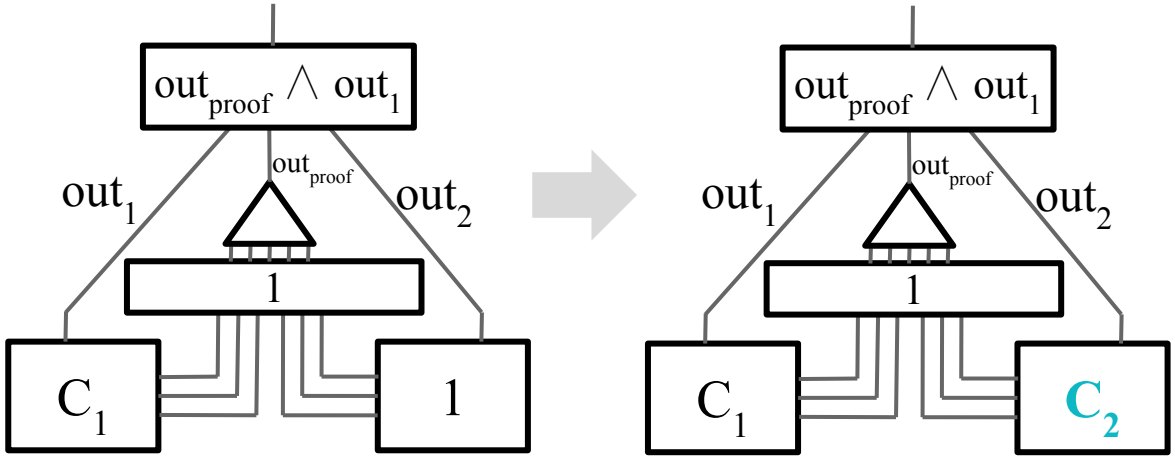
=

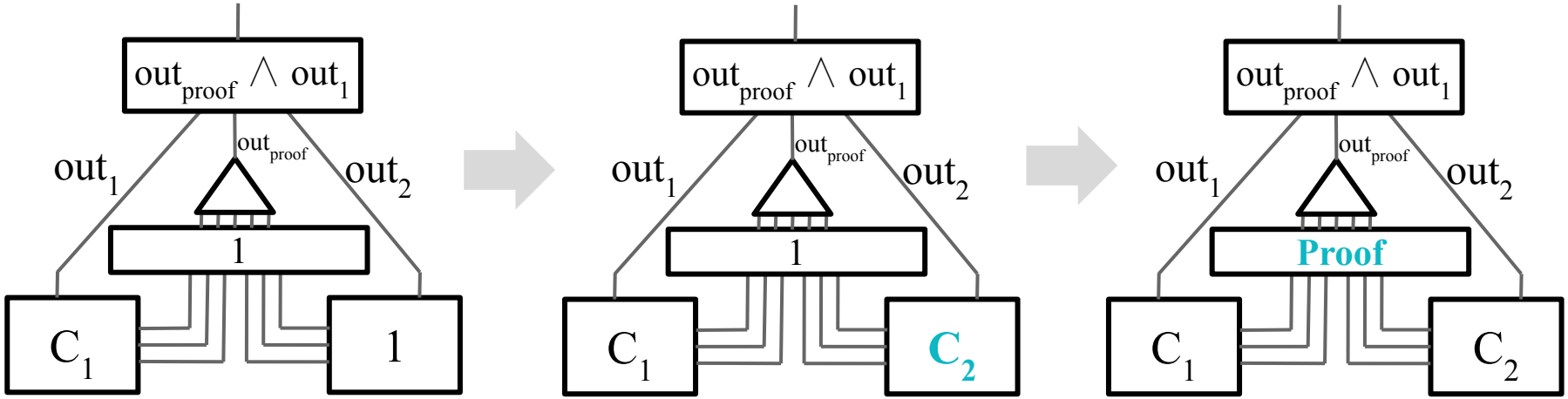
```
// do something

~~If Enc(o, r) = C then~~
    ~~// do special~~

~~Else~~
    // proceed normally
```

