Lookup Arguments

The 13th BIU Winter School on Cryptography

Mary Maller, Ethereum Foundation and PQShield, February 2023, Tel-Aviv

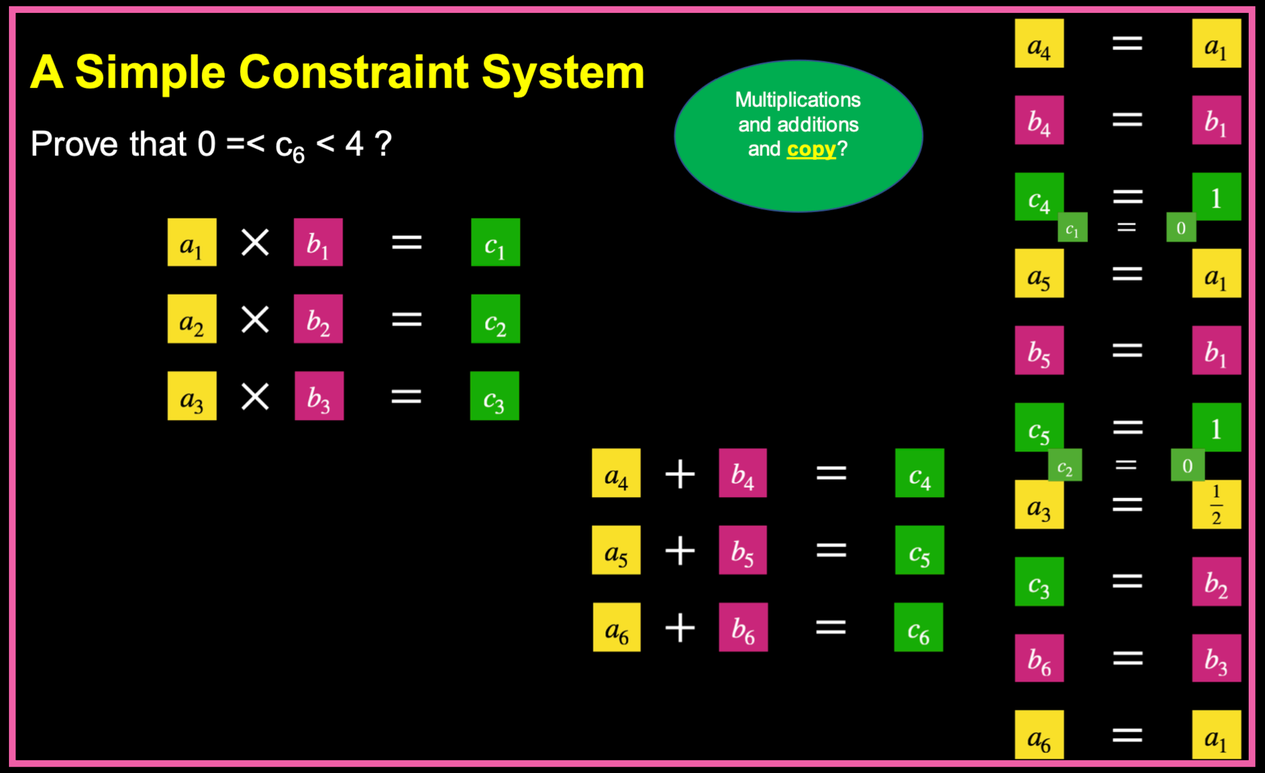
Previously

- Described Plonkish Arithmetisation System in detail:
  - Addition and multiplication constraints
  - Copy constraints
  - Selector polynomials
  - Custom Constraints
- Now we will discuss lookup constraints

Revisiting Simple Constraint System

Prove that $0 \leq x < 4$

Quite a lot of constraints



Revisiting Simple Constraint System

Prove that $0 \leq x \leq 4$

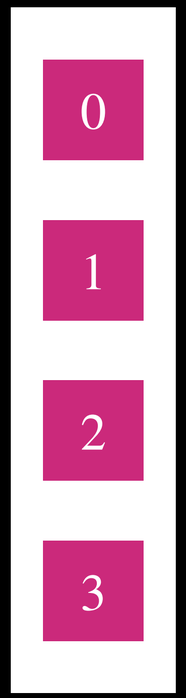
Alternatively

- Prove that x is inside table T

Witness value

x

Table T of allowed range



Revisiting Simple Constraint System

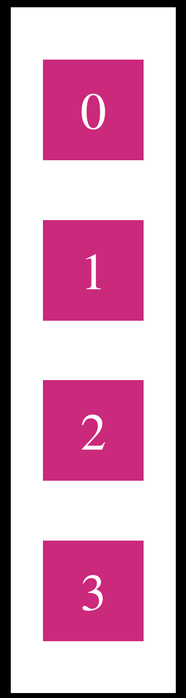
Prove that $0 \leq x \leq 4$

Alternatively

Witness value

x

Table T of allowed range



- Prove that x is inside table T
- Polynomial $t(X)$ such that:
  - $t(1) = 0$
  - $t(2) = 1$
  - $t(3) = 2$
  - $t(4) = 3$

Revisiting Simple Constraint System

Prove that $0 \leq x \leq 4$

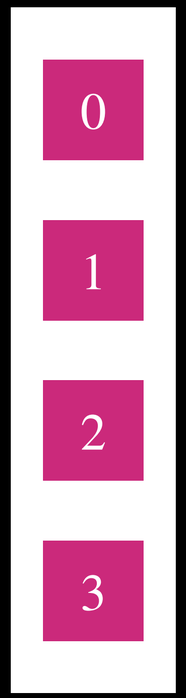
From 6 constraints + many copy constraints to 1 custom constraint

Alternatively

Witness value

x

Table T of allowed range



- Prove that x is inside table T
- Polynomial $t(X)$ such that:
  - $t(1) = 0$
  - $t(2) = 1$
  - $t(3) = 2$
  - $t(4) = 3$
- Prove that $x \in \{t(1), t(2), t(3), t(4)\}$

Revisiting Simple Constraint System

Almost free range constraints

Witness values

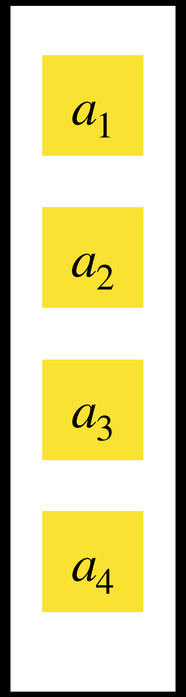
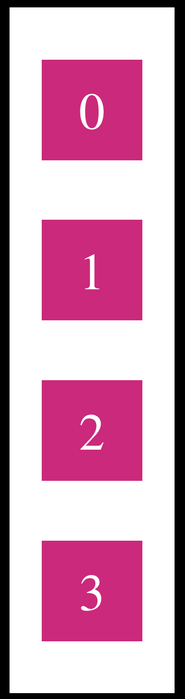


Table T of allowed range

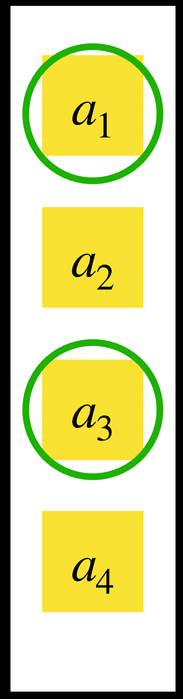


- Prove that a_1, a_2, a_3, a_4 is inside table T
- Polynomial $t(X)$ such that:
  - $t(1) = 0$
  - $t(2) = 1$
  - $t(3) = 2$
  - $t(4) = 3$
- Witness polynomial $a(X)$ such that:
  - $a(1) = a_1$
  - $a(2) = a_2$
  - $a(3) = a_3$
  - $a(4) = a_4$
- Prove that $a(i) \in \{t(1), t(2), t(3), t(4)\}$ for $i \in \{0,1,2,3\}$

Revisiting Simple Constraint System

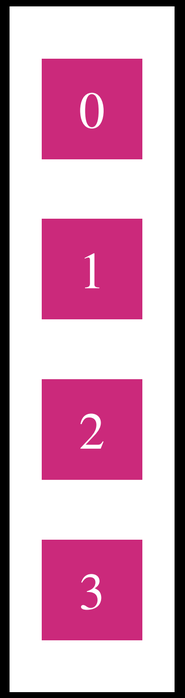
Selector Polynomials

Witness values



We will ignore selector polynomials in this talk.

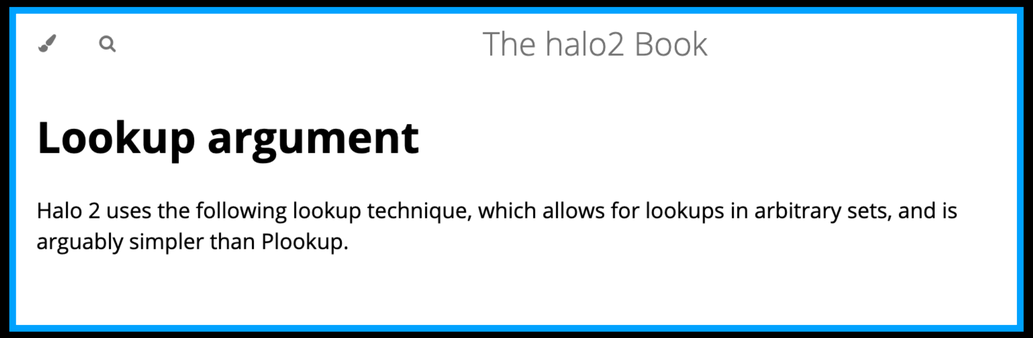
Table T of allowed range



- Prove that a_1, a_3 is inside table T
- Polynomial $t(X)$ such that:
  - $t(1) = 0$
  - $t(2) = 1$
  - $t(3) = 2$
  - $t(4) = 3$
- Witness polynomial $a(X)$ such that:
  - $a(1) = a_1$
  - $a(2) = a_2$
  - $a(3) = a_3$
  - $a(4) = a_4$
- Prove that $a(i) \in \{t(1), t(2), t(3), t(4)\}$ for $i \in \{0,2\}$

The Halo2 Strategy

Want to Prove a Lookup Argument.



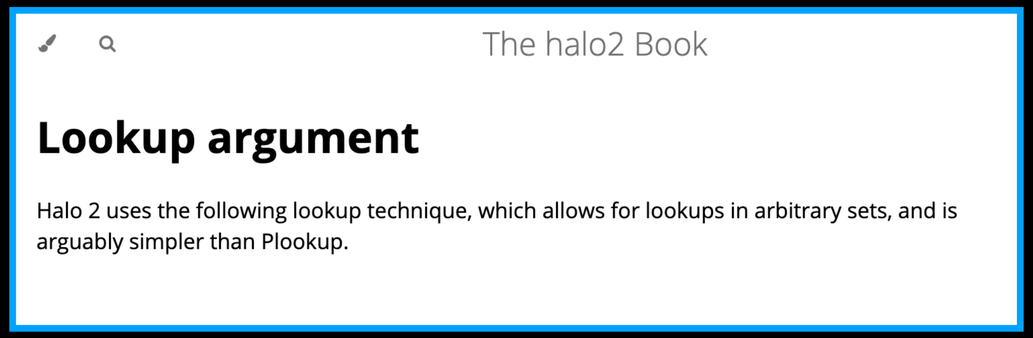
https://zcash.github.io/halo2/design/proving-system/lookup.html

The Halo2 Strategy

Want to Prove a Lookup Argument.

Prove that

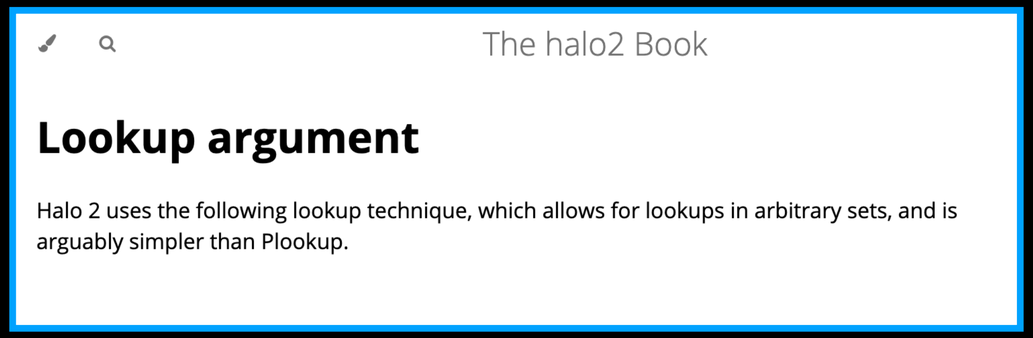
$a(i) \in \{t(1), t(2), t(3), t(4)\}$ for $i \in \{0,1,2,3\}$



https://zcash.github.io/halo2/design/proving-system/lookup.html

The Halo2 Strategy

Want to Prove a Lookup Argument.



https://zcash.github.io/halo2/design/proving-system/lookup.html

Prove that

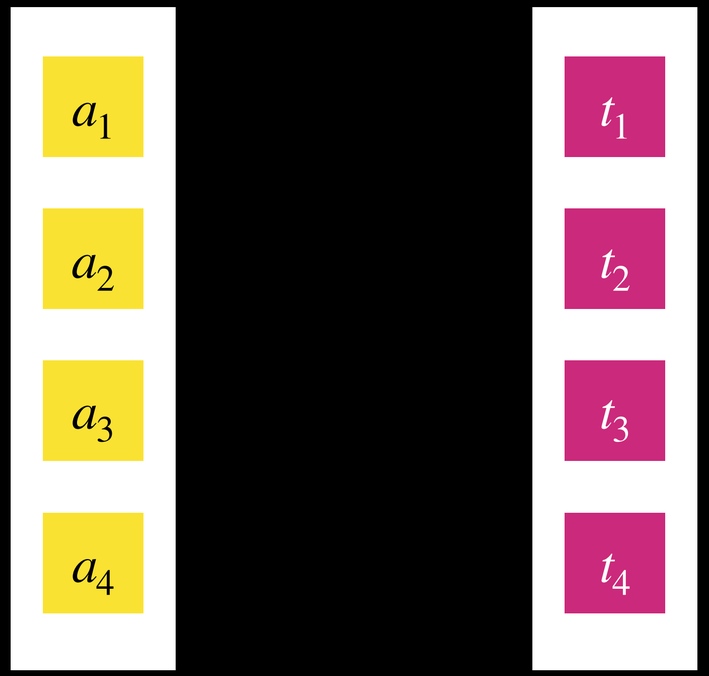
$a(i) \in \{t(1), t(2), t(3), t(4)\}$ for $i \in \{0,1,2,3\}$

- We will try and fail to use a copy argument to prove our lookup constraint.
- We will try and fail to use a permutation argument to prove our lookup constraint.
- We will then discuss how Halo2 actually does it.

The Halo2 Strategy

Can we use a copy argument?

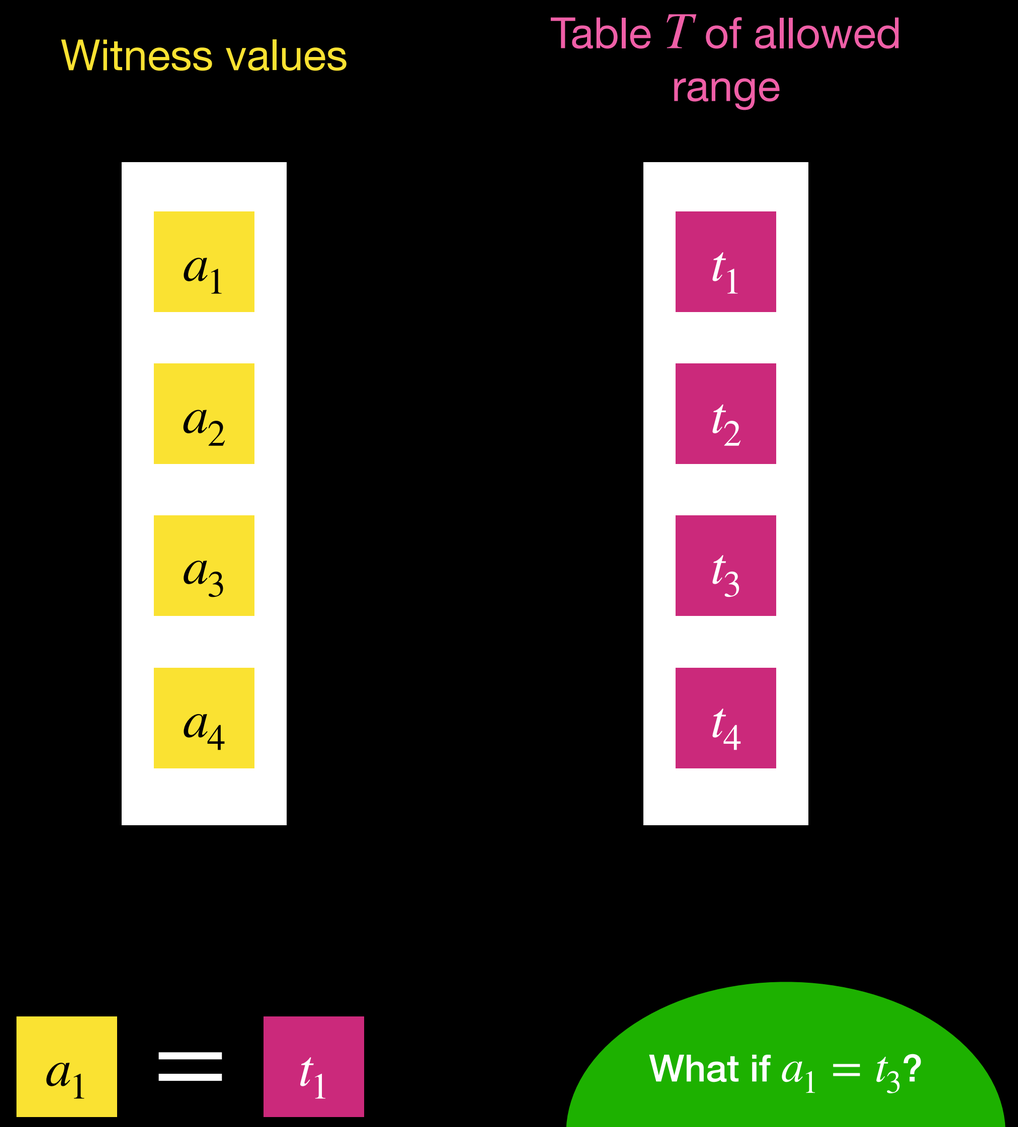
- Polynomial $t(X)$ such that:
  - $t(1) = 0$
  - $t(2) = 1$
  - $t(3) = 2$
  - $t(4) = 3$
- Witness polynomial $a(X)$ such that:
  - $a(1) = a_1$
  - $a(2) = a_2$
  - $a(3) = a_3$
  - $a(4) = a_4$
- Prove that $\{a_1, a_2, a_3, a_4\} \in \{t_1, t_2, t_3, t_4\}$



The Halo2 Strategy

Can we use a copy argument?

- Polynomial $t(X)$ such that:
  - $t(1) = 0$
  - $t(2) = 1$
  - $t(3) = 2$
  - $t(4) = 3$
- Witness polynomial $a(X)$ such that:
  - $a(1) = a_1$
  - $a(2) = a_2$
  - $a(3) = a_3$
  - $a(4) = a_4$
- Prove that $\{a_1, a_2, a_3, a_4\} \in \{t_1, t_2, t_3, t_4\}$



The Halo2 Strategy

Can we use a permutation argument?