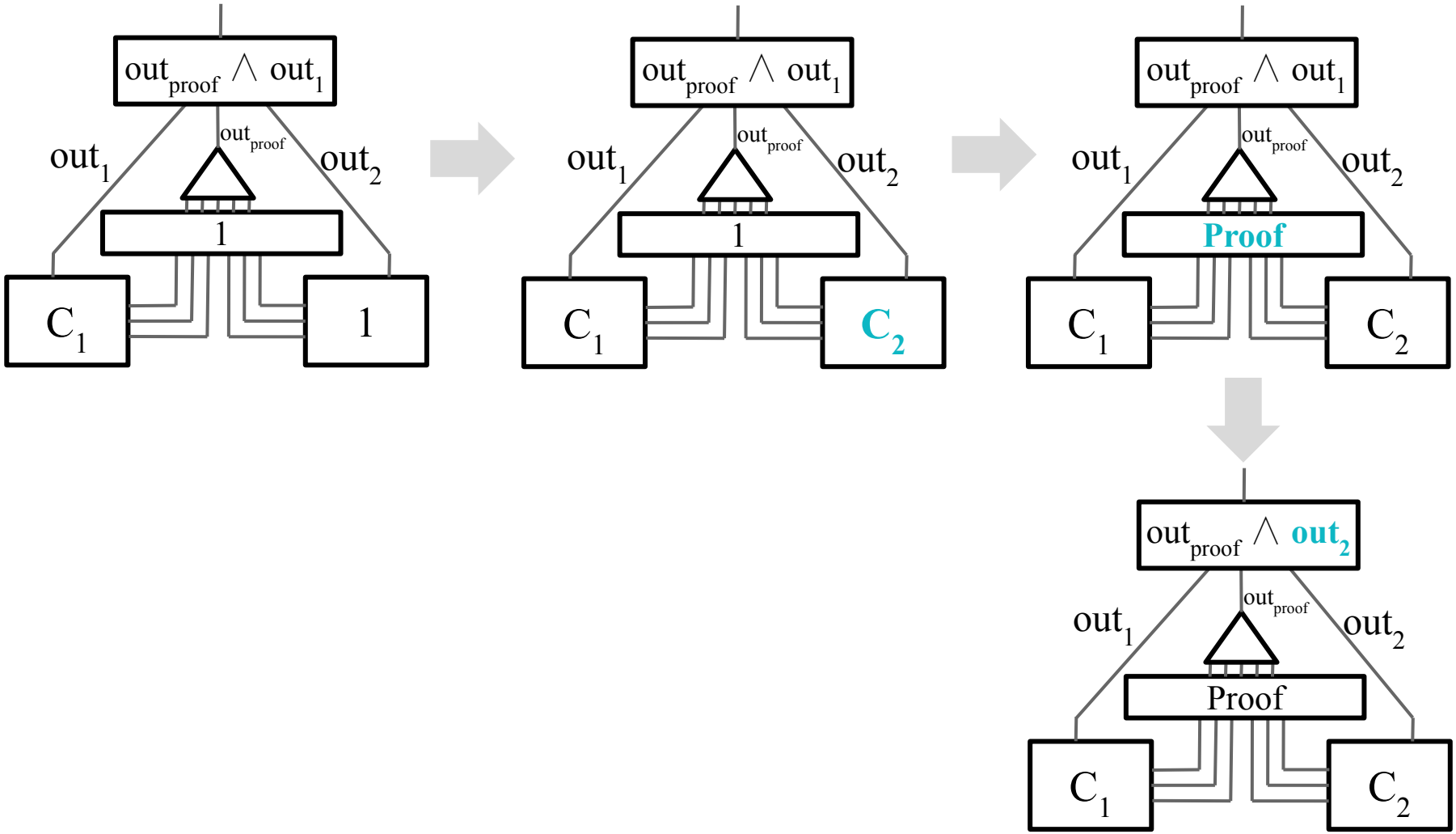
PKE w/ short proof of correct decryption