- Polynomial $t(X)$ such that:
  - $t(1) = 0$
  - $t(2) = 1$
  - $t(3) = 2$
  - $t(4) = 3$
- Witness polynomial $a(X)$ such that:
  - $a(1) = a_1$
  - $a(2) = a_2$
  - $a(3) = a_3$
  - $a(4) = a_4$
- Prove that $\{a_1, a_2, a_3, a_4\} \in \{t_1, t_2, t_3, t_4\}$

Witness values

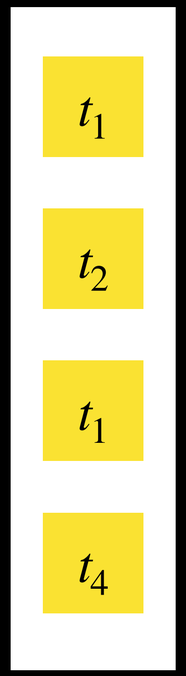
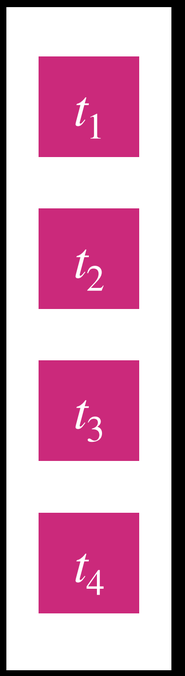


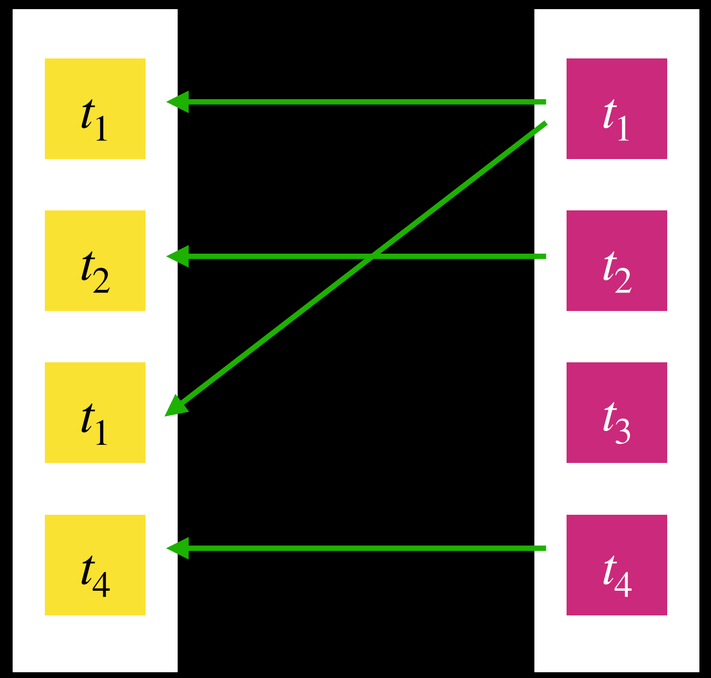
Table T of allowed range



The Halo2 Strategy

Can we use a permutation argument?

- Polynomial $t(X)$ such that:
  - $t(1) = 0$
  - $t(2) = 1$
  - $t(3) = 2$
  - $t(4) = 3$
- Witness polynomial $a(X)$ such that:
  - $a(1) = a_1$
  - $a(2) = a_2$
  - $a(3) = a_3$
  - $a(4) = a_4$
- Prove that $\{a_1, a_2, a_3, a_4\} \in \{t_1, t_2, t_3, t_4\}$



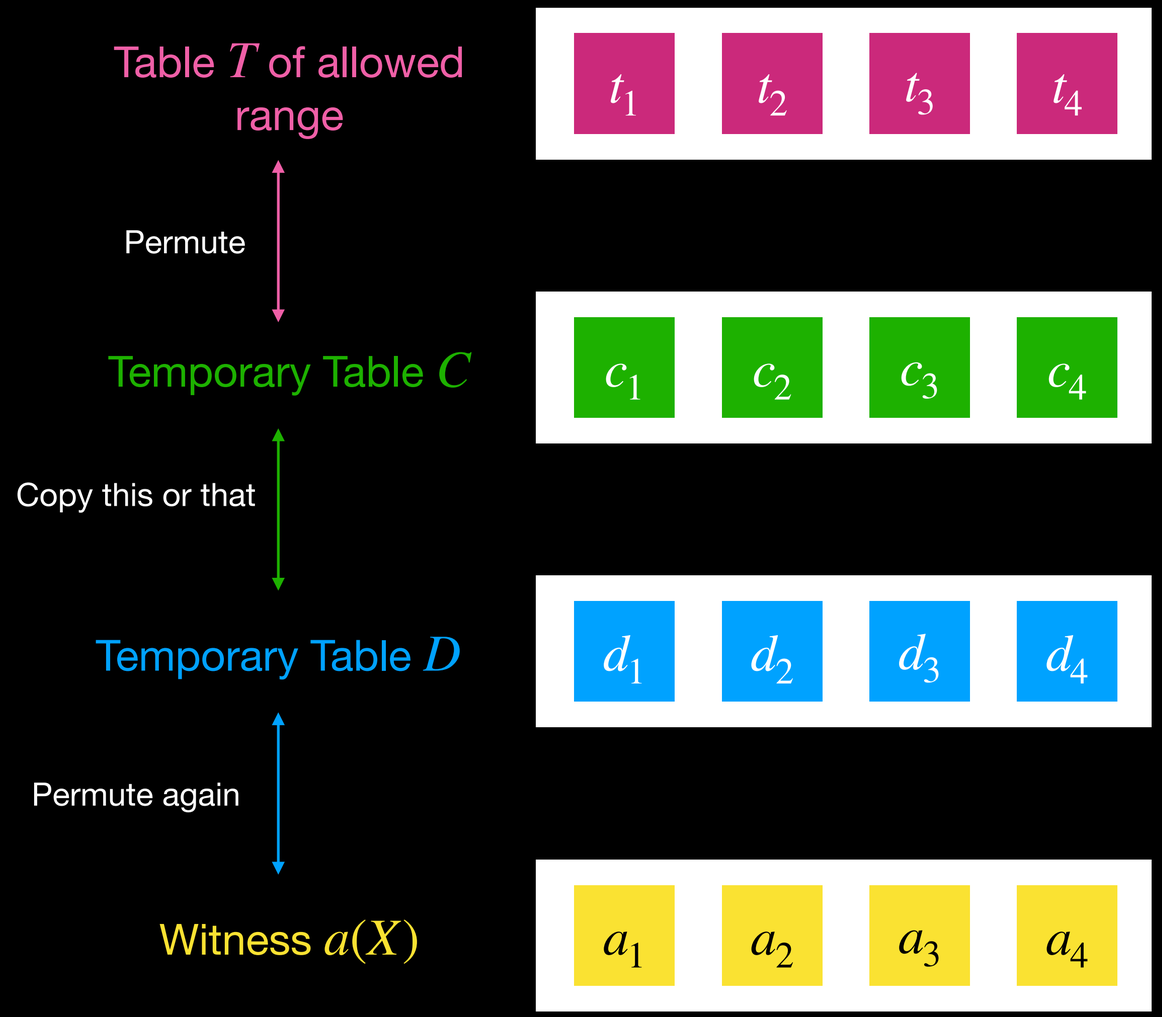
What if there are repeated witness values?

Then permutation is not satisfied.

The Halo2 Strategy

Can we use a permutation argument?

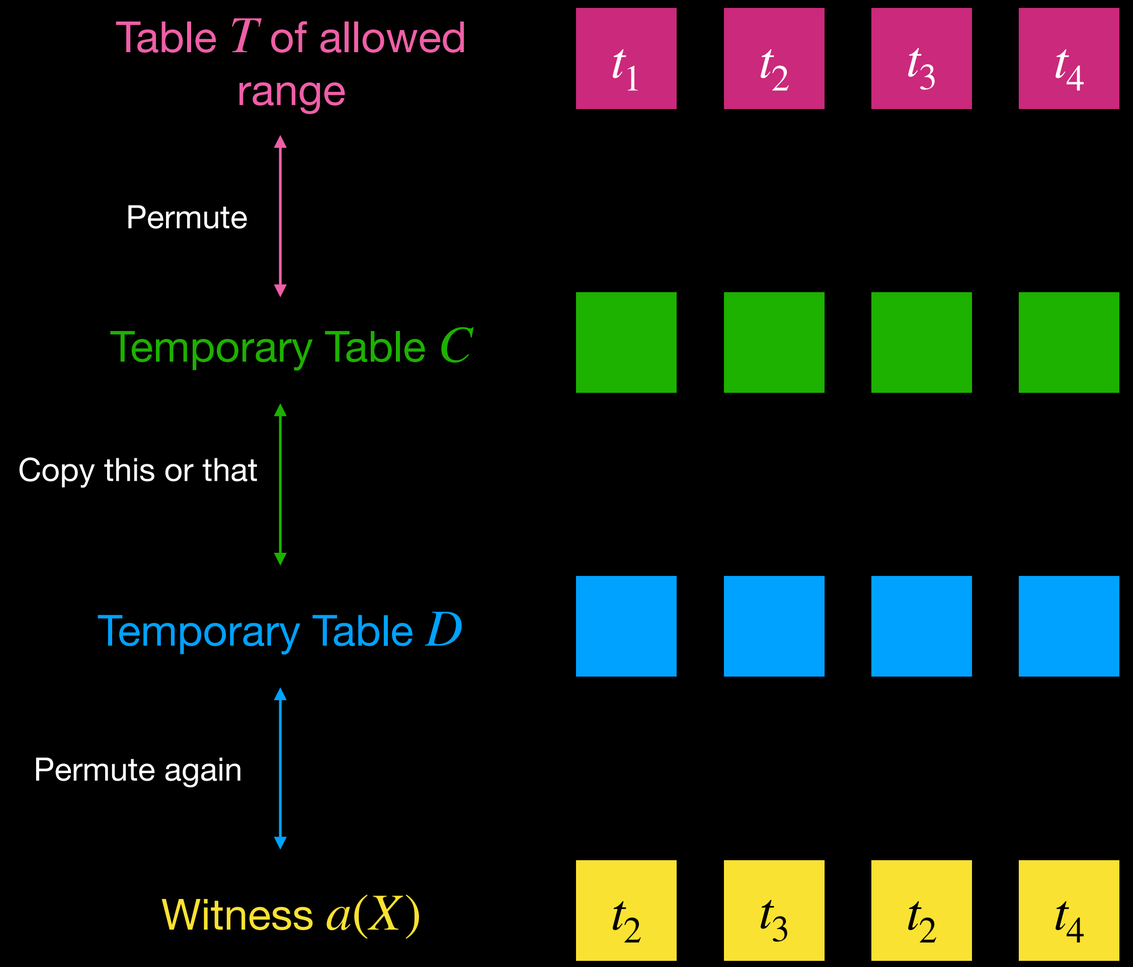
- Polynomial $t(X)$ such that:
  - $t(1) = 0$
  - $t(2) = 1$
  - $t(3) = 2$
  - $t(4) = 3$
- Witness polynomial $a(X)$ such that:
  - $a(1) = a_1$
  - $a(2) = a_2$
  - $a(3) = a_3$
  - $a(4) = a_4$
- Prove that $\{a_1, a_2, a_3, a_4\} \in \{t_1, t_2, t_3, t_4\}$



The Halo2 Strategy

Can we use a permutation argument?

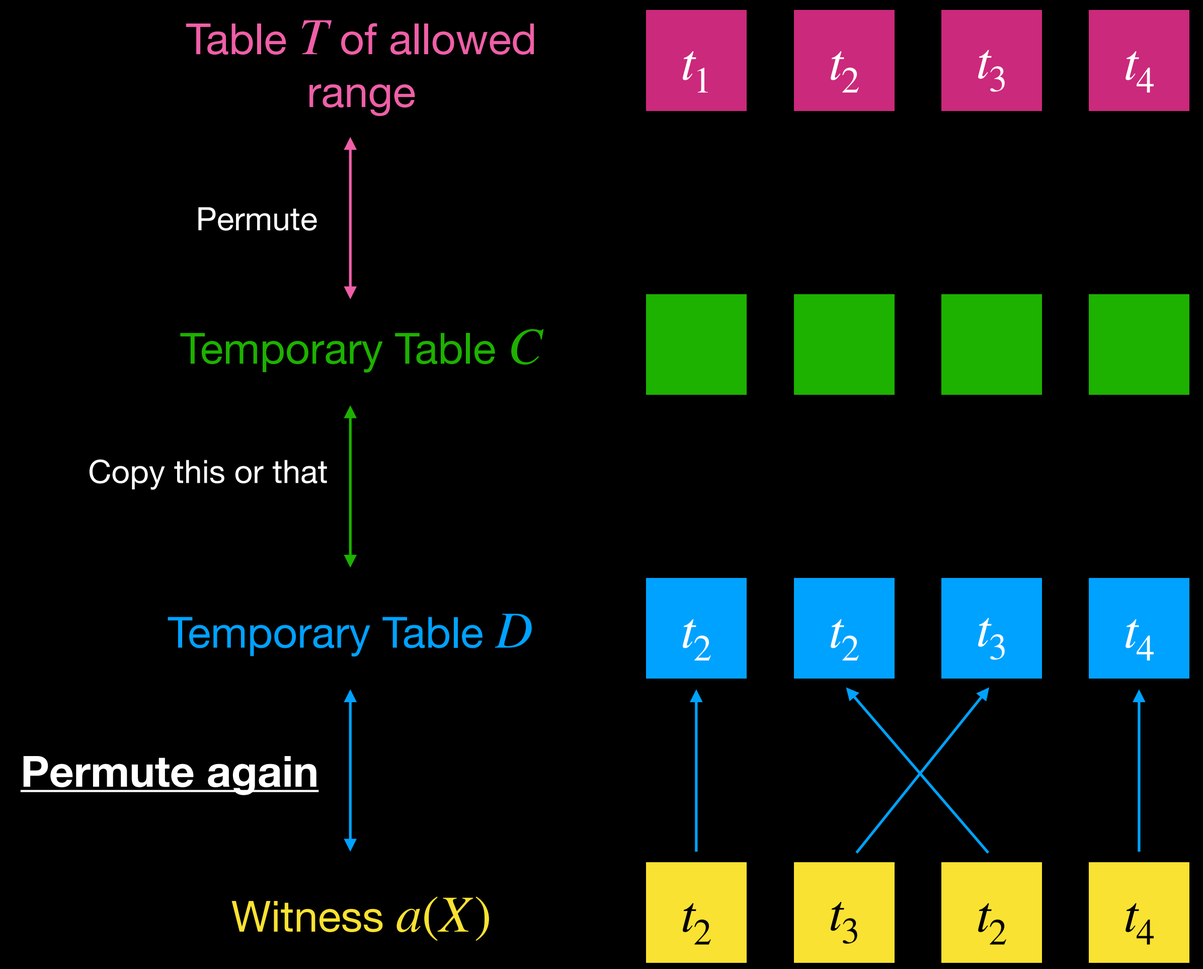
- Polynomial $t(X)$ such that:
  - $t(1) = 0$
  - $t(2) = 1$
  - $t(3) = 2$
  - $t(4) = 3$
- Witness polynomial $a(X)$ such that:
  - $a(1) = a_1$
  - $a(2) = a_2$
  - $a(3) = a_3$
  - $a(4) = a_4$
- Prove that $\{a_1, a_2, a_3, a_4\} \in \{t_1, t_2, t_3, t_4\}$



The Halo2 Strategy

Can we use a permutation argument?

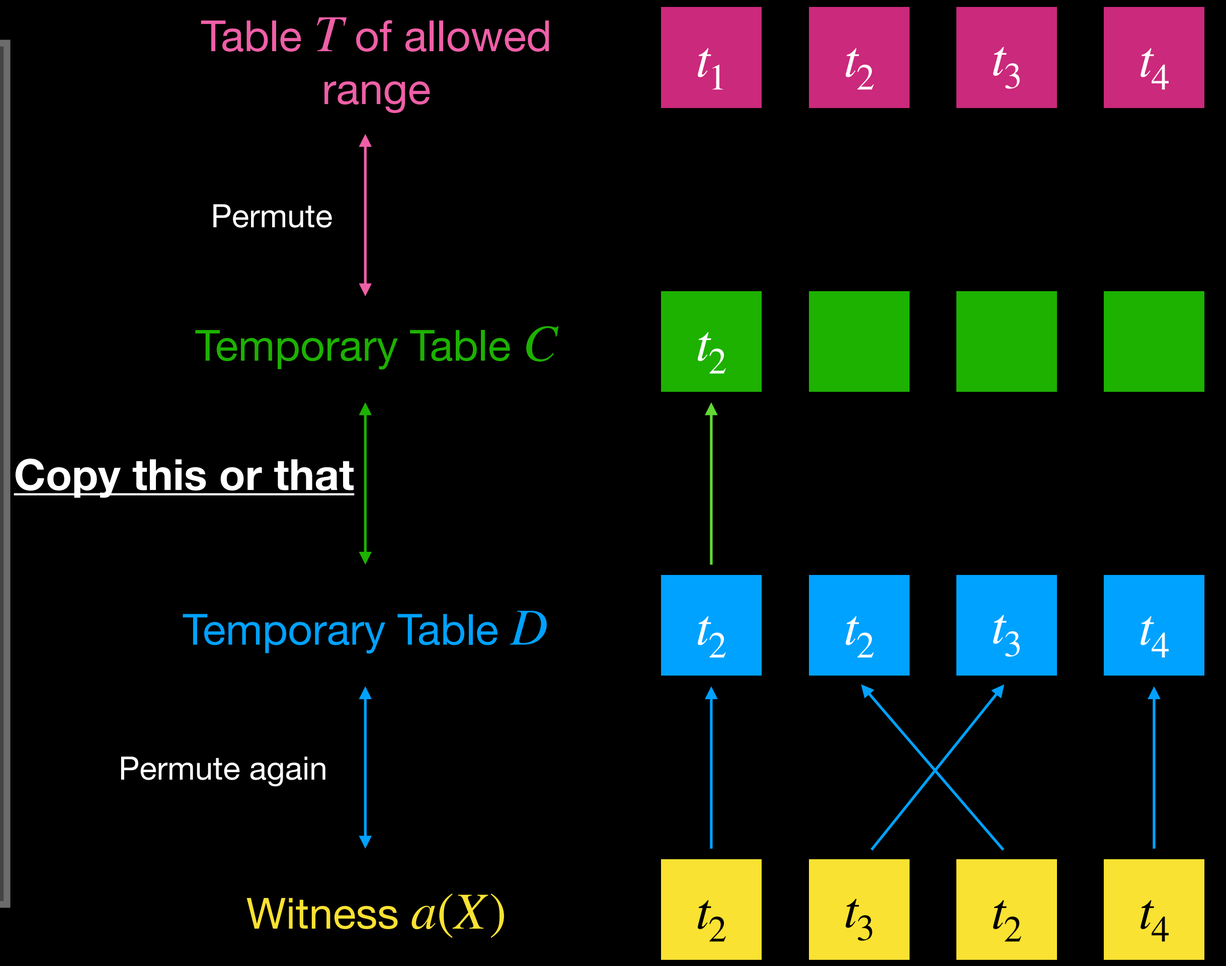
- Polynomial $t(X)$ such that:
  - $t(1) = 0$
  - $t(2) = 1$
  - $t(3) = 2$
  - $t(4) = 3$
- Witness polynomial $a(X)$ such that:
  - $a(1) = a_1$
  - $a(2) = a_2$
  - $a(3) = a_3$
  - $a(4) = a_4$
- Prove that $\{a_1, a_2, a_3, a_4\} \in \{t_1, t_2, t_3, t_4\}$



The Halo2 Strategy

Can we use a permutation argument?