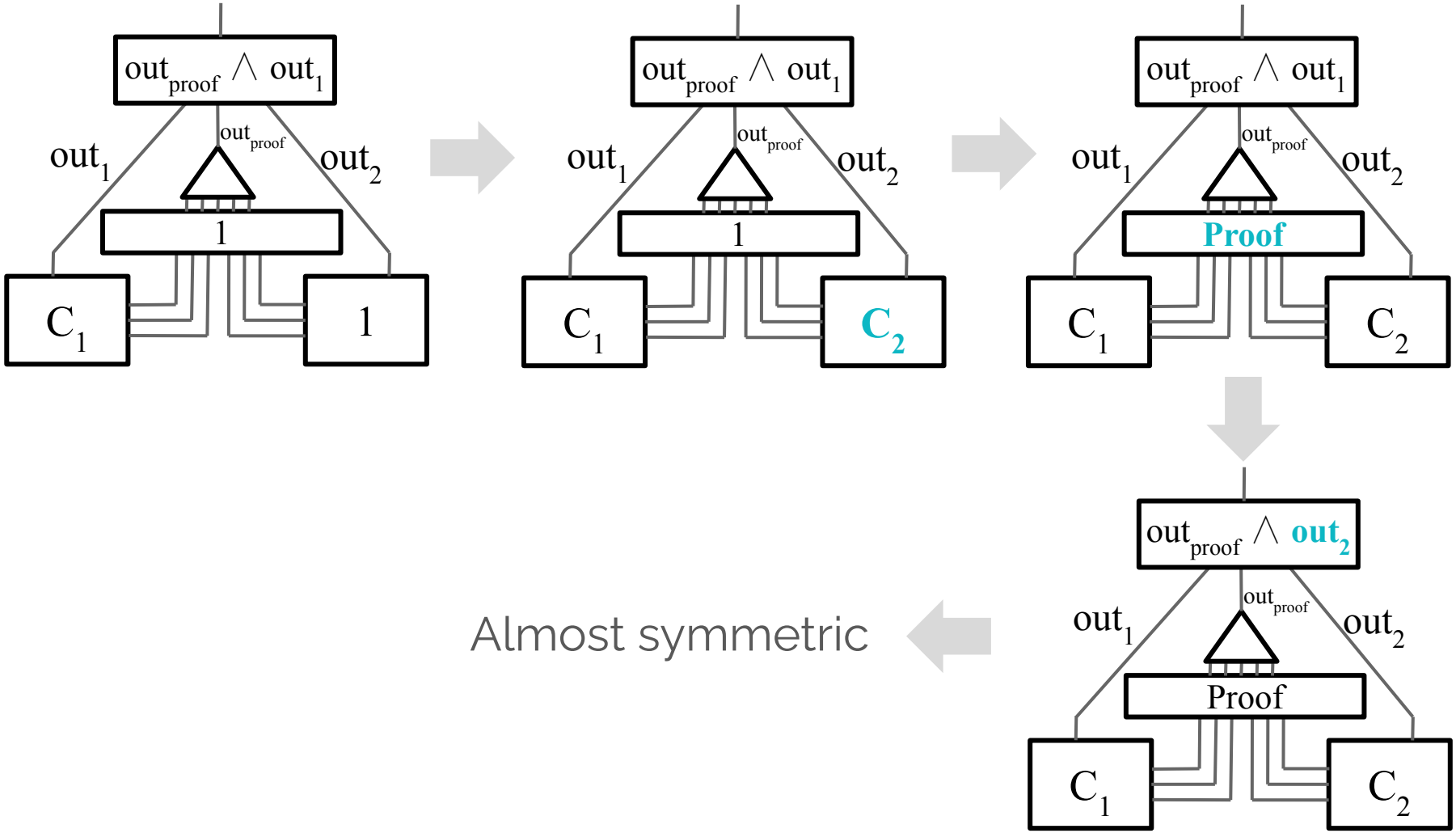
Padded C_1

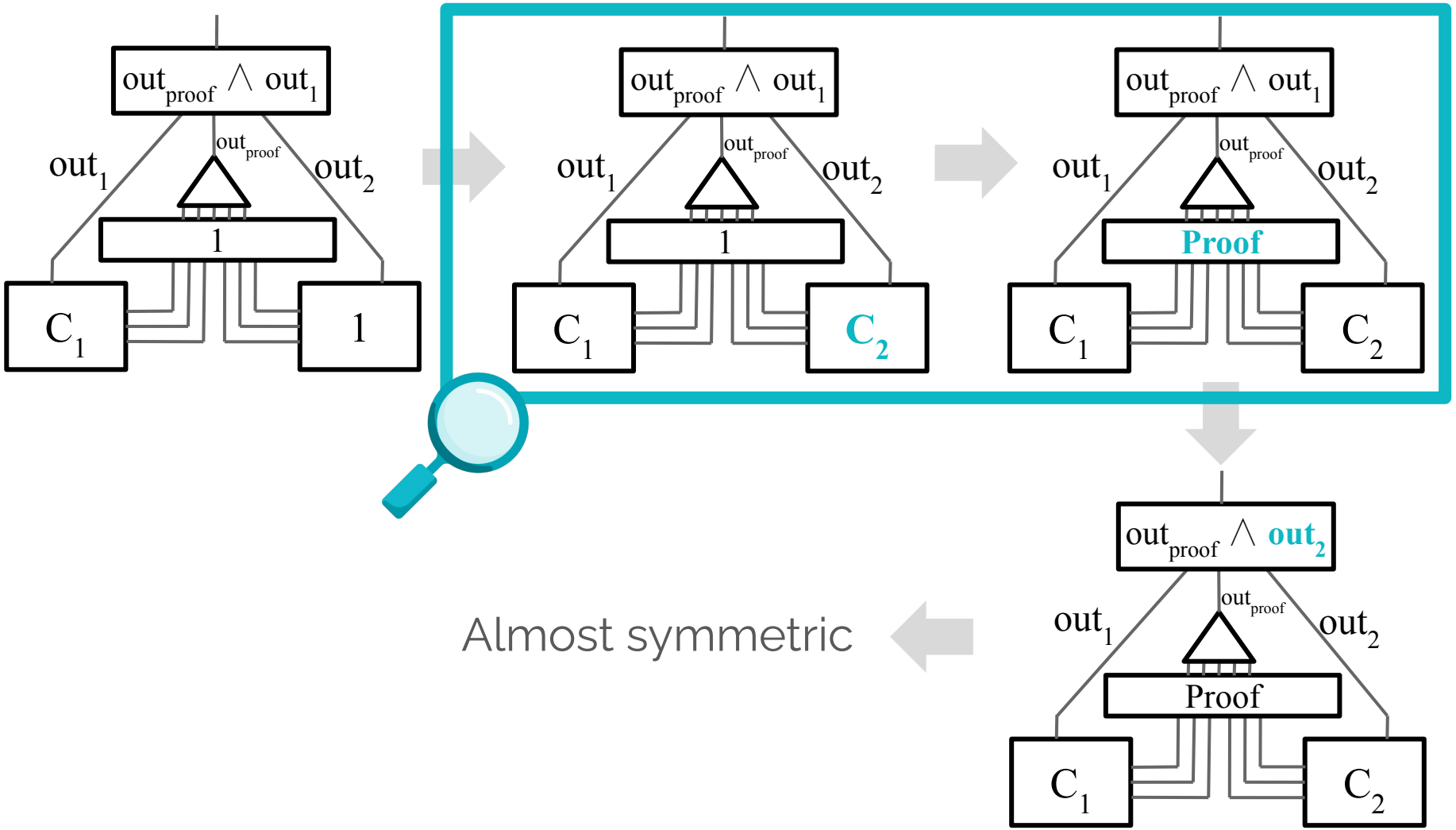

out_proof AND out_1
out_proof
AND
UCirc(C_proof ≡ 1)
out_1
out_2
UCirc(C_1)
UCirc(C_2 ≡ 1)