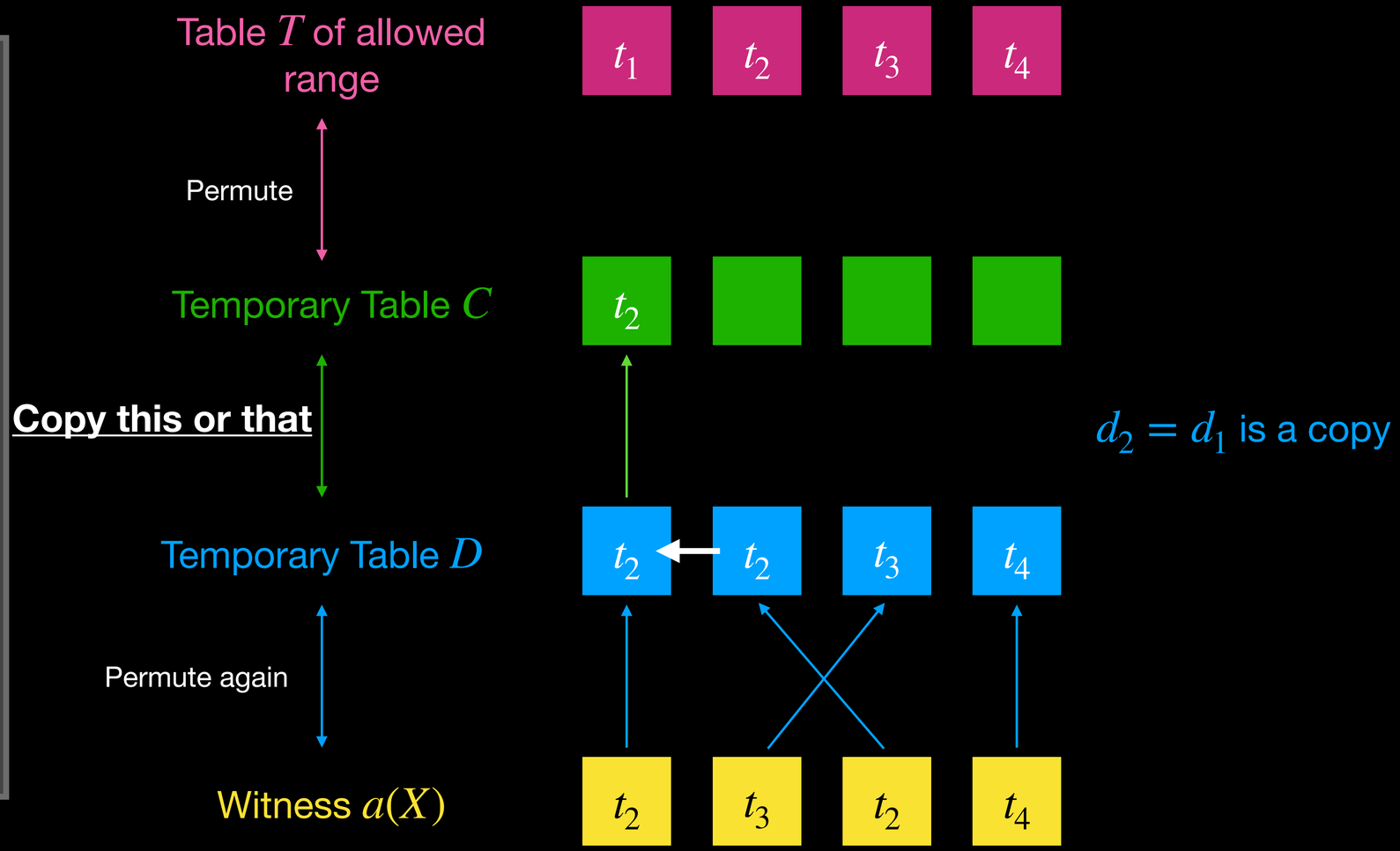
- Polynomial $t(X)$ such that:
  - $t(1) = 0$
  - $t(2) = 1$
  - $t(3) = 2$
  - $t(4) = 3$
- Witness polynomial $a(X)$ such that:
  - $a(1) = a_1$
  - $a(2) = a_2$
  - $a(3) = a_3$
  - $a(4) = a_4$
- Prove that $\{a_1, a_2, a_3, a_4\} \in \{t_1, t_2, t_3, t_4\}$



The Halo2 Strategy

Can we use a permutation argument?

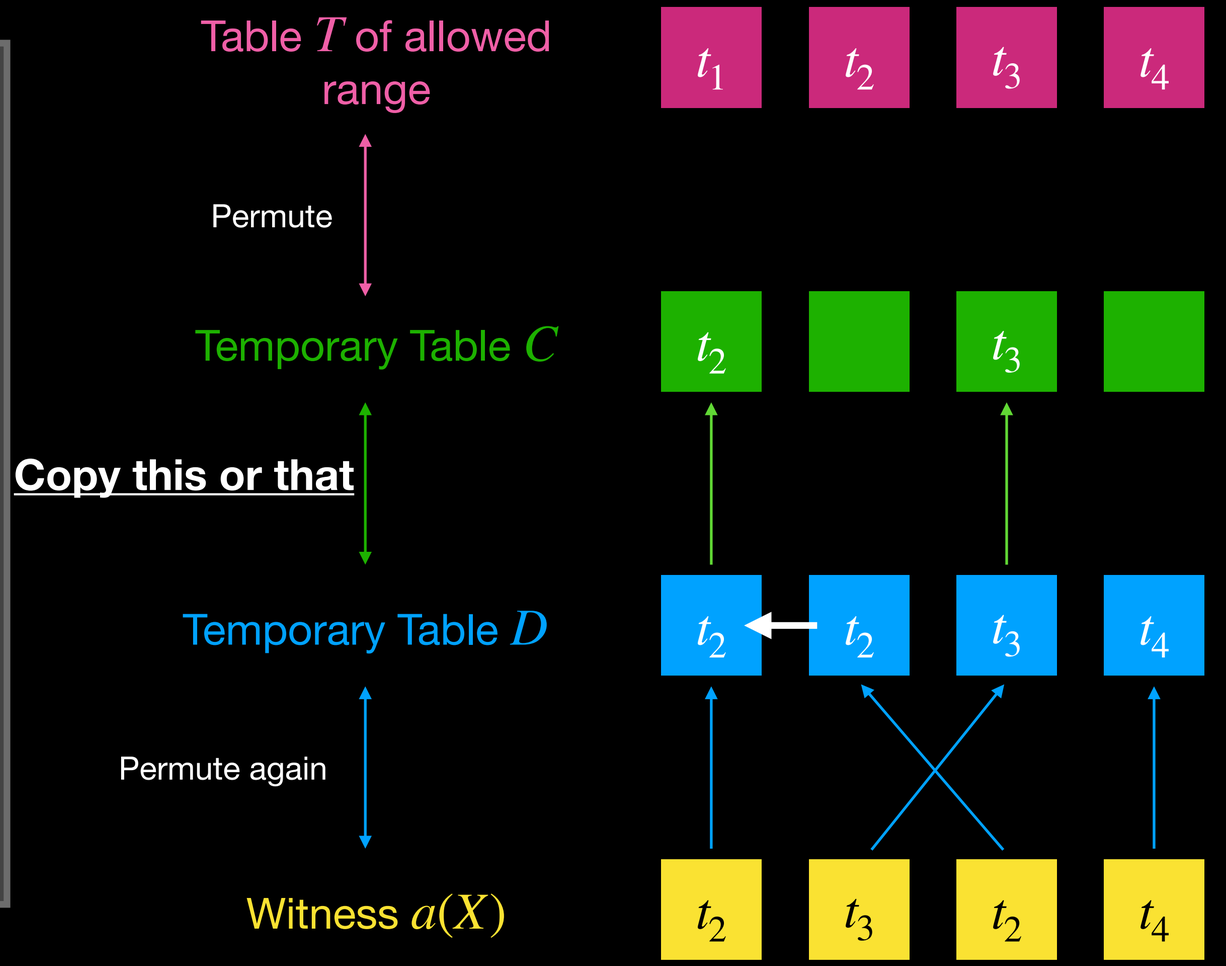
- Polynomial $t(X)$ such that:
  - $t(1) = 0$
  - $t(2) = 1$
  - $t(3) = 2$
  - $t(4) = 3$
- Witness polynomial $a(X)$ such that:
  - $a(1) = a_1$
  - $a(2) = a_2$
  - $a(3) = a_3$
  - $a(4) = a_4$
- Prove that $\{a_1, a_2, a_3, a_4\} \in \{t_1, t_2, t_3, t_4\}$



The Halo2 Strategy

Can we use a permutation argument?

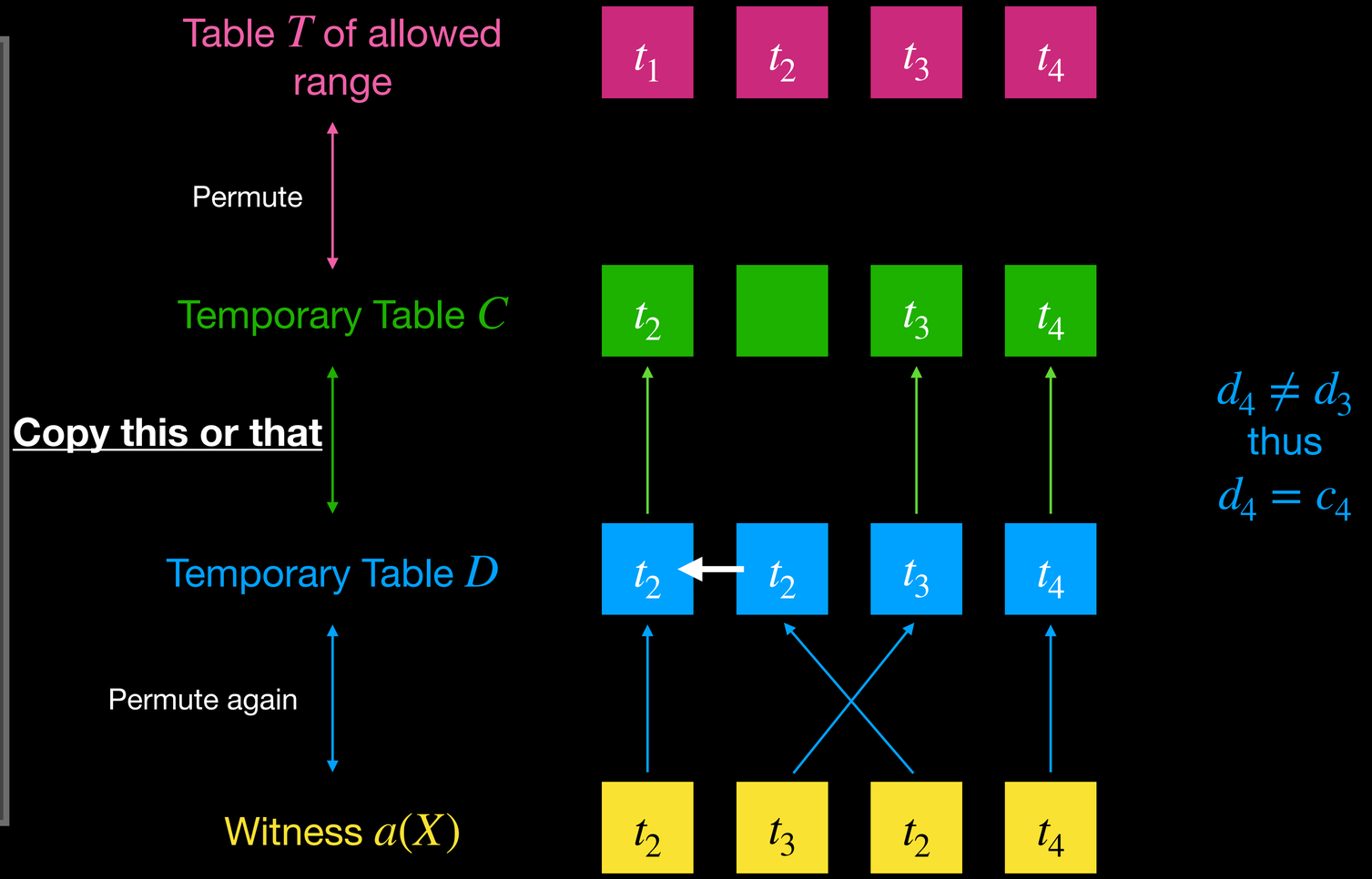
- Polynomial $t(X)$ such that:
  - $t(1) = 0$
  - $t(2) = 1$
  - $t(3) = 2$
  - $t(4) = 3$
- Witness polynomial $a(X)$ such that:
  - $a(1) = a_1$
  - $a(2) = a_2$
  - $a(3) = a_3$
  - $a(4) = a_4$
- Prove that $\{a_1, a_2, a_3, a_4\} \in \{t_1, t_2, t_3, t_4\}$



The Halo2 Strategy

Can we use a permutation argument?

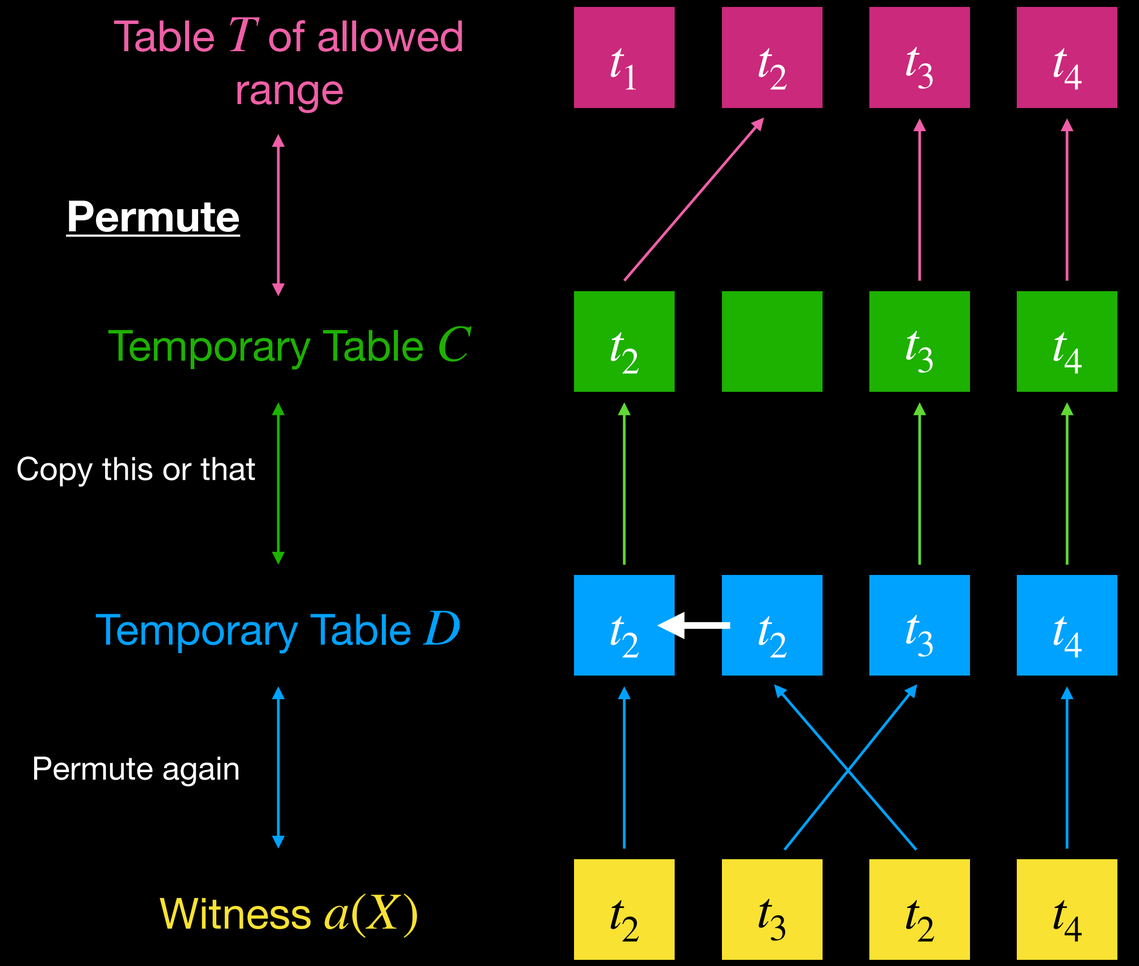
- Polynomial $t(X)$ such that:
  - $t(1) = 0$
  - $t(2) = 1$
  - $t(3) = 2$
  - $t(4) = 3$
- Witness polynomial $a(X)$ such that:
  - $a(1) = a_1$
  - $a(2) = a_2$
  - $a(3) = a_3$
  - $a(4) = a_4$
- Prove that $\{a_1, a_2, a_3, a_4\} \in \{t_1, t_2, t_3, t_4\}$



The Halo2 Strategy

Can we use a permutation argument?

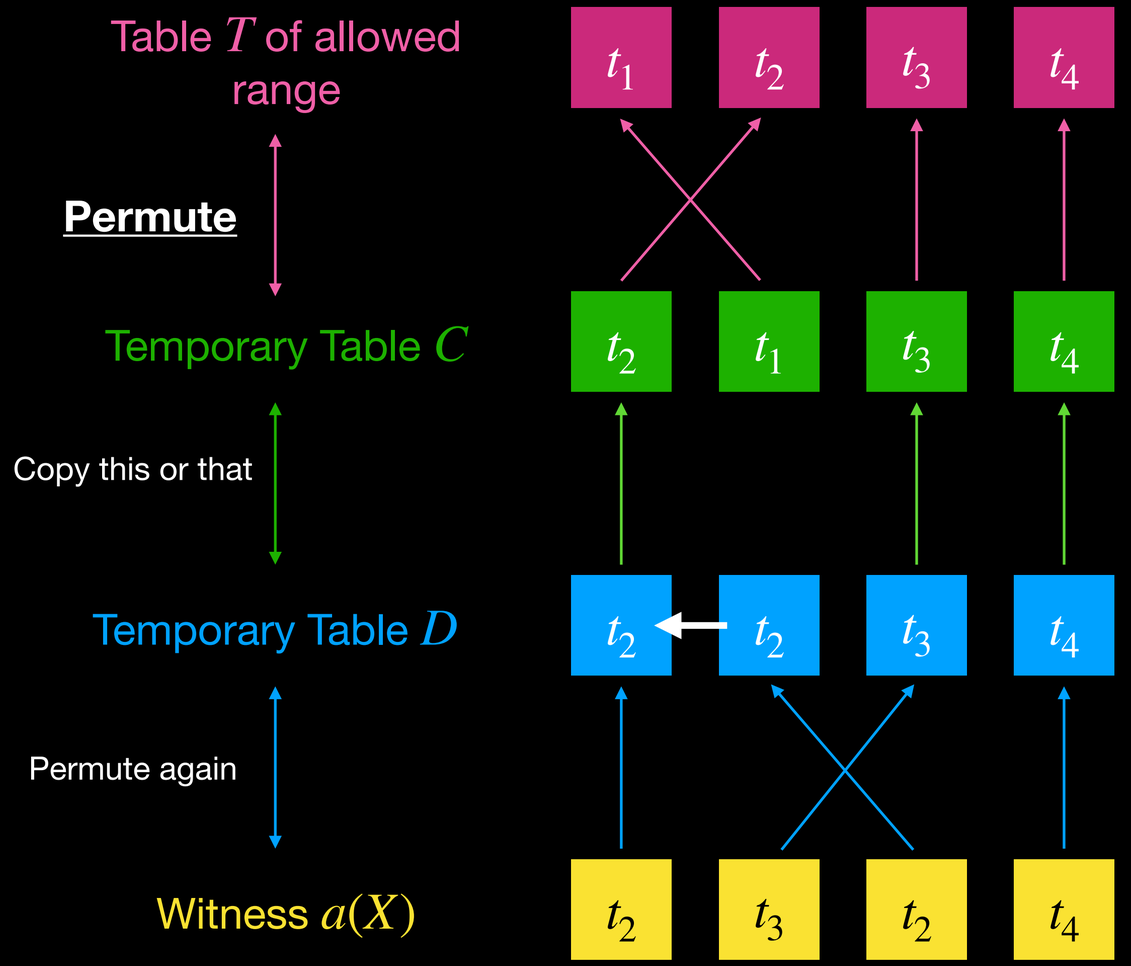
- Polynomial $t(X)$ such that:
  - $t(1) = 0$
  - $t(2) = 1$
  - $t(3) = 2$
  - $t(4) = 3$
- Witness polynomial $a(X)$ such that:
  - $a(1) = a_1$
  - $a(2) = a_2$
  - $a(3) = a_3$
  - $a(4) = a_4$
- Prove that $\{a_1, a_2, a_3, a_4\} \in \{t_1, t_2, t_3, t_4\}$



The Halo2 Strategy

Can we use a permutation argument?