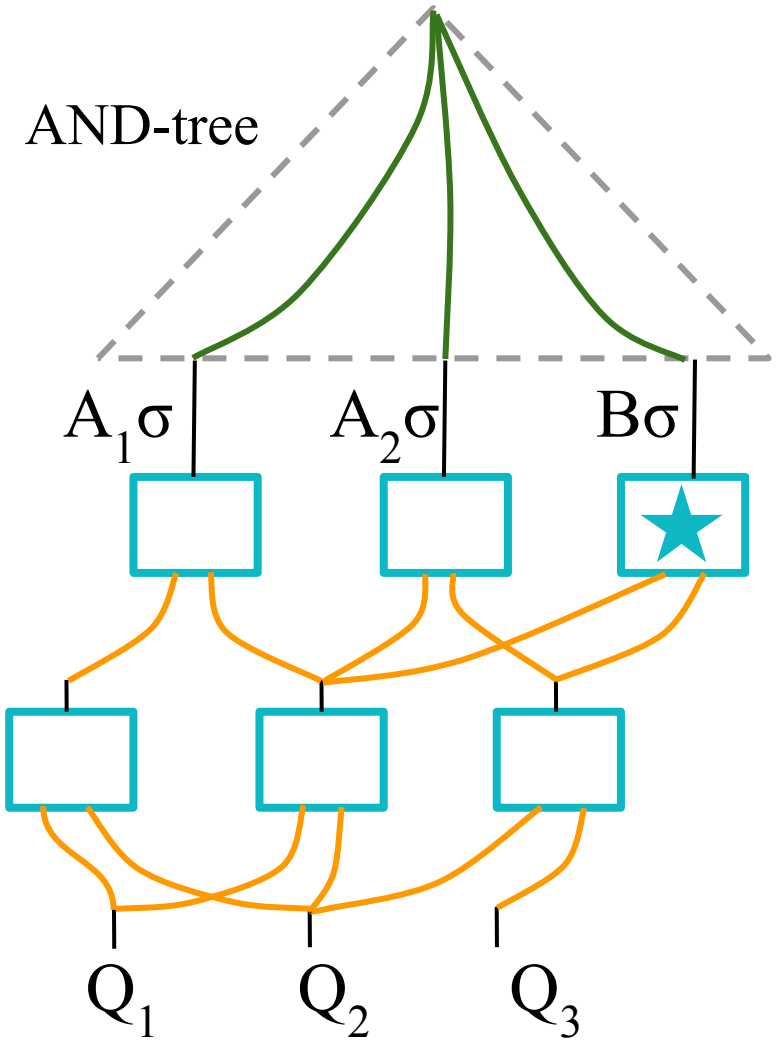

out_proof ∧ out_1
out_1
out_proof
out_2
1
C_1
1
out_proof ∧ out_1
out_1
out_proof
out_2
1
C_1
C_2


out_proof ∧ out_1
out_1
out_proof
out_2
1
C_1
1
out_proof ∧ out_1
out_1
out_proof
out_2
1
C_1
C_2
out_proof ∧ out_1
out_1
out_proof
out_2
Proof
C_1
C_2


out_proof ∧ out_1
out_1
out_proof
out_2
1
C_1
1
out_proof ∧ out_1
out_1
out_proof
out_2
1
C_1
C_2
out_proof ∧ out_1
out_1
out_proof
out_2
Proof
C_1
C_2
out_proof ∧ out_2
out_1
out_proof
out_2
Proof
C_1
C_2


out_proof ∧ out_1
out_1
out_proof
out_2
1
C_1
1
out_proof ∧ out_1
out_1
out_proof
out_2
1
C_1
C_2
out_proof ∧ out_1
out_1
out_proof
out_2
Proof
C_1
C_2
out_proof ∧ out_2
out_1
out_proof
out_2
Proof
C_1
C_2
Almost symmetric


out_proof ∧ out_1
out_1
out_proof
out_2
1
C_1
1
C_2
Proof
out_proof ∧ out_2
Almost symmetric