- Polynomial $t(X)$ such that:
  - $t(1) = 0$
  - $t(2) = 1$
  - $t(3) = 2$
  - $t(4) = 3$
- Witness polynomial $a(X)$ such that:
  - $a(1) = a_1$
  - $a(2) = a_2$
  - $a(3) = a_3$
  - $a(4) = a_4$
- Prove that $\{a_1, a_2, a_3, a_4\} \in \{t_1, t_2, t_3, t_4\}$

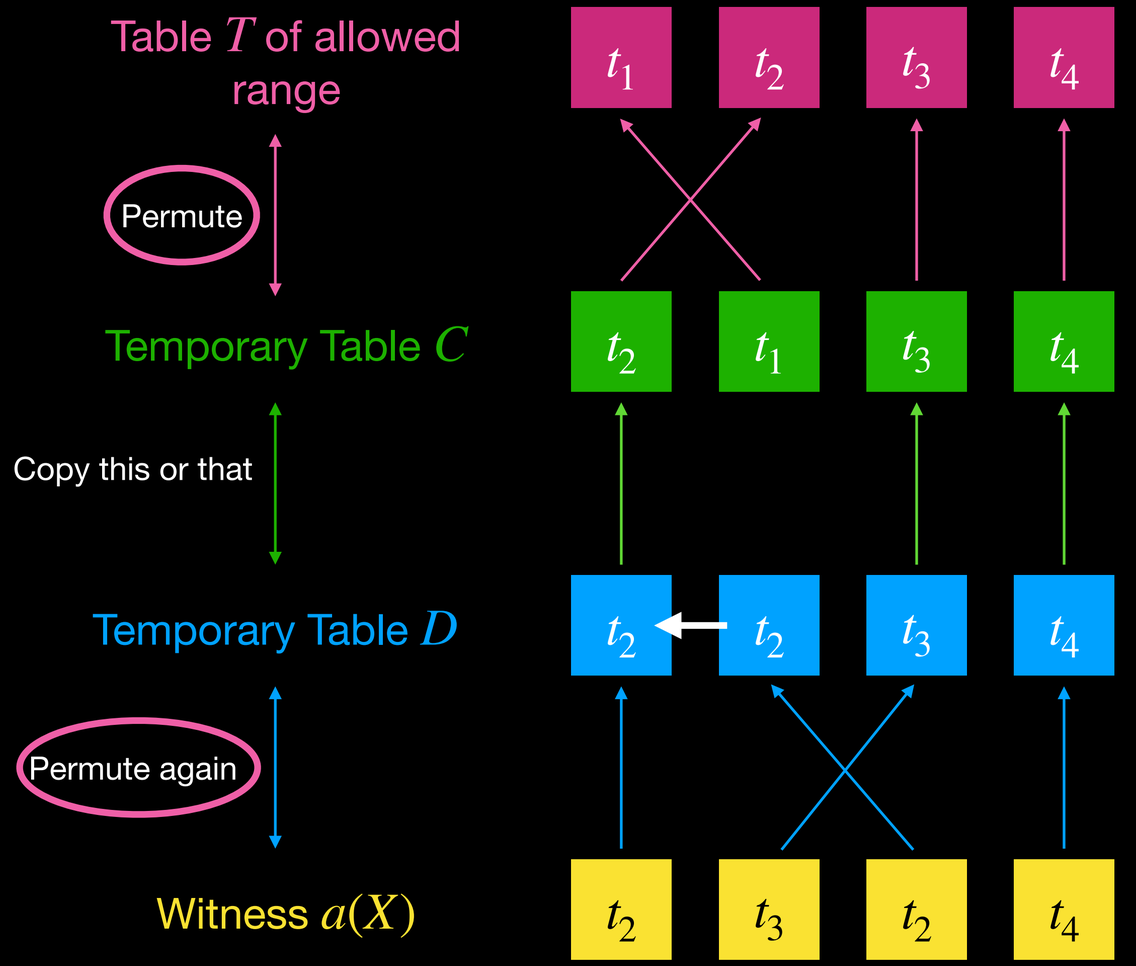


The Halo2 Strategy

Can we use a permutation argument?

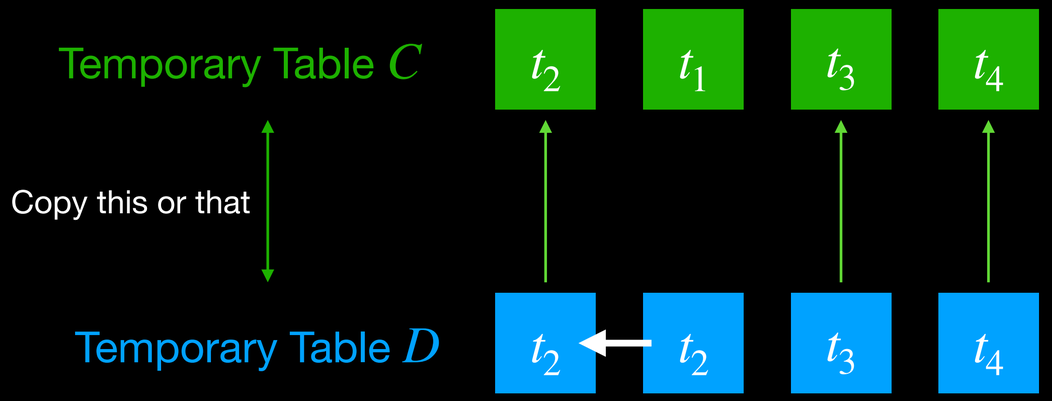
Permutation arguments discussed in prior talks.

- Polynomial $t(X)$ such that:
  - $t(1) = 0$
  - $t(2) = 1$
  - $t(3) = 2$
  - $t(4) = 3$
- Witness polynomial $a(X)$ such that:
  - $a(1) = a_1$
  - $a(2) = a_2$
  - $a(3) = a_3$
  - $a(4) = a_4$
- Prove that $\{a_1, a_2, a_3, a_4\} \in \{t_1, t_2, t_3, t_4\}$



The Halo2 Strategy

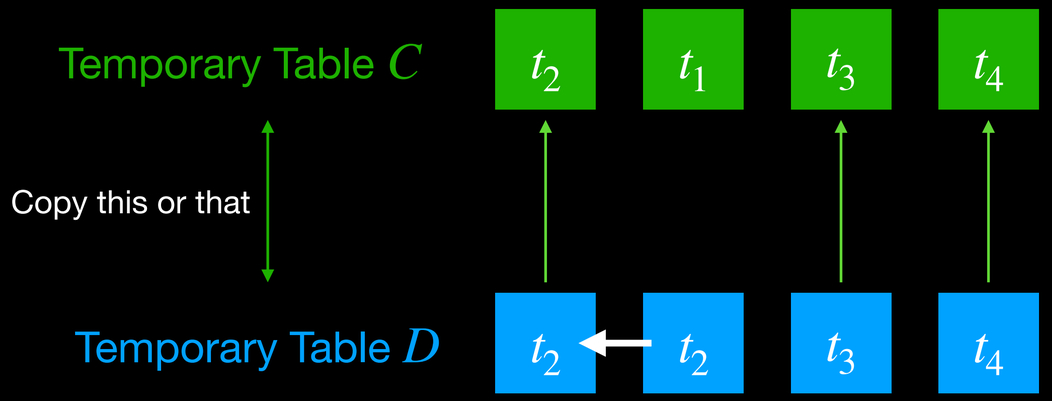
Copy this or that constraints?



- Aim to show that $c(X)$, $d(X)$ is such that:
  - $c(i) = d(i)$
  - OR $d(i+1) = d(i)$
- Prove that
  - $b(X)[c(X) - d(X)] + (\mathrm{id}(X) - b(X))[d(X) - d(X-1)] = q'(X)z(X)$
- Prove that
  - $b(X)(\mathrm{id}(X) - b(X)) = q(X)z(X)$

The Halo2 Strategy

Copy this or that constraints?

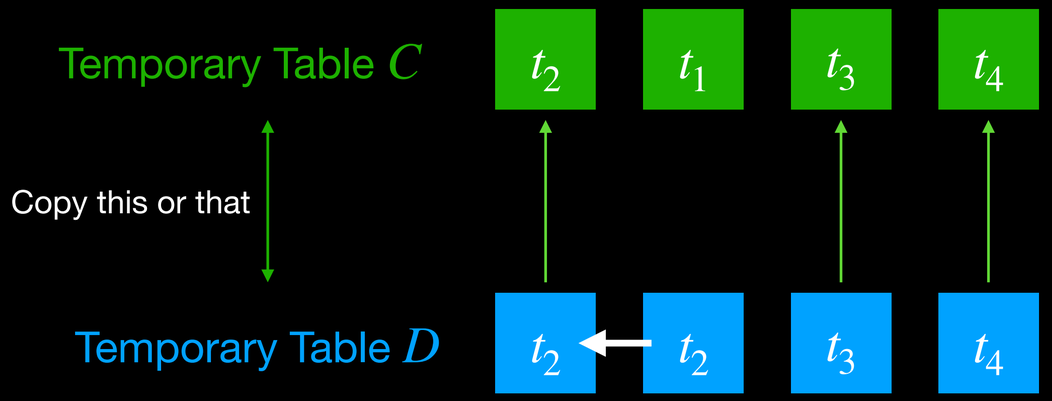


- Aim to show that $c(x),\ d(X)$ is such that:
  - $c(i) = d(i)$
  - OR $d(i+1) = d(i)$
- Prove that
  - $[c(X) - d(X)] = q'(X)z(X)$
- Prove that
  - $b(X)(\mathrm{id}(X) - b(X)) = q(X)z(X)$

When $c(i) = d(i)$ have $c(i) - d(i) = 0$

The Halo2 Strategy

Copy this or that constraints?



- Aim to show that $c(x),\ d(X)$ is such that:
  - $c(i) = d(i)$
  - OR $d(i+1) = d(i)$
- Prove that
  - $[c(X) - d(X)] = q'(X)z(X)$
- Prove that
  - $b(X)(\mathrm{id}(X) - b(X)) = q(X)z(X)$

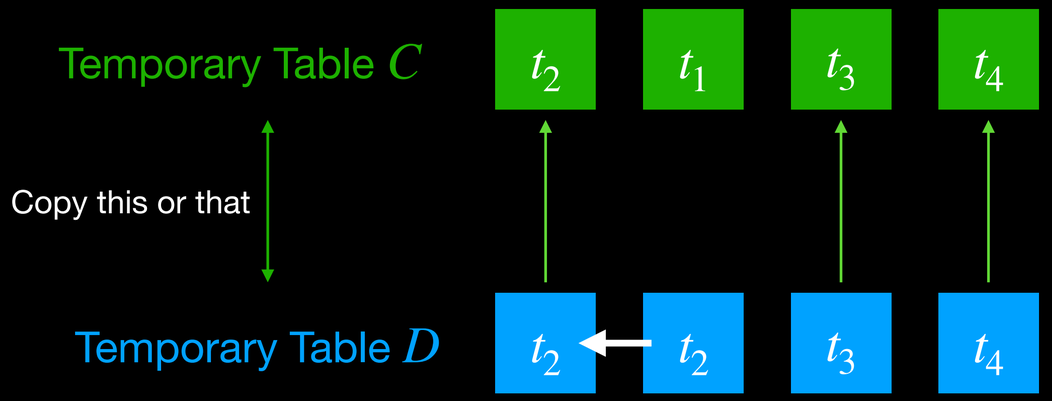
Recall that $z(X) = (X-1)(X-2)(X-3)(X-4)$

Vanishing polynomial

When $c(i) = d(i)$ have $c(i) - d(i) = 0$

The Halo2 Strategy

Copy this or that constraints?



- Aim to show that $c(x),\ d(X)$ is such that:
  - $c(i) = d(i)$
  - OR $d(i+1) = d(i)$
- Prove that
  - $[c(X) - d(X)] = q'(X)z(X)$
- Prove that
  - $b(X)(\mathrm{id}(X) - b(X)) = q(X)z(X)$

Recall that $z(X) = (X-1)(X-2)(X-3)(X-4)$

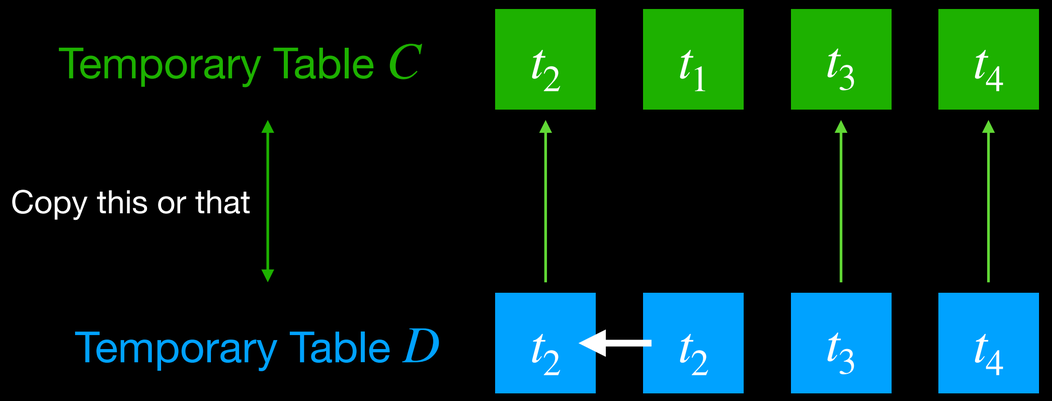
Vanishing polynomial

When $c(i) = d(i)$ have $c(i) - d(i) = 0$

Then copy this or that is not satisfied when $d(i+1) = d(i)$

The Halo2 Strategy

Copy this or that constraints?



- Aim to show that $c(x),\ d(X)$ is such that:
  - $c(i) = d(i)$
  - OR $d(i+1) = d(i)$
- Prove that
  - $b(X)[c(X) - d(X)] = q'(X)z(X)$
- Prove that
  - $b(X)(\mathrm{id}(X) - b(X)) = q(X)z(X)$

When $c(i) = d(i)$ have $b(i) = 1$

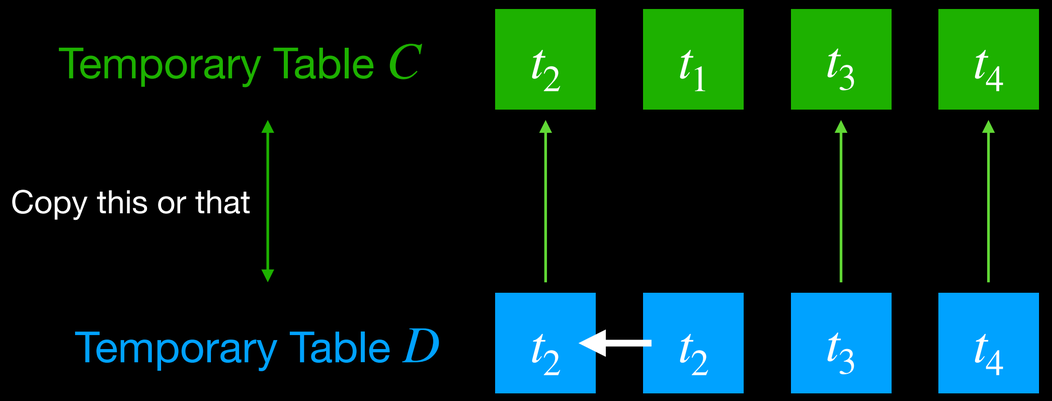
When $c(i) \neq d(i)$ have $b(i) = 0$

The identity polynomial $\mathrm{id}(i) = 1$

The vanishing polynomial $z(i) = 0$

The Halo2 Strategy

Copy this or that constraints?



- Aim to show that $c(x),\ d(X)$ is such that:
  - $c(i) = d(i)$
  - OR $d(i+1) = d(i)$
- Prove that
  - $b(X)[c(X) - d(X)] = q'(X)z(X)$
- Prove that
  - $b(X)(\mathrm{id}(X) - b(X)) = q(X)z(X)$

Then copy this or that is always satisfied when $b(i) = 0$

When $c(i) = d(i)$ have $b(i) = 1$

When $c(i) \neq d(i)$ have $b(i) = 0$

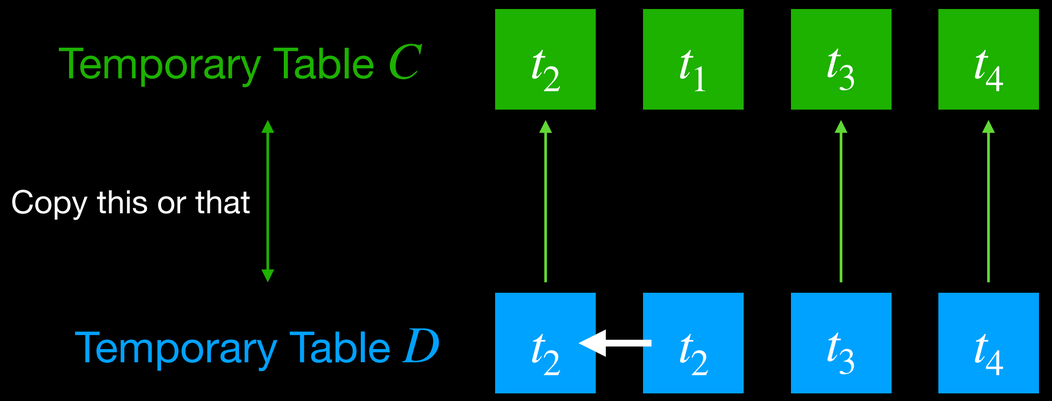
The identity polynomial $\mathrm{id}(i) = 1$

The vanishing polynomial $z(i) = 0$

Polynomial $b(X)$ *is such that* $b(i)(1 - b(i)) = 0 \Rightarrow b(i) \in \{0,1\}$

The Halo2 Strategy

Copy this or that constraints?



- Aim to show that $c(x),\ d(X)$ is such that:
  - $c(i) = d(i)$
  - OR $d(i+1) = d(i)$
- Prove that
  - $b(X)[c(X) - d(X)] = q'(X)z(X)$
- Prove that
  - $b(X)(\mathrm{id}(X) - b(X)) = q(X)z(X)$

$b(i) \in \{0,1\}$

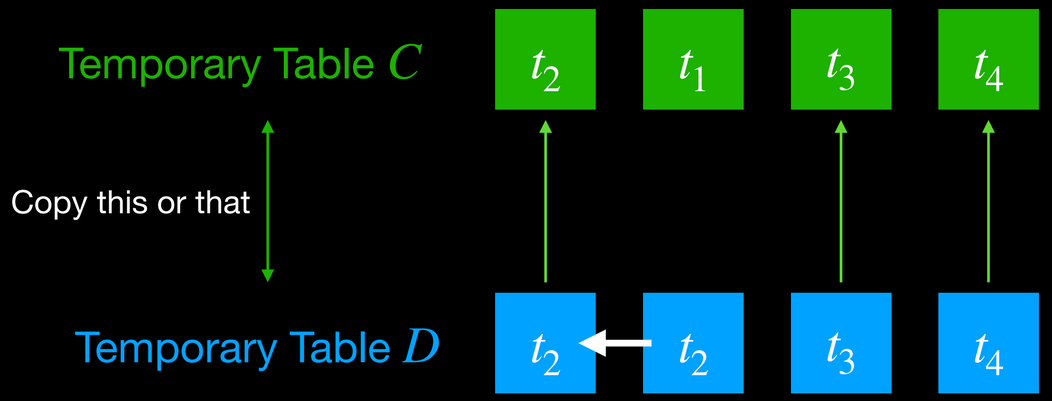
When $c(i) = d(i)$ have $b(i) = 1$

When $c(i) \neq d(i)$ have $b(i) = 0$

When $c(i) = d(i)$ have $1 \times [c(i) - d(i)] = 0$

The Halo2 Strategy

Copy this or that constraints?