Differing subcircuit: all gates adjacent to and

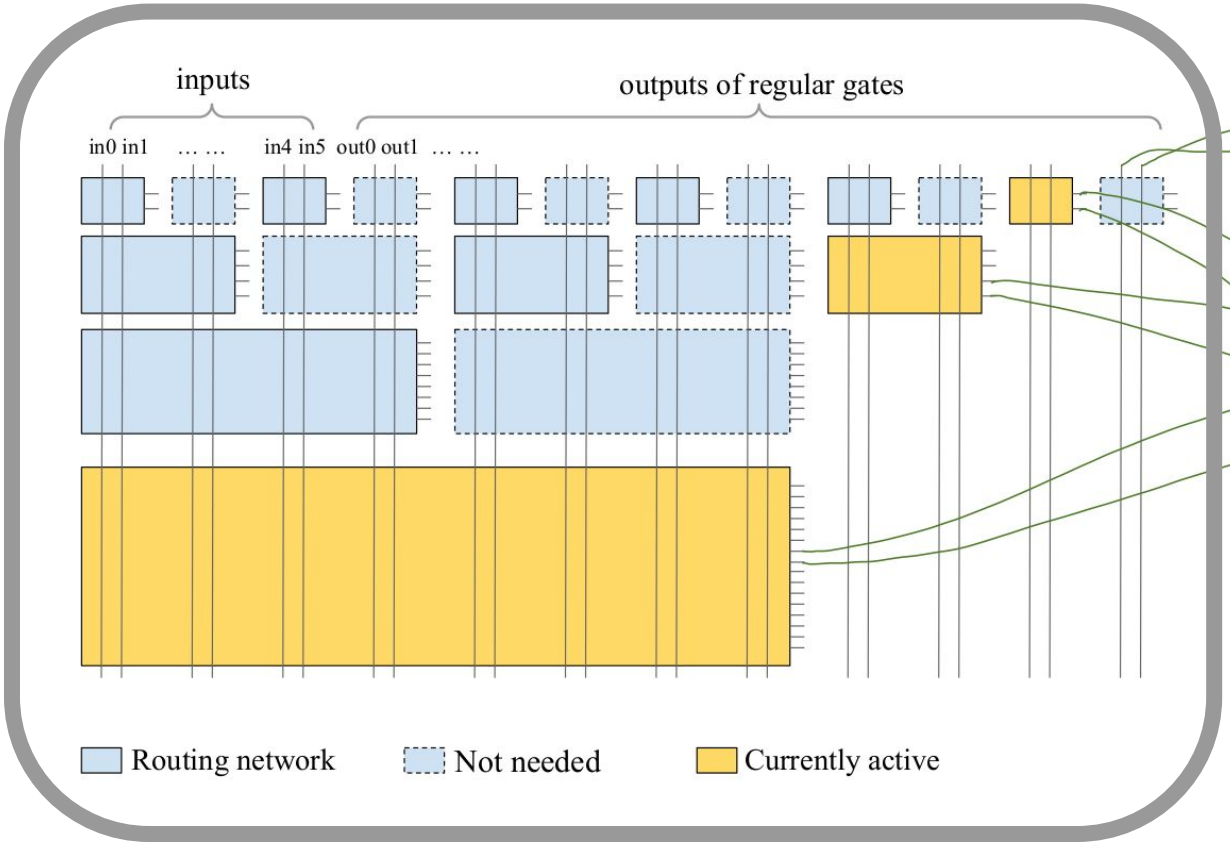
Regular gate

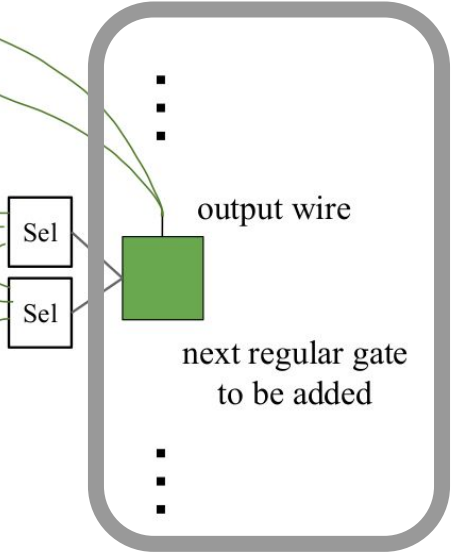
Regular gate being reconfigured

Path through routing network

Path from leaf to root

1 Routing gates

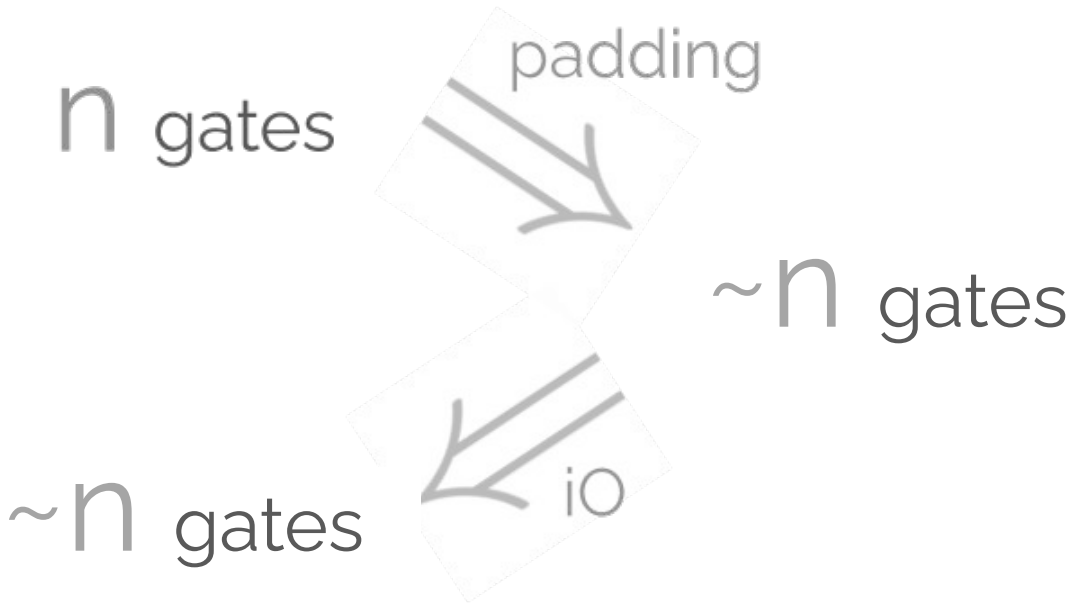




2 Regular gates

Challenge 1

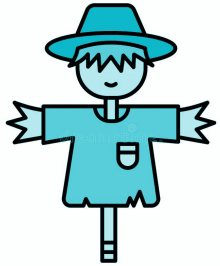
padding



assume: proof size ~ circuit size

iO for log-equiv circ

Challenge 2



ct_1, σ_1 | | ct_2, σ_2

Assert σ_1, σ_2 are valid MACs for ct_1, ct_2 using K^1_{mac}, K^2_{mac}

$m_1, m_2 = Dec(ct_1, ct_2)$ using K^1_{enc}, K^2_{enc}

Compute $out = g(m_1, m_2)$

Output $ct' = Enc(out)$, $\sigma' = MAC(ct')$ using $K^{out}_{enc}, K^{out}_{MAC}$

can mix ct, σ from 2 evaluations

ct_1, σ_1 | | ct_2, σ_2

Assert σ_1, σ_2 are valid MACs for ct_1, ct_2 using K^1_{mac}, K^2_{mac}

$m_1, m_2 = Dec(ct_1, ct_2)$ using K^1_{enc}, K^2_{enc}

Compute $out = g(m_1, m_2)$

Output $ct' = Enc(out)$, $\sigma' = MAC(ct')$ using $K^{out}_{enc}, K^{out}_{MAC}$

ct_1, h_1, σ_1 ct_2, h_2, σ_2 h',

π

Assert π proves that (ct_1, ct_2, h_1, h_2) consistent w/ h'

Assert σ_1, σ_2 valid for $(ct_1, h_1), (ct_2, h_2)$ using K^1_{mac}, K^2_{mac}

$m_1, m_2 = Dec(ct_1, ct_2)$ using K^1_{enc}, K^2_{enc}

Compute $out = g(m_1, m_2)$

Output $ct' = Enc(out)$, $\sigma' = MAC(ct', h')$ using $K^{out}_{enc}, K^{out}_{MAC}$

ct_1, h_1, σ_1 ct_2, h_2, σ_2 h',

π

Assert π proves that (ct_1, ct_2, h_1, h_2) consistent w/ h'

Ass $h_1)$, $(ct_2$

m_1, g $K^1_{enc'}$

Compute out = $g(m_1, m_2)$

Output ct' = Enc(out), σ' = MAC(ct', h') using K^{out}_{enc}, K^{out}_{MAC}

SSB hashes

BARG proof

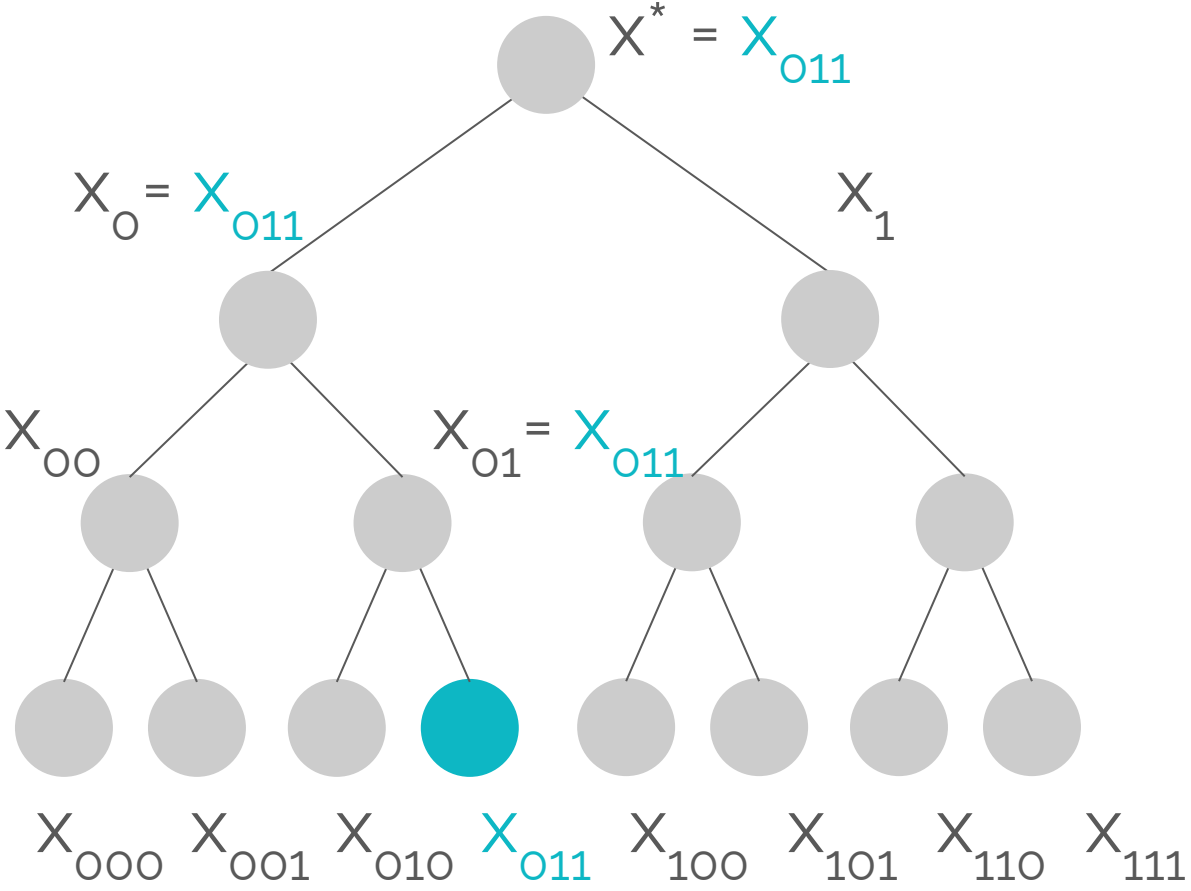
Avoid BARGs

through

non-blackbox

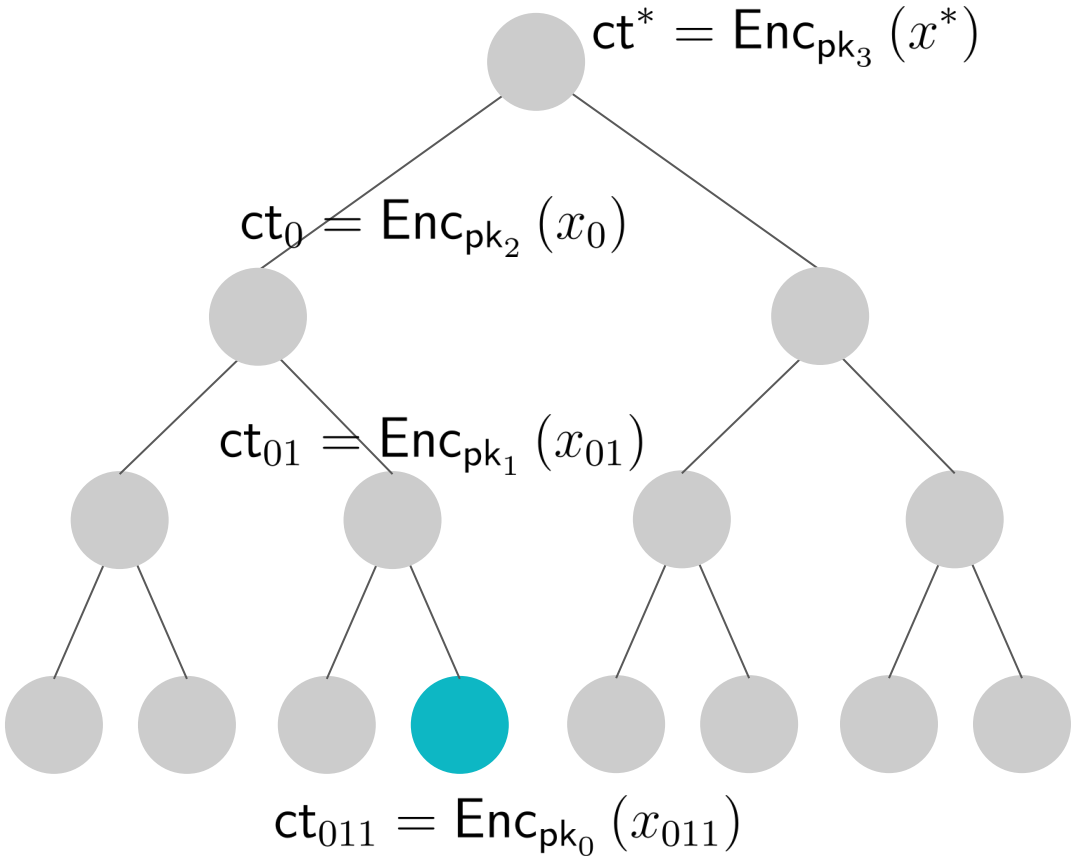
use of **SSB hash**

[HW15]