- Aim to show that $c(x),\ d(X)$ is such that:
  - $c(i) = d(i)$
  - OR $d(i+1) = d(i)$
- Prove that
  - $b(X)[c(X) - d(X)] + (\mathrm{id}(X) - b(X))[d(X) - d(X-1)] = q'(X)z(X)$
- Prove that
  - $b(X)(\mathrm{id}(X) - b(X)) = q(X)z(X)$

$b(X)$ *is such that* $b(i) \in \{0,1\}$

When $c(i) = d(i)$ have $b(i) = 1$

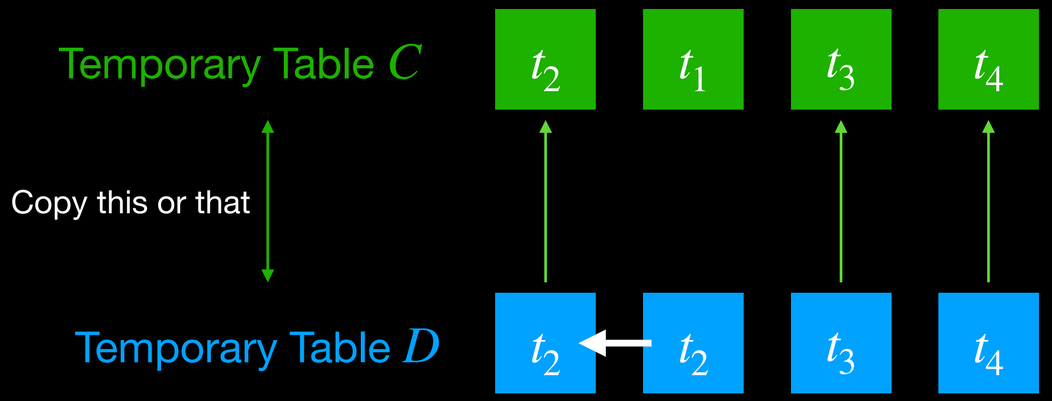
When $c(i) \neq d(i)$ have $b(i) = 0$

When $c(i) = d(i)$ have $1 \times [c(i) - d(i)] + 0 = 0$

When $d(i+1) = d(i)$ have $0 + (1-0) \times [d(i+1) - d(i)] = 0$

The Halo2 Strategy

Copy this or that constraints?



- Aim to show that $c(x),\ d(X)$ is such that:
  - $c(i) = d(i)$
  - OR $d(i+1) = d(i)$
- Prove that
  - $b(X)[c(X) - d(X)] + (\mathrm{id}(X) - b(X))[d(X) - d(X-1)] = q'(X)z(X)$
- Prove that
  - $b(X)(\mathrm{id}(X) - b(X)) = q(X)z(X)$

$b(X)$ *is such that* $b(i) \in \{0,1\}$

When $c(i) = d(i)$ have $b(i) = 1$

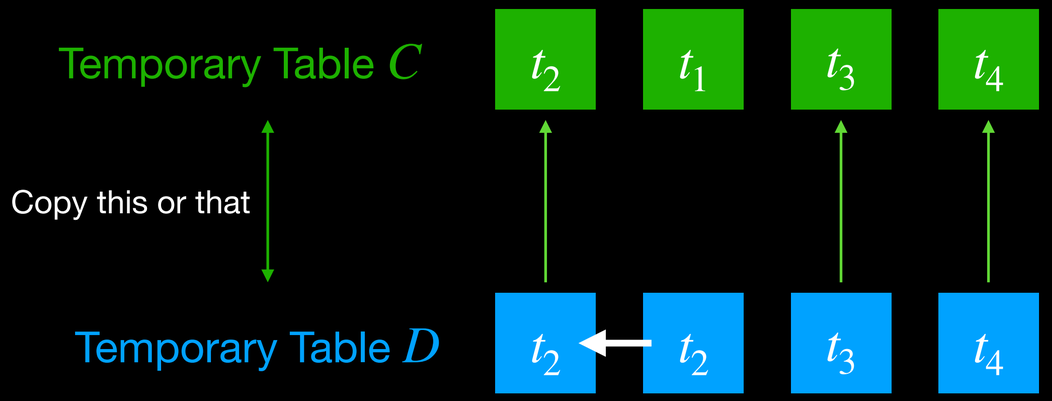
When $c(i) \neq d(i)$ have $b(i) = 0$

When $c(i) = d(i)$ have $1 \times [c(i) - d(i)] + 0 = 0$

When $d(i+1) = d(i)$ have $0 + (1-0) \times [d(i+1) - d(i)] = 0$

The Halo2 Strategy

Copy this or that constraints?



- Aim to show that $c(x),\ d(X)$ is such that:
  - $c(i) = d(i)$
  - OR $d(i+1) = d(i)$
- Prove that
  - $b(X)[c(X) - d(X)] + (\mathrm{id}(X) - b(X))[d(X) - d(X-1)] = q'(X)z(X)$
- Prove that
  - $b(X)(\mathrm{id}(X) - b(X)) = q(X)z(X)$

Works

$b(X)$ *is such that* $b(i) \in \{0,1\}$

When $c(i) = d(i)$ have $b(i) = 1$

When $c(i) \neq d(i)$ have $b(i) = 0$

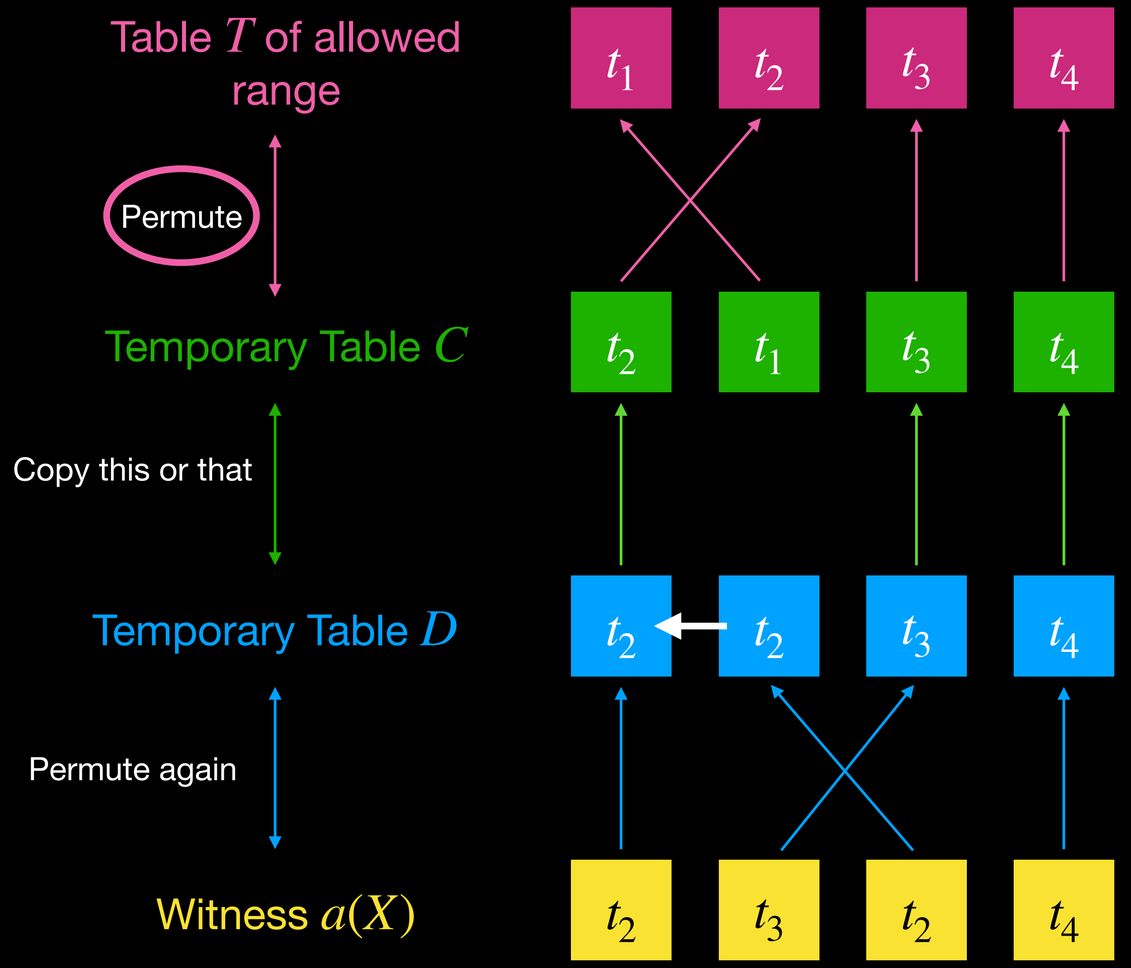
When $c(i) = d(i)$ have $1 \times [c(i) - d(i)] + 0 = 0$

When $d(i+1) = d(i)$ have $0 + (1-0) \times [d(i+1) - d(i)] = 0$

The Halo2 Strategy

Slow for big tables?

- Halo2 runs a permutation argument over $t(X)$.
- Prover time depends linearly on $t(X)$.
- Want large tables e.g. $\{t_1, \ldots, t_{2^{27}}\}$



The Caulk+ Strategy

Fast for big tables.

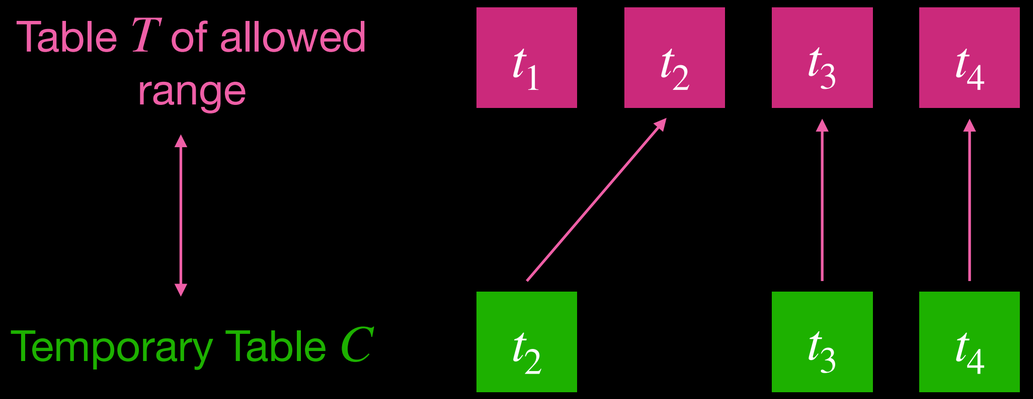
Caulk+: Table-independent lookup arguments

Jim Posen[1] and Assimakis A. Kattis[2]

[1] Ulvetanna jimpo AT ulvetanna.io
[2] New York University kattis AT cs.nyu.edu

- Halo2 runs a permutation argument over $t(X)$.
- Prover time depends linearly on $t(X)$.
- Want large tables e.g. $\{t_1, \ldots, t_{2^{27}}\}$
- We would like a temporary table $c(X)$ that is independent of the size of $t(X)$.

Possible with preprocessing.



The Caulk+ Strategy

Preprocessing.

Fast amortized KZG proofs

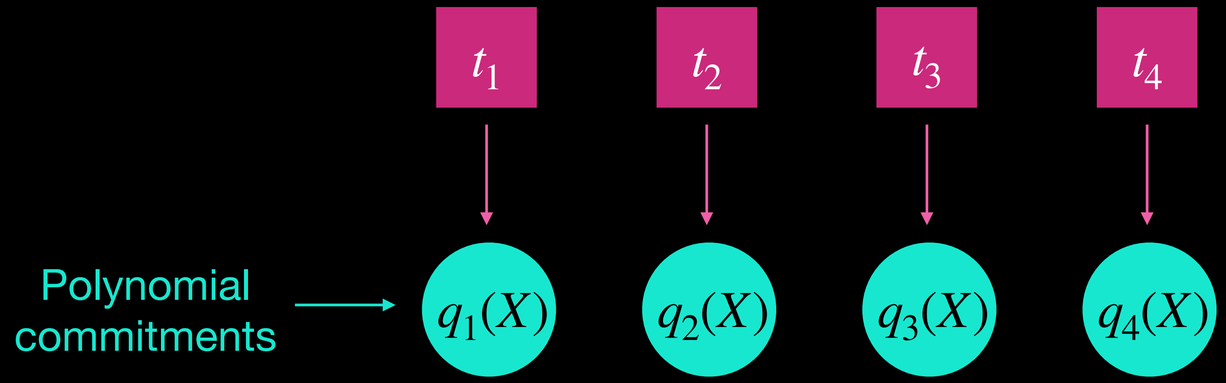
Dankrad Feist[*1] and Dmitry Khovratovich[†2]

[1,2]Ethereum Foundation

January 11, 2023

Abstract

In this note we explain how to compute n KZG proofs for a polynomial of degree d in time superlinear of $(t + d)$. Our technique is used in lookup arguments and vector commitment schemes.



$$t(X) - t(i) = q_i(X)(X - i)$$

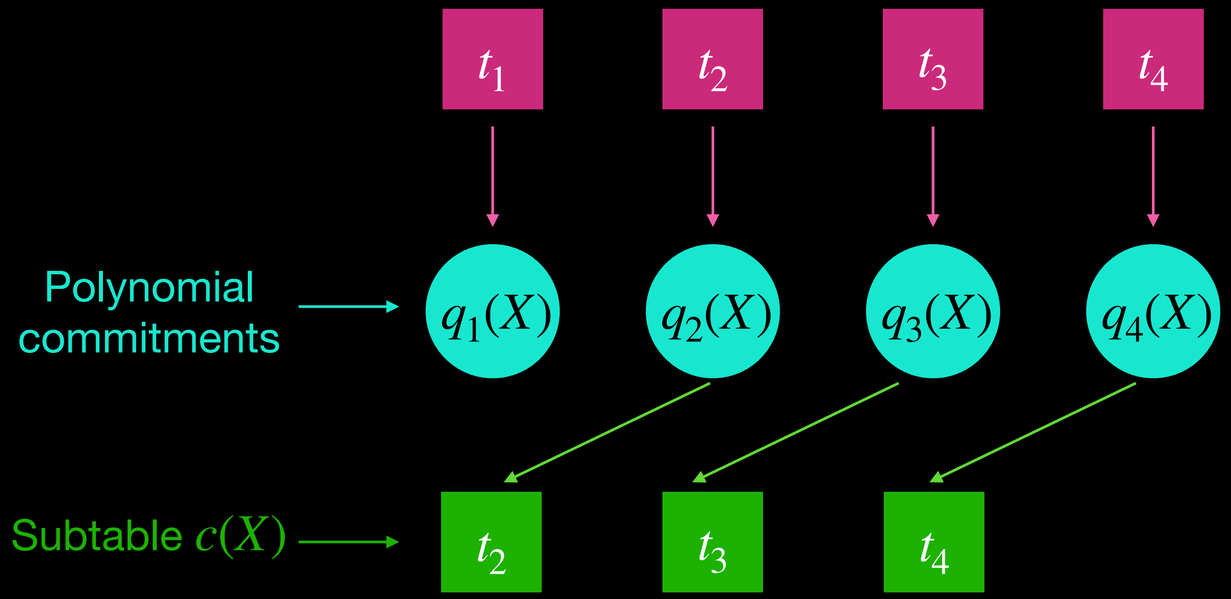
- Polynomial commitments $q_i(X)$ can be precomputed quickly.

The Caulk+ Strategy

Obtaining a subtable

If $c(X)$ defined over I is a subtable then

$$t(X) - c(X) = q(X)z_I(X)$$



- Polynomial commitments $q_i(X)$ can be precomputed quickly.
- Subtable $c(X)$ contains entries in $t(X)$ over domain $z_I(X)$

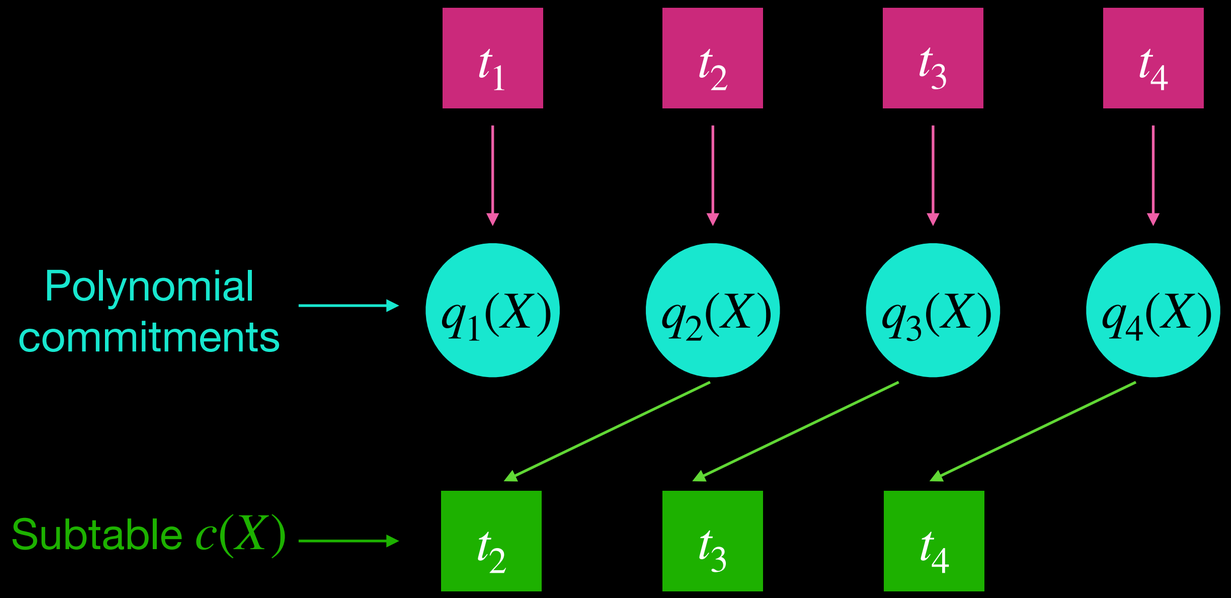
The Caulk+ Strategy

Obtaining a subtable

$$t(X) - c(X) = q(X)z_I(X)$$

Vanishing polynomial over $I = \{2,3,4\}$

$$z_I(X) = (X-2)(X-3)(X-4)$$



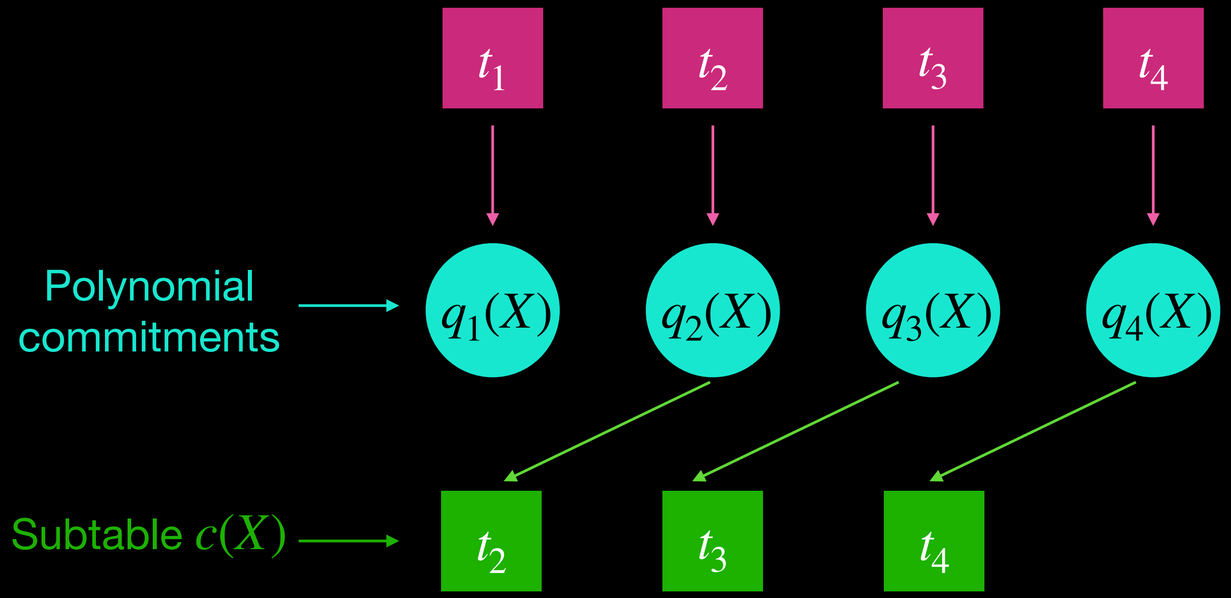
- At all points in $i \in I$ require $c(i) = t(i)$.
- Can prove $z_I(X)$ divides $z(X)$ and thus is valid.

- Polynomial commitments $q_i(X)$ can be precomputed quickly.
- Subtable $c(X)$ contains entries in $t(X)$ over domain $z_I(X)$

The Caulk+ Strategy

Obtaining a subtable

$$t(X) - c(X) = q(X)z_I(X)$$



Precomputed values such that

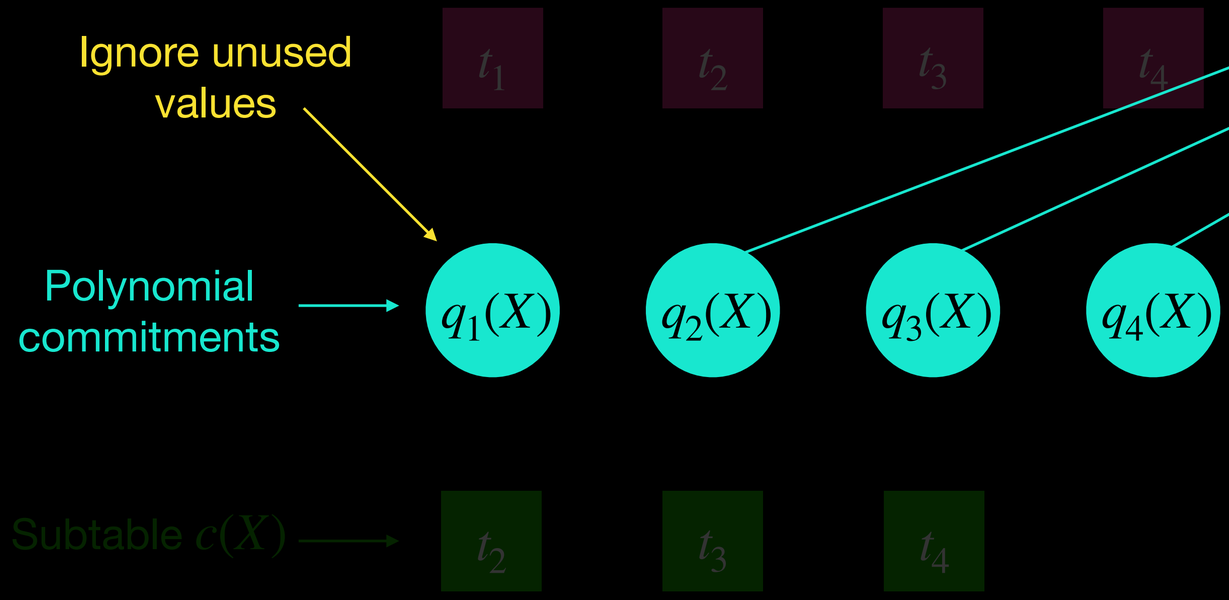
- $t(X) - c(2) = q_2(X)(X - 2)$
- $t(X) - c(3) = q_3(X)(X - 3)$
- $t(X) - c(4) = q_4(X)(X - 4)$

- Polynomial commitments $q_i(X)$ can be precomputed quickly.
- Subtable $c(X)$ contains entries in $t(X)$ over domain $z_I(X)$

The Caulk+ Strategy

Obtaining a subtable

$$t(X) - c(X) = q(X)z_I(X)$$



Precomputed values such that

- $t(X) - c(2) = q_2(X)(X - 2)$
- $t(X) - c(3) = q_3(X)(X - 3)$
- $t(X) - c(4) = q_4(X)(X - 4)$

Can quickly compute

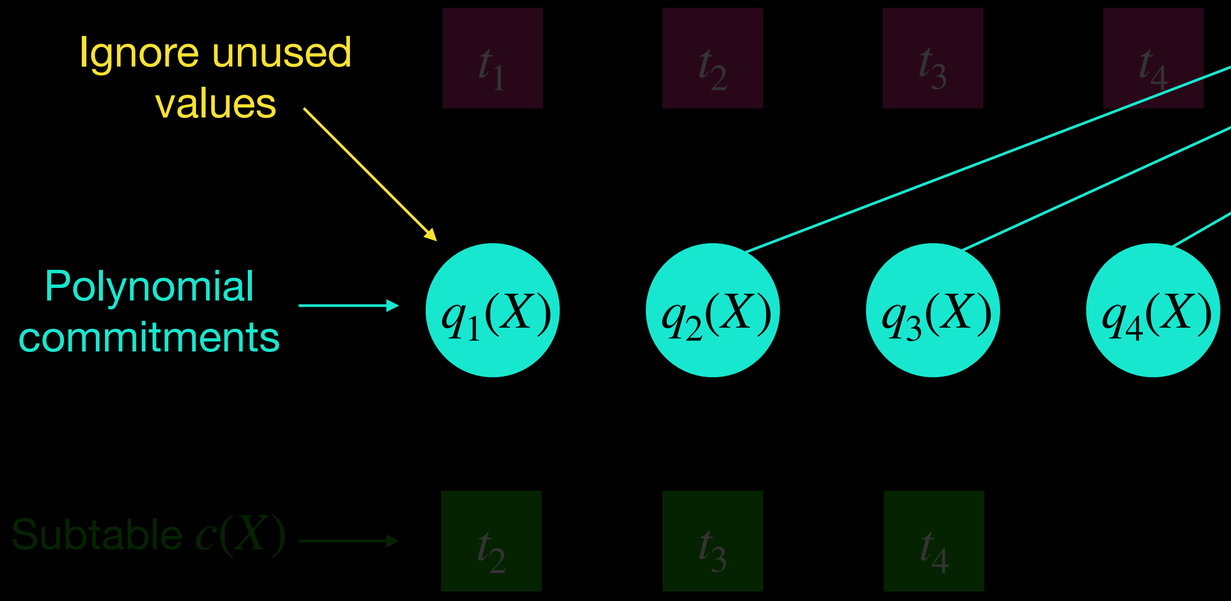
$$q(X) = f(\ q_2(X),\ q_3(X),\ q_4(X)\ )$$

- Polynomial commitments $q_i(X)$ can be precomputed quickly.
- Subtable $c(X)$ contains entries in $t(X)$ over domain $z_I(X)$

The Caulk+ Strategy

Obtaining a subtable

$$t(X) - c(X) = q(X)z_I(X)$$



Precomputed values such that

- $t(X) - c(2) = q_2(X)(X - 2)$
- $t(X) - c(3) = q_3(X)(X - 3)$
- $t(X) - c(4) = q_4(X)(X - 4)$

Can quickly compute

$$q(X) = f(\ q_2(X),\ q_3(X),\ q_4(X)\ )$$

Once we have a subtable how do we use it?

- Polynomial commitments $q_i(X)$ can be precomputed quickly.
- Subtable $c(X)$ contains entries in $t(X)$ over domain $z_I(X)$

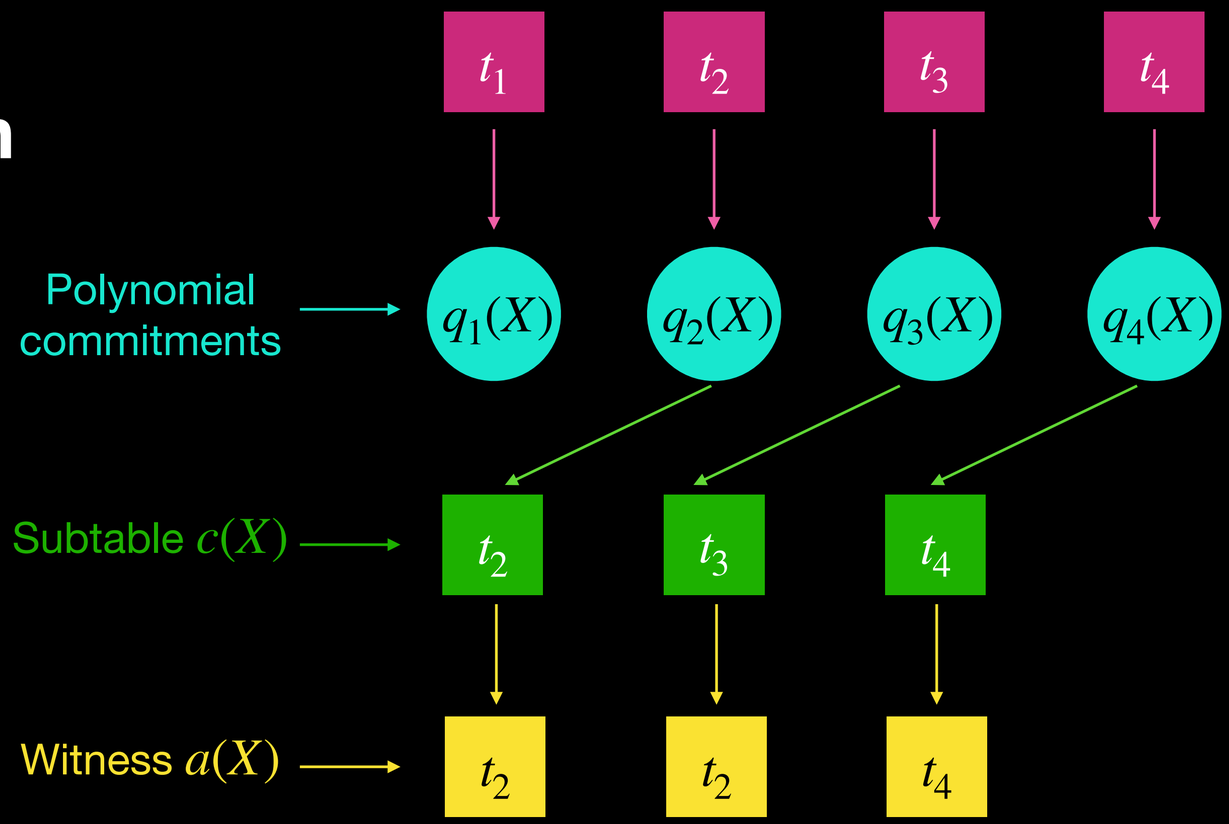
The Baloo Strategy

Prove lookup over hidden domain

Ba*loo*: Nearly Optimal Lookup Arguments

Arantxa Zapico*, Ariel Gabizon[3], Dmitry Khovratovich[1], Mary Maller[1], and Carla Ràfols[2]

[1] Ethereum Foundation
[2] Universitat Pompeu Fabra
[3] Zeta Function Technologies
arantxa.zapico@upf.edu, ariel.gabizon@gmail.com, khovratovich@gmail.com, mary.maller@ethereum.org, carla.rafols@upf.edu



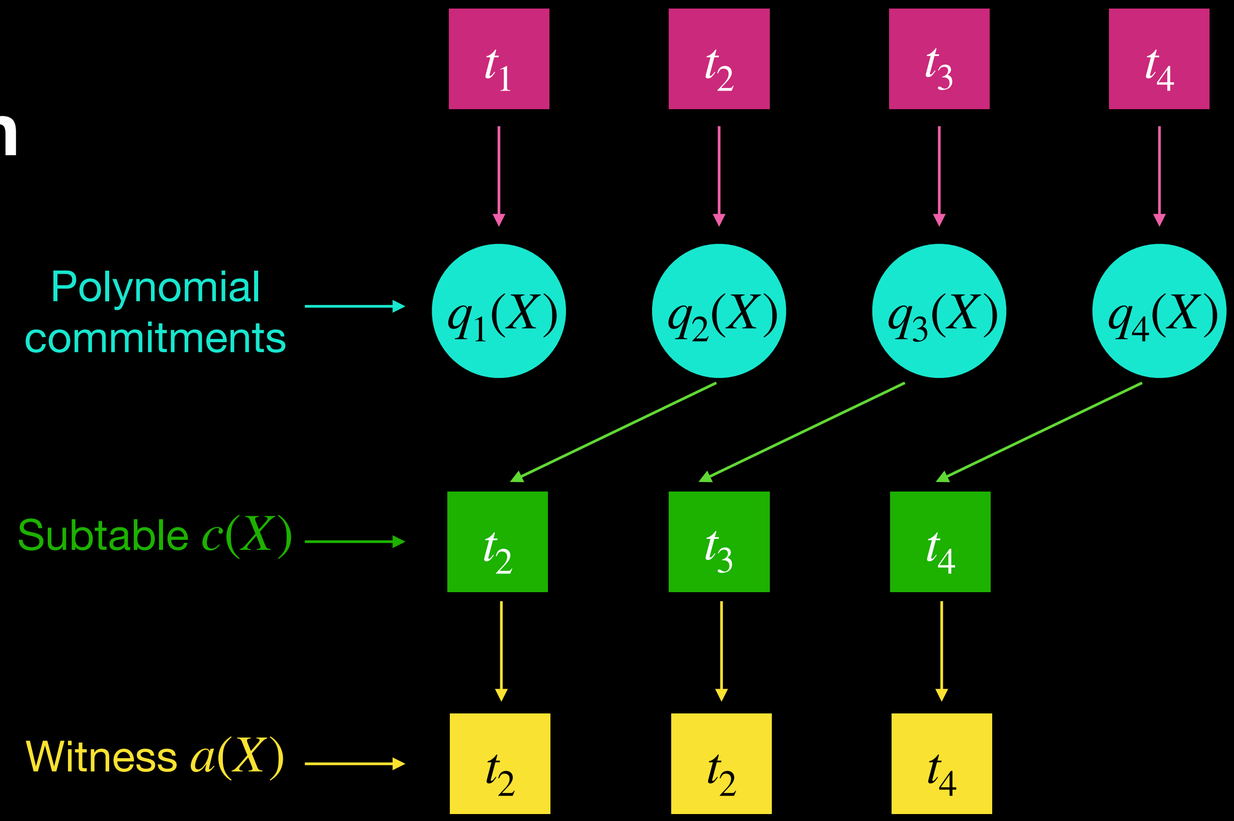
- Polynomial commitments $q_i(X)$ can be precomputed quickly.
- Subtable $c(X)$ contains entries in $t(X)$ over domain $z_I(X)$

The Baloo Strategy

Prove lookup over hidden domain

$$\sum_{j \in I} e_{i,j} c(j) = a_i$$

Lincheck Argument



Aurora: Transparent Succinct Arguments for R1CS

Eli Ben-Sasson
eli@cs.technion.ac.il
Technion

Alessandro Chiesa
alexch@berkeley.edu
UC Berkeley

Michael Riabzev
mriabzev@cs.technion.ac.il
Technion

Nicholas Spooner
nick.spooner@berkeley.edu
UC Berkeley

Madars Virza
madars@mit.edu
MIT Media Lab

Nicholas P. Ward
npward@berkeley.edu
UC Berkeley

May 8, 2019

- Polynomial commitments $q_i(X)$ can be precomputed quickly.
- Subtable $c(X)$ contains entries in $t(X)$ over domain $z_I(X)$

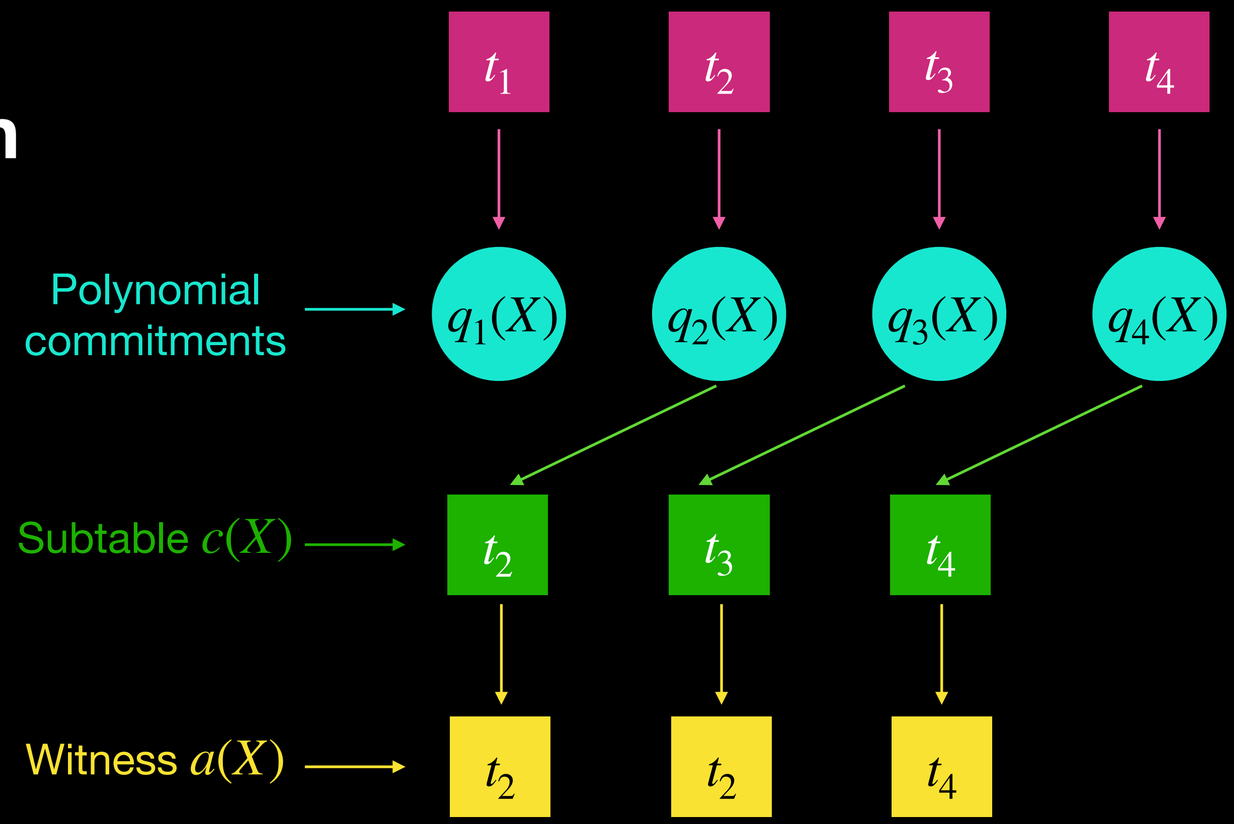
The Baloo Strategy

Prove lookup over hidden domain

$$\sum_{j \in I} e_{i,j} c(j) = a_i$$

Lincheck Argument over unit matrix

$$\mathbf{e}_i \in \{(1,0,0), (0,1,0), (0,0,1)\}$$



- Polynomial commitments $q_i(X)$ can be precomputed quickly.
- Subtable $c(X)$ contains entries in $t(X)$ over domain $z_I(X)$