SSB hash

[HW15]



Public key

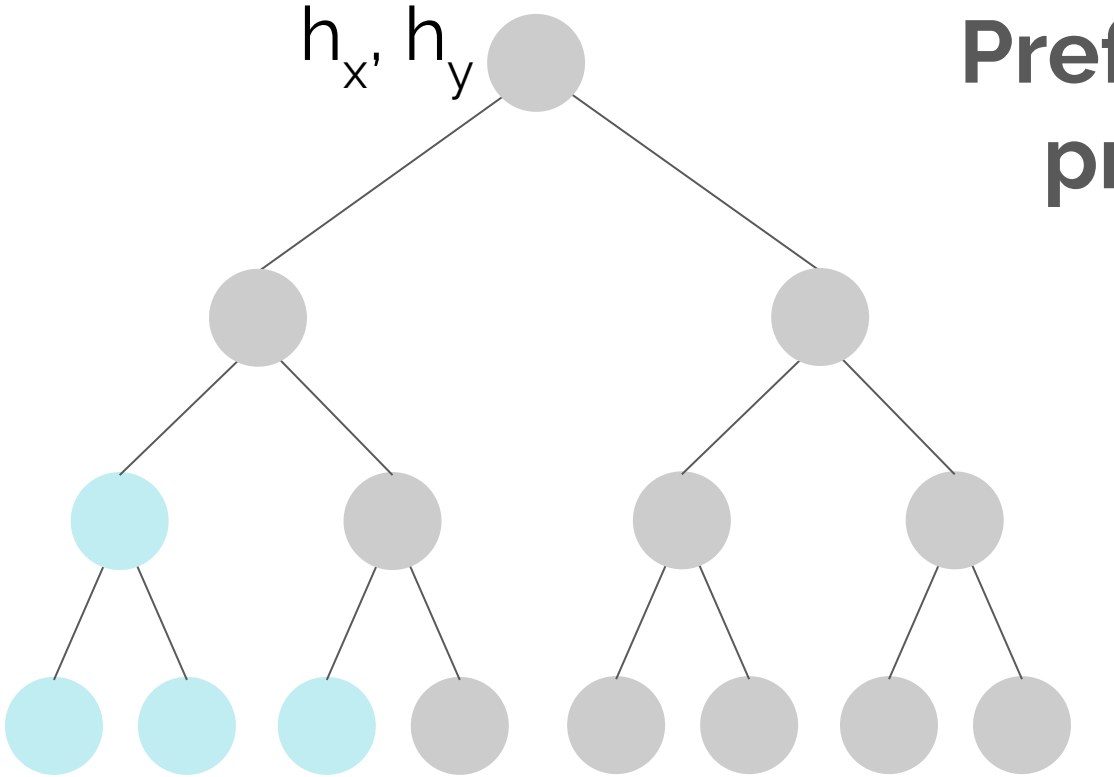
$\mathsf{pk}_3,\ \mathsf{Enc}_{\mathsf{pk}_3}(\mathsf{sk}_2, \mathsf{idx}_3)$

$\mathsf{pk}_2,\ \mathsf{Enc}_{\mathsf{pk}_2}(\mathsf{sk}_1, \mathsf{idx}_2)$

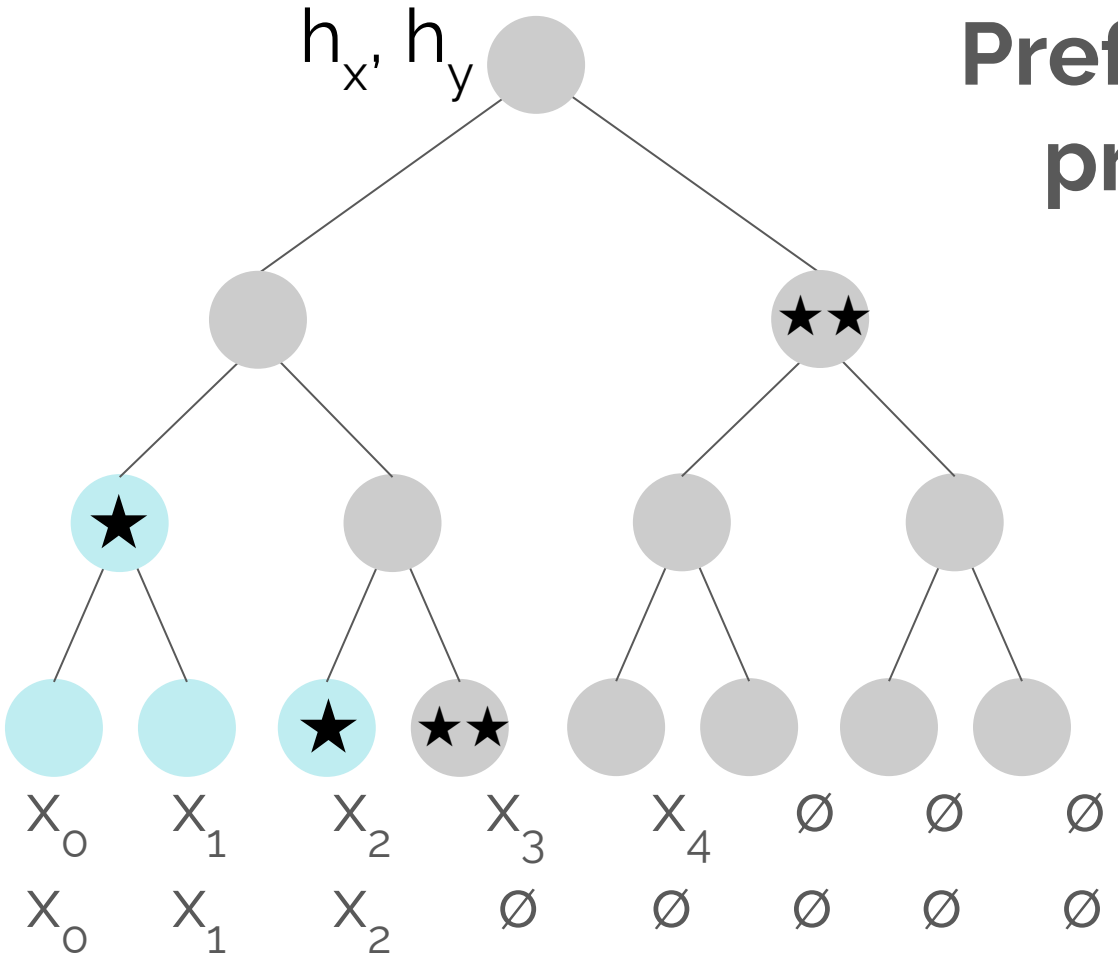
$\mathsf{pk}_1,\ \mathsf{Enc}_{\mathsf{pk}_1}(\mathsf{sk}_0, \mathsf{idx}_1)$