The Baloo Strategy

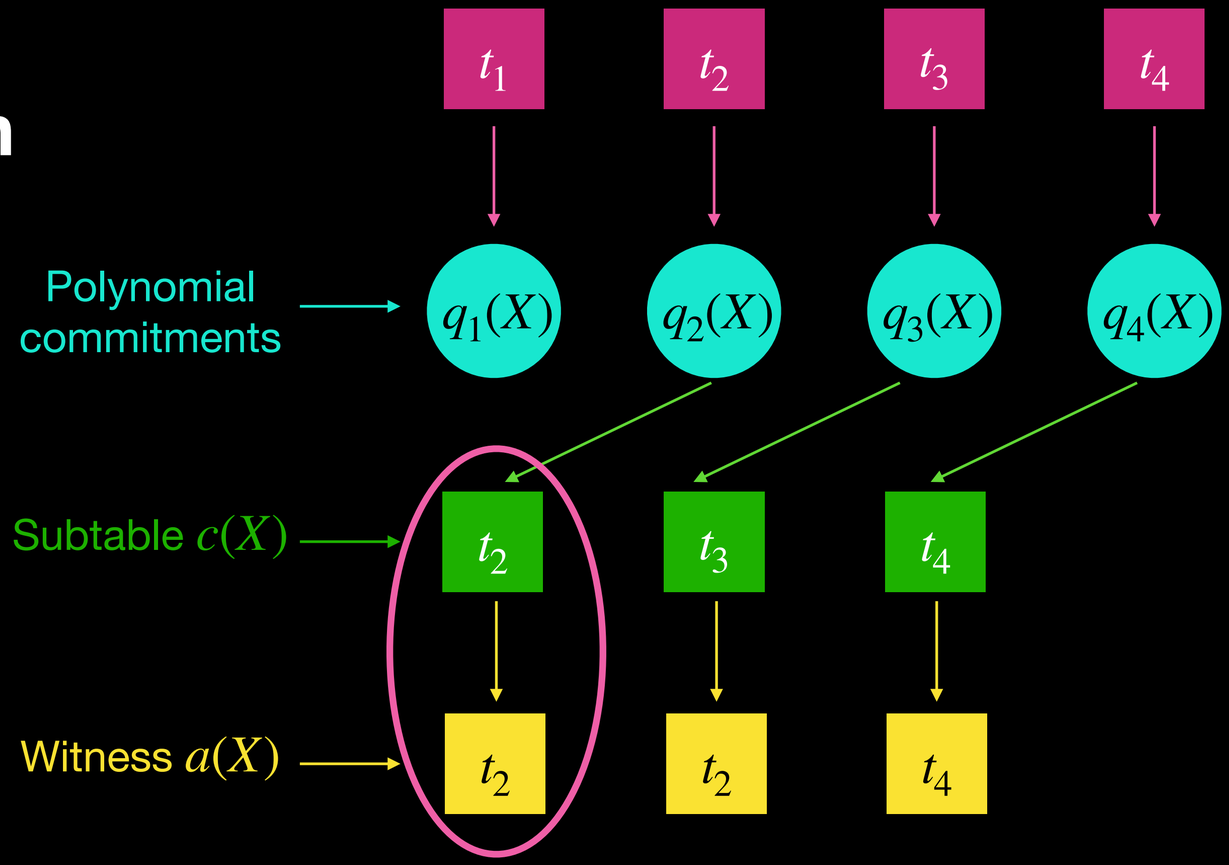
Prove lookup over hidden domain

$$\sum_{j \in I} e_{i,j} c(j) = a_i$$

Lincheck Argument over unit matrix

$$\mathbf{e}_i \in \{(1,0,0), (0,1,0), (0,0,1)\}$$

$$\mathbf{e}_1 = (1,0,0)$$



- Polynomial commitments $q_i(X)$ can be precomputed quickly.
- Subtable $c(X)$ contains entries in $t(X)$ over domain $z_I(X)$

The Baloo Strategy

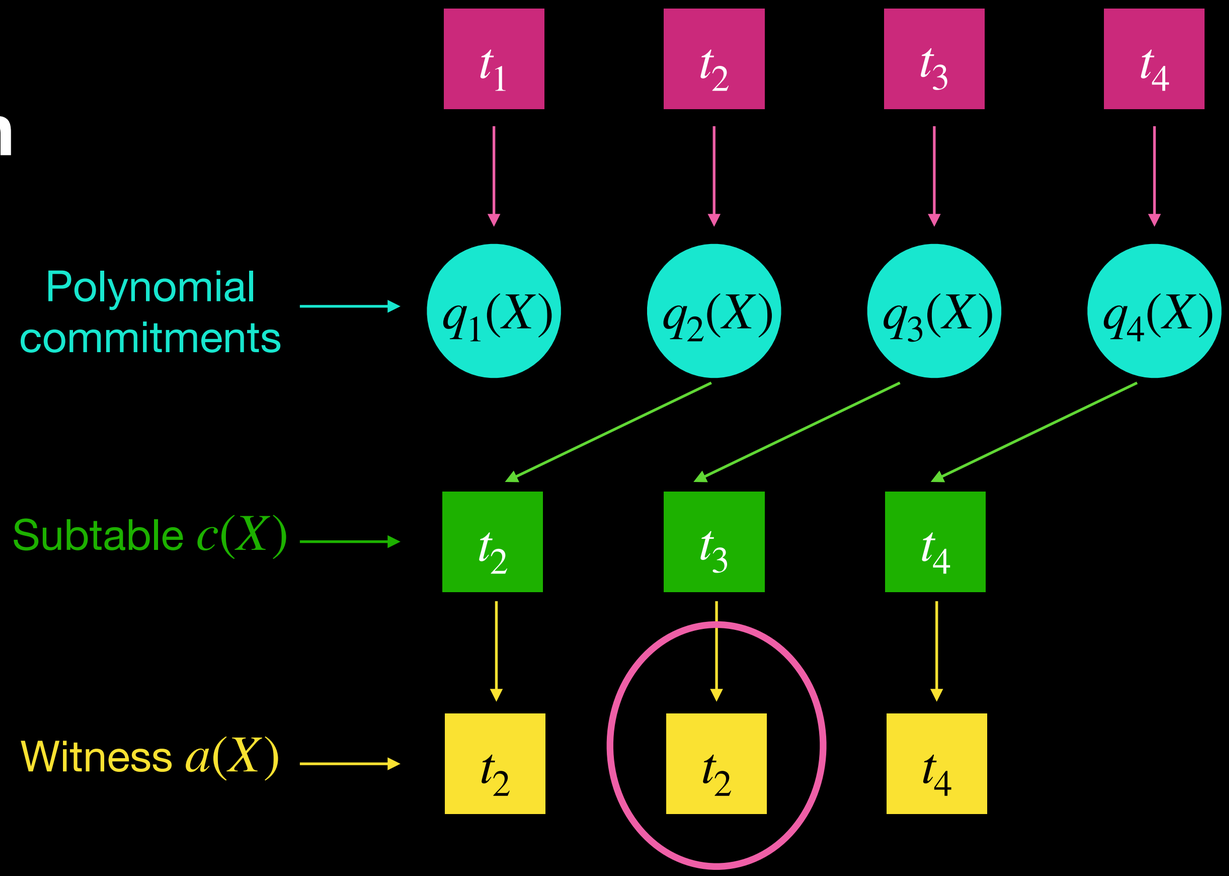
Prove lookup over hidden domain

$$\sum_{j \in I} e_{i,j} c(j) = a_i$$

Lincheck Argument over unit matrix

$$\mathbf{e}_i \in \{(1,0,0), (0,1,0), (0,0,1)\}$$

$$\mathbf{e}_1 = (1,0,0)$$
$$\mathbf{e}_2 = (1,0,0)$$



- Polynomial commitments $q_i(X)$ can be precomputed quickly.
- Subtable $c(X)$ contains entries in $t(X)$ over domain $z_I(X)$

The Baloo Strategy

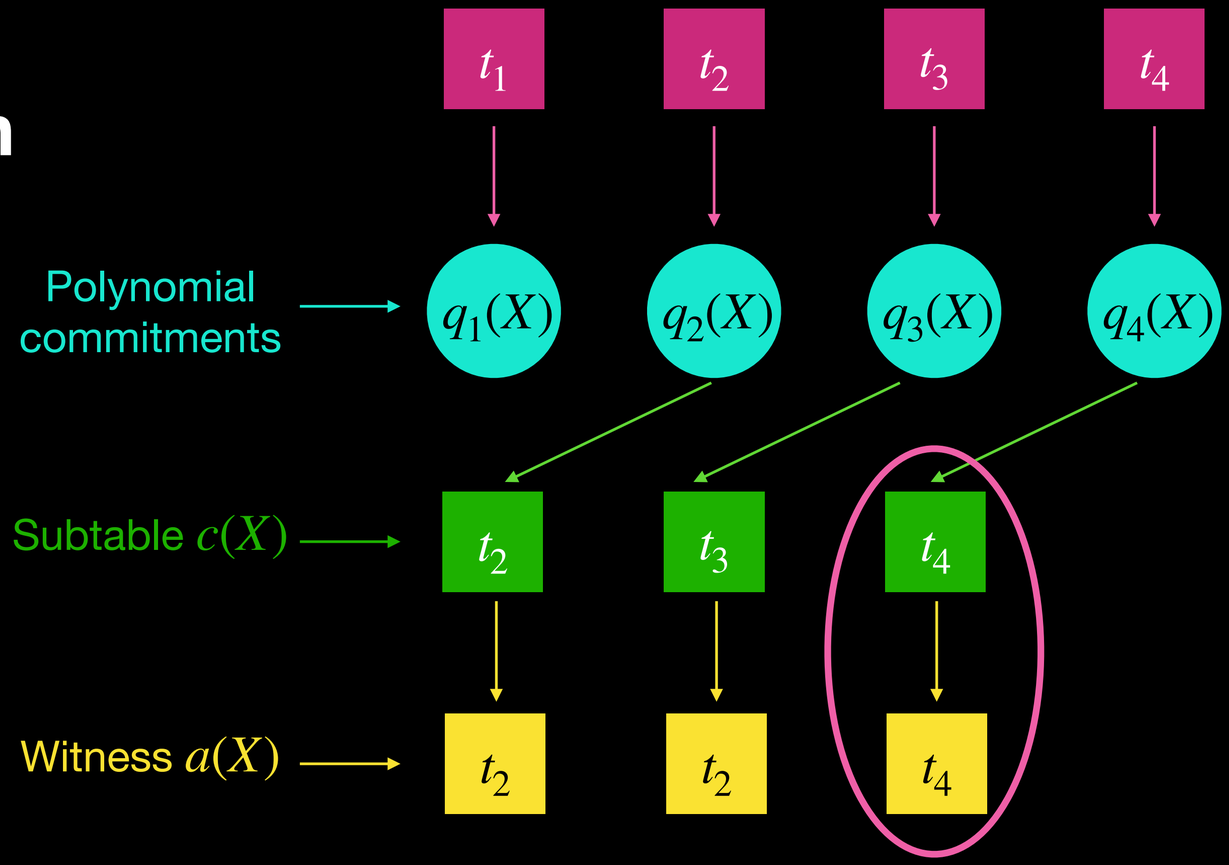
Prove lookup over hidden domain

$$\sum_{j \in I} e_{i,j} c(j) = a_i$$

Lincheck Argument over unit matrix

$$\mathbf{e}_i \in \{(1,0,0), (0,1,0), (0,0,1)\}$$

$$\mathbf{e}_1 = (1,0,0)$$
$$\mathbf{e}_2 = (1,0,0)$$
$$\mathbf{e}_3 = (0,0,1)$$



- Polynomial commitments $q_i(X)$ can be precomputed quickly.
- Subtable $c(X)$ contains entries in $t(X)$ over domain $z_I(X)$

The Baloo Strategy

Prove lookup over hidden domain

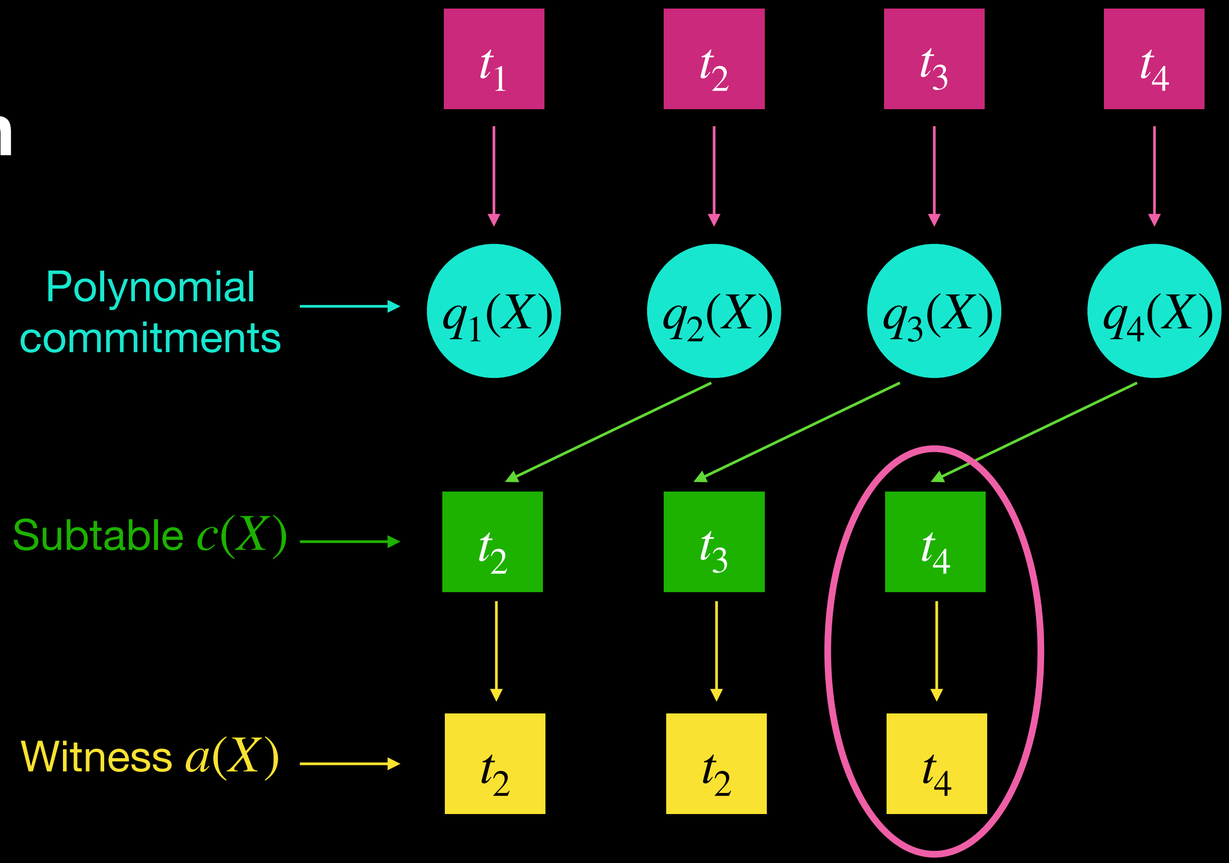
$$\sum_{j \in I} e_{i,j} c(j) = a_i$$

Lincheck Argument over unit matrix

$$\mathbf{e}_i \in \{(1,0,0), (0,1,0), (0,0,1)\}$$

$$\mathbf{e}_1 = (1,0,0)$$
$$\mathbf{e}_2 = (1,0,0)$$
$$\mathbf{e}_3 = (0,0,1)$$

Repeated units allowed



- Polynomial commitments $q_i(X)$ can be precomputed quickly.
- Subtable $c(X)$ contains entries in $t(X)$ over domain $z_I(X)$

The Baloo Strategy

Prove lookup over hidden domain

$$\sum_{j \in I} e_{i,j} c(j) = a_i$$

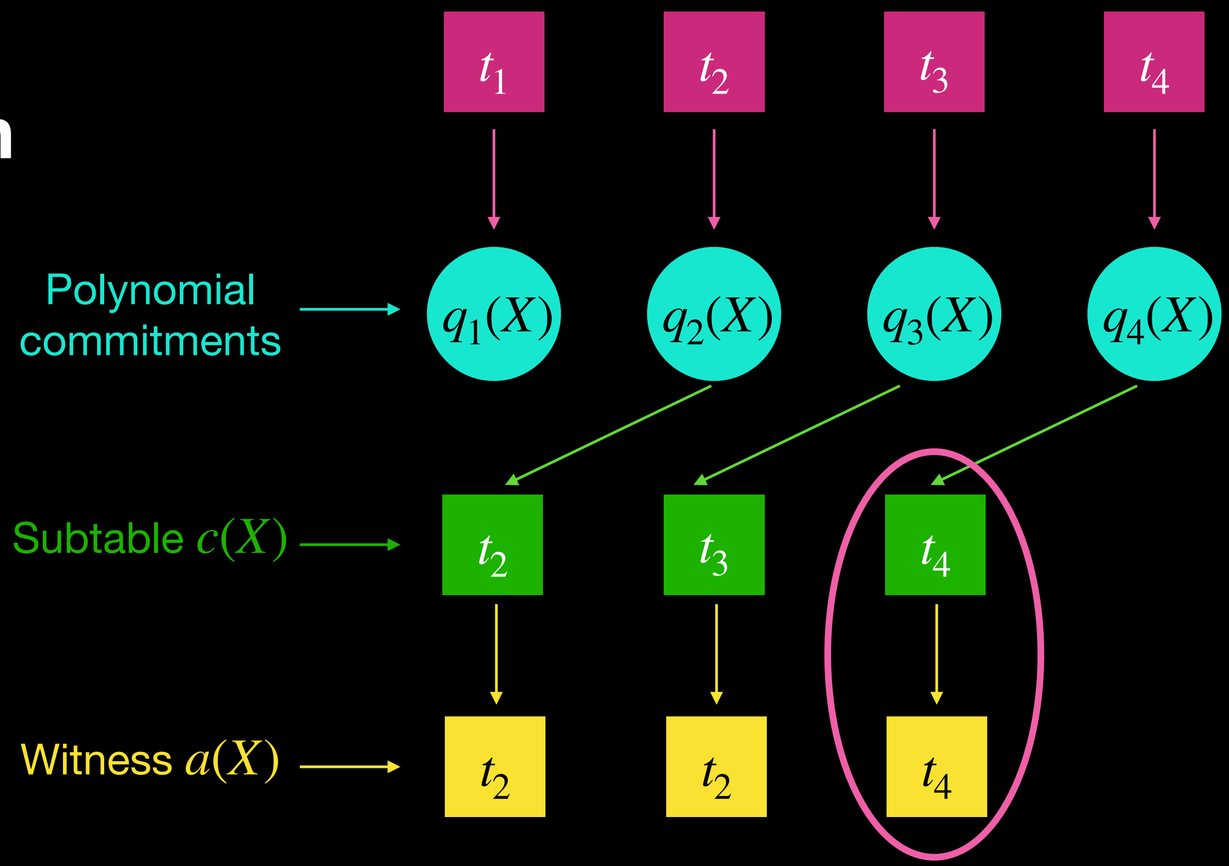
Lincheck Argument over unit matrix

$$\mathbf{e}_i \in \{(1,0,0), (0,1,0), (0,0,1)\}$$

$$\mathbf{e}_1 = (1,0,0)$$
$$\mathbf{e}_2 = (1,0,0)$$
$$\mathbf{e}_3 = (0,0,1)$$

Repeated units allowed

Lagrange polynomials are the unit vectors of the evaluation domain.



- Polynomial commitments $q_i(X)$ can be precomputed quickly.
- Subtable $c(X)$ contains entries in $t(X)$ over domain $z_I(X)$

The Baloo Strategy

Recall Lagrange Polynomials

$$L_i(X) = \begin{cases} 1 \text{ if } X = i \\ 0 \text{ if } X \neq i \end{cases}$$

$$L_i(X) = \frac{\prod_{j \in I, j \neq i}(X - j)}{\prod_{j \in I, j \neq i}(i - j)}$$

Lagrange polynomials are the unit vectors of the evaluation domain.