pk_0

Prefix consistency proof for h_x, h_y

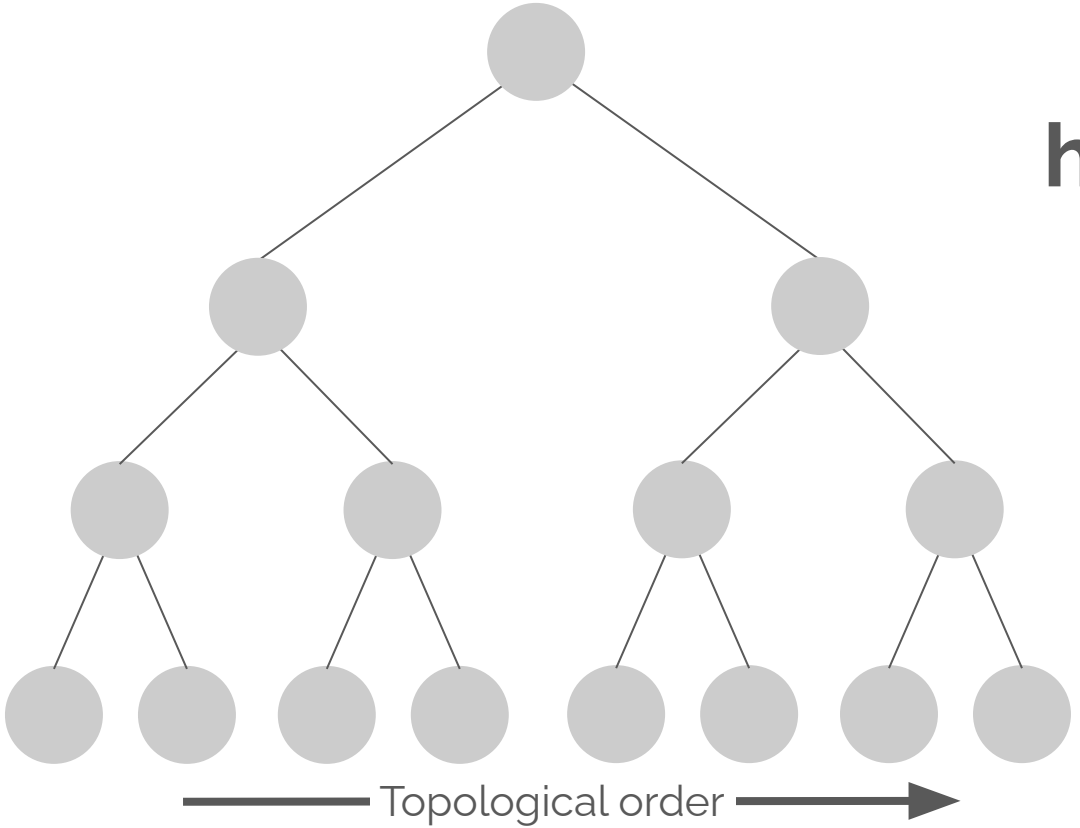


Prefix consistency proof for h_x, h_y



★: 1 hash

★★: 2 hashes



Compute all N hashes & proofs incrementally in ~N time

every wire: hash dependent wires
every gate: prove input hash consistent with output hash

Quasilinear $iO^{\mathcal{EF}}$ for TM

Use RAM obfuscator to obfuscate [JLL23]

$UObf^{M}(L)$

On inp length L, output **$iO^{\mathcal{EF}}(OTM(M))$**

Quasilinear $iO^{\mathcal{EF}}$ for TM

Use RAM obfuscator to obfuscate [JLL23]

$UObf^{M}(L)$

On inp length L, output **$iO^{\mathcal{EF}}(OTM(M))$**

$\widetilde{O}_\lambda(1)$ size and obf time

$\widetilde{O}_\lambda(T + N_{\text{proof}})$ eval time

Assume: $|M| \leq \widetilde{O}_\lambda(1)$

Also in our paper

eprint/2025/307

- Detailed construction and proofs
- Applications to MIFE
- Applications to iO for TM

Open questions

- Prove/disprove the input-len barrier
- Concretely efficient iO?

Thank you !

elainershi@gmail.com