The Baloo Strategy

Recall Lagrange Polynomials

$$L_i(X) = \begin{cases} 1 \text{ if } X = i \\ 0 \text{ if } X \neq i \end{cases}$$

Lagrange polynomials are the unit vectors of the evaluation domain.

$$L_i(X) = \frac{\prod_{j \in I, j \neq i} (X - j)}{\prod_{j \in I, j \neq i} (i - j)}$$

If $\tau_i(X) = \prod_{j \in I, j \neq i} (X - j)$ then $L_i(X) = \frac{\tau_i(X)}{\tau_i(i)}$

The Baloo Strategy

Recall Lagrange Polynomials

$$L_i(X) = \begin{cases} 1 \text{ if } X = i \\ 0 \text{ if } X \neq i \end{cases}$$

Lagrange polynomials are the unit vectors of the evaluation domain.

$$L_i(X) = \frac{\prod_{j \in I, j \neq i}(X - j)}{\prod_{j \in I, j \neq i}(i - j)}$$

For $z_I(X)$ the vanishing polynomial over I

If $\tau_i(X) = \prod_{j \in I, j \neq i}(X - j) = \frac{z_I(X)}{X - i}$ then $L_i(X) = \frac{\tau_i(X)}{\tau_i(i)}$

The Baloo Strategy

Prove lookup over hidden domain

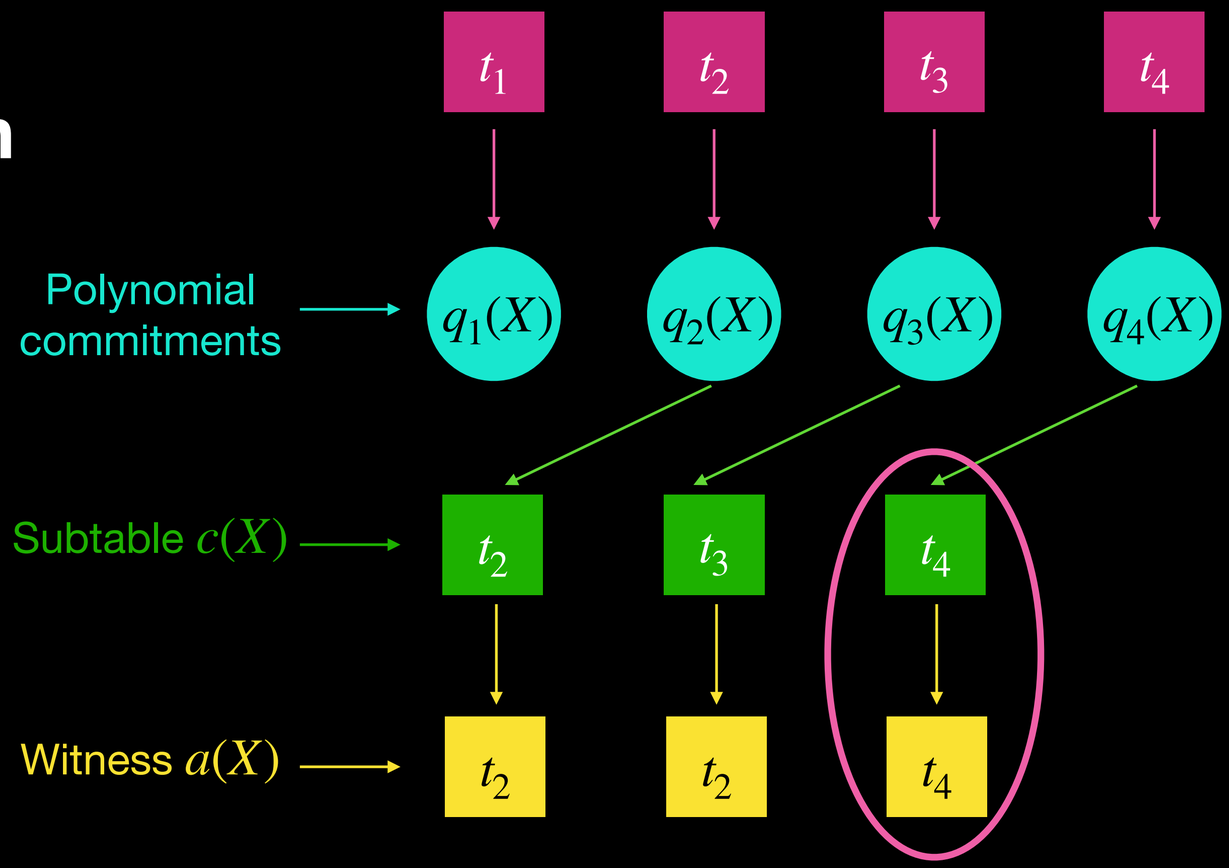
$$\sum_{j \in I} \frac{\tau_i(j)}{\tau_i(v_i)} c(j) = a_i$$

Lincheck Argument over unit matrix

$$\mathbf{e}_i \in \{(1,0,0), (0,1,0), (0,0,1)\}$$

$$\mathbf{e}_1 = (1,0,0)$$
$$\mathbf{e}_2 = (1,0,0)$$
$$\mathbf{e}_3 = (0,0,1)$$

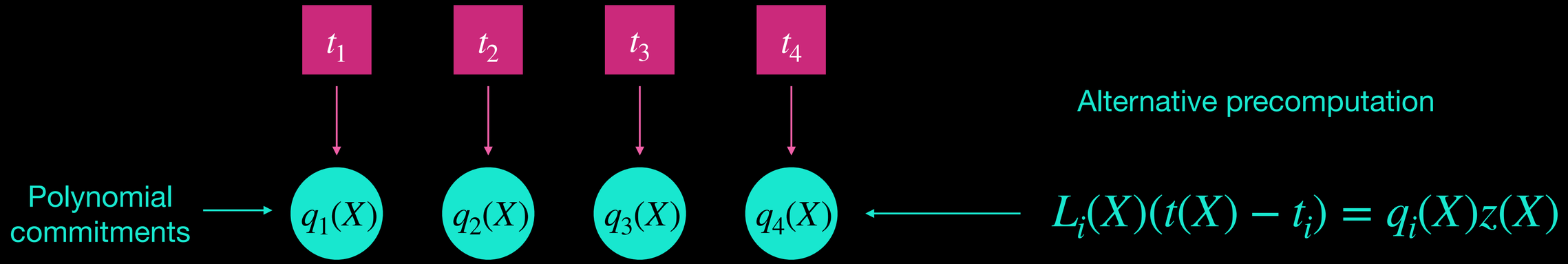
Unit vectors $e_i(X) = \frac{\tau_j(X)}{\tau_j(v)}$ and $\tau_j(X) = \frac{z_I(X)}{X - v}$ for some v



- Polynomial commitments $q_i(X)$ can be precomputed quickly.
- Subtable $c(X)$ contains entries in $t(X)$ over domain $z_I(X)$
- Define unit vectors with respect to $z_I(X)$

The CQ Strategy

Cached Quotients



Cached quotients for fast lookups

Liam Eagen
Blockstream

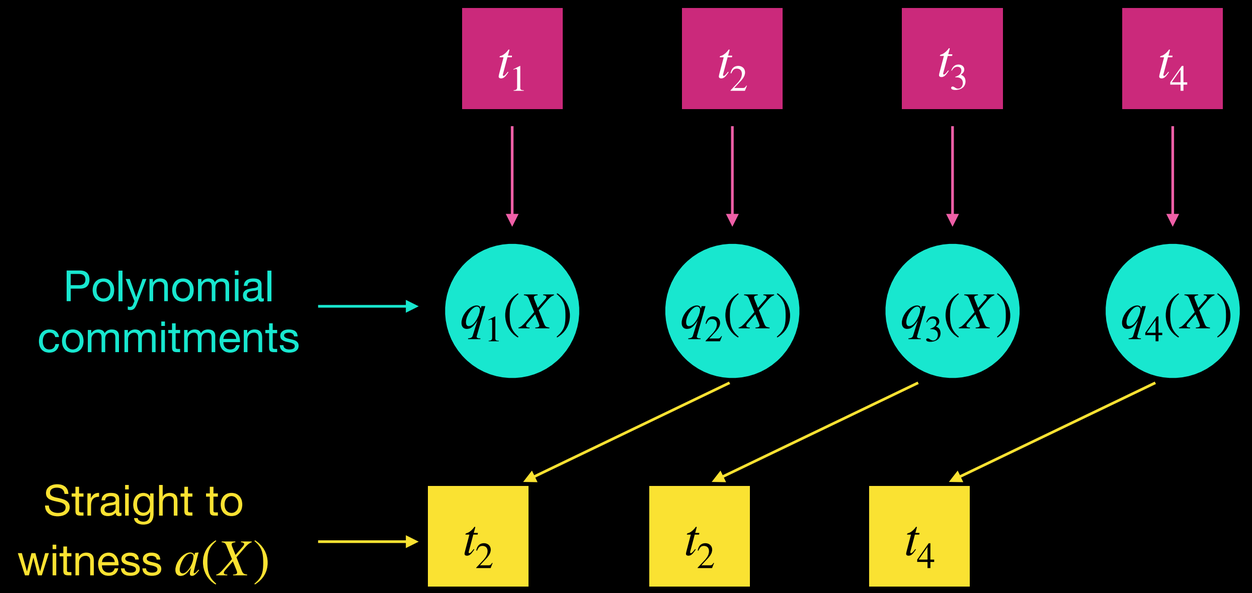
Dario Fiore
IMDEA software institute

Ariel Gabizon
Zeta Function Technologies

January 8, 2023

The CQ Strategy

Cached Quotients



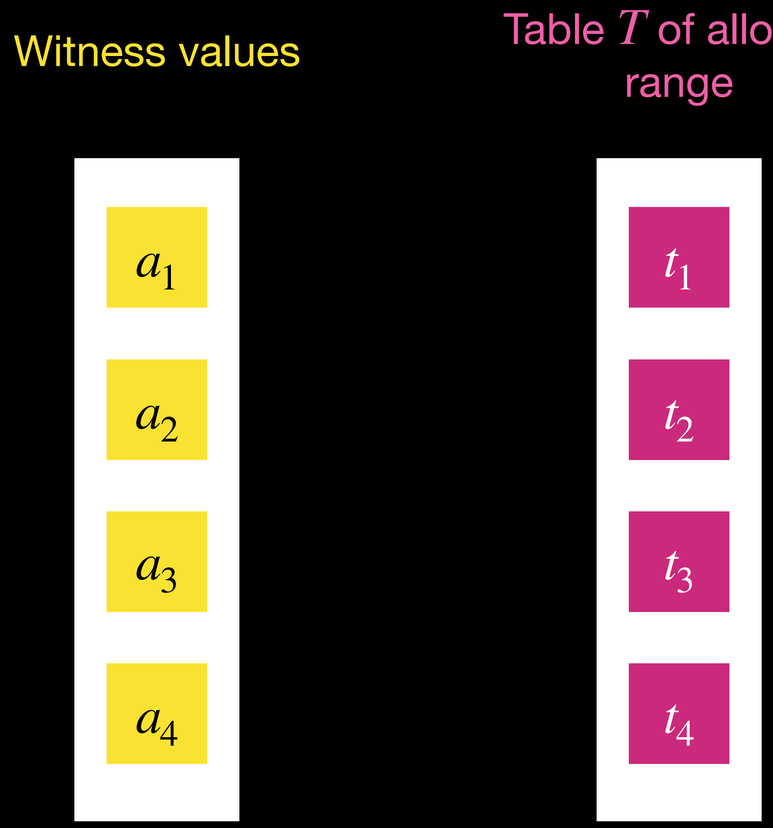
There exists $\mathbf{m}$ such that

$$\sum_{\text{all } t_j} \frac{m_j}{Y + t_j} = \sum_{\text{all } a_i} \frac{1}{Y + a_i}$$

Prover time
$O(m \log(m)),\ m = \deg(a(X))$ ***is independent from*** $t(X)$

Lookup Constraints in Plonkish

Function Tables



Lookup Constraints in Plonkish

Function Tables

$$a_1 \times b_1 \times c_1 - d_1 - e_1 = 0$$
$$a_2 + b_2 - c_2 = 0$$
$$a_3 \times b_3 \times c_3 - d_3 - e_3 = 0$$
$$a_4 + b_4 - c_4 = 0$$

$$a_1 = b_1 = c_1 = d_3 = a_4$$
$$x_1 = a_2 = d_1$$
$$y_1 = e_1$$
$$1 = b_2$$
$$a_3 = b_3 = c_3$$
$$e_3 = c_2$$
$$2 = b_4$$

u1	v1	w1	t1	r1	s1
u2	v2	w2	t2	r2	s2
u3	v3	w3	t3	r3	s3
u4	v4	w4	t4	r4	s4
u5	v5	w5	t5	r5	s5
u6	v6	w6	t6	r6	s6
u7	v7	w7	t7	r7	s7
u8	v8	w8	t8	r8	s8

output

Plonkish uses multiple columns

Lookup Constraints in Plonkish

Function Tables

Witness values

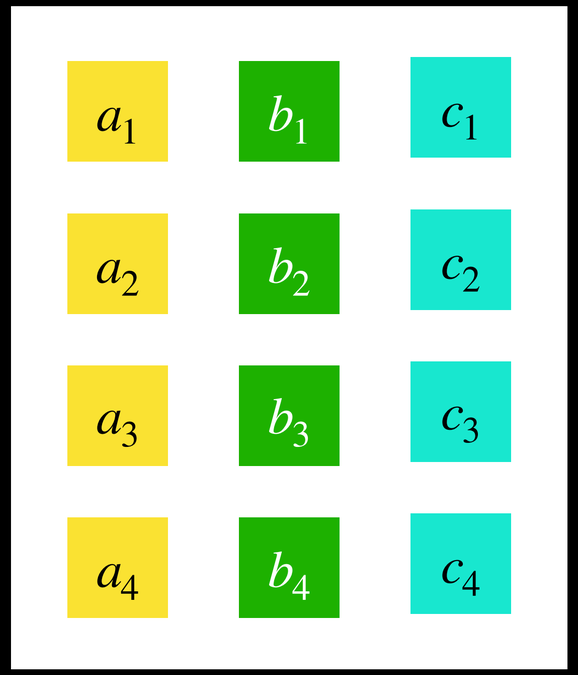
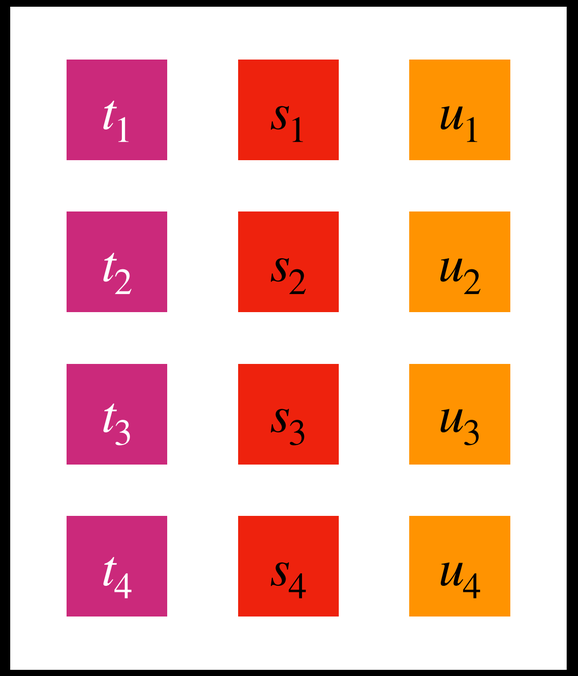


Table T of allowed range



Plonkish uses multiple columns

Lookup Constraints in Plonkish

Function Tables

Can lookup function tables e.g.

$$a_i \vee b_i = c_i$$

Witness values

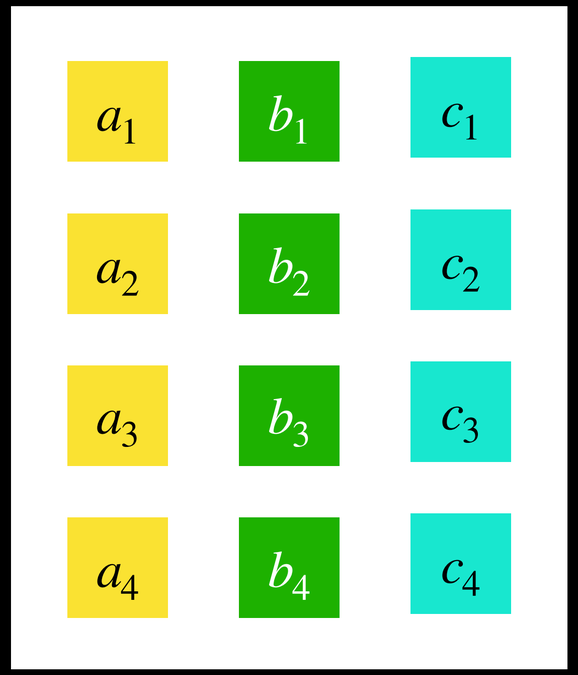
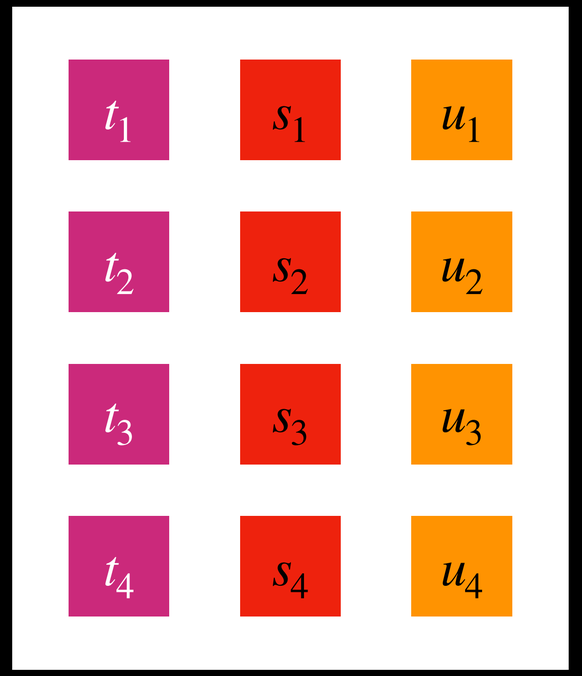


Table T of allowed range



Plonkish uses multiple columns

Lookup Constraints in Plonkish

Function Tables

Can lookup function tables e.g.

$$a_i \vee b_i = c_i$$

Witness values

a_1	b_1	c_1
a_2	b_2	c_2
a_3	b_3	c_3
a_4	b_4	c_4

All possible inputs and outputs.

Table T of allowed range

0	$\vee$	0	=	0
0	$\vee$	1	=	1
1	$\vee$	0	=	1
1	$\vee$	1	=	1

Plonkish uses multiple columns

Lookup Constraints in Plonkish

Function Tables

Can lookup function tables e.g.

$$a_i \vee b_i = c_i$$

Witness values

a_1	b_1	c_1
a_2	b_2	c_2
a_3	b_3	c_3
a_4	b_4	c_4

All possible inputs and outputs.

Table T of allowed range

0	$\vee$	0	=	0
0	$\vee$	1	=	1
1	$\vee$	0	=	1
1	$\vee$	1	=	1

Prove that $(a_i, b_i, c_i) \in (t_j, u_j, v_j)$

Lookup Constraints in Plonkish

Function Tables

Witness values

$$a_1 + b_1X + c_1X^2$$

$$a_2 + b_2X + c_2X^2$$

$$a_3 + b_3X + c_3X^2$$

$$a_4 + b_4X + c_4X^2$$

Table T of allowed range

$$t_1 + s_1X + u_1X^2$$

$$t_2 + s_2X + u_2X^2$$

$$t_3 + s_3X + u_3X^2$$

$$t_4 + s_4X + u_4X^2$$

- **Prove that** $(a_i, b_i, c_i) \in (t_j, u_j, v_j)$
- **Prove that** $a_i + b_iX + c_iX^2 \in (t_j + s_jX + u_jX^2)$

Lookup Constraints in Plonkish

Function Tables

Witness values

$$a_1 + b_1X + c_1X^2$$

$$a_2 + b_2X + c_2X^2$$

$$a_3 + b_3X + c_3X^2$$

$$a_4 + b_4X + c_4X^2$$

Table T of allowed range

$$t_1 + s_1X + u_1X^2$$

$$t_2 + s_2X + u_2X^2$$

$$t_3 + s_3X + u_3X^2$$

$$t_4 + s_4X + u_4X^2$$

- **Prove that** $(a_i, b_i, c_i) \in (t_j, u_j, v_j)$
- **Prove that** $a_i + b_iX + c_iX^2 \in (t_j + s_jX + u_jX^2)$

Random linear combination

Because random oracles exist..

Final Thoughts

- All sublinear time lookup arguments use pairings.
- Pairings are not post-quantum secure and require special, non-standardised, elliptic curves.
- Open question: can we design a practical, sublinear time lookup argument (in the size of the table) without pairings?

Final Thoughts

- All sublinear time lookup arguments use pairings.
- Pairings are not post-quantum secure and require special, non-standardised, elliptic curves.
- Open question: can we design a practical, sublinear time lookup argument (in the size of the table) without pairings?

Thanks for listening!