Lookup Arguments: Improvements, Extensions and Applications to Zero-Knowledge Decision Trees

Matteo Campanelli[1] **A. Faonio**[2] Dario Fiore[3] Tianyu Li[4] Helger Lipmaa[5]

Protocol Labs (now at Matter Labs)

EURECOM
Sophia Antipolis

IMDEA Software Institute

Delft University of Technology

University of Tartu

Proof Systems that are:

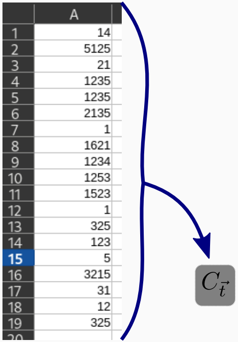
1. **Non-Interactive** single message from $\mathcal{P}$ to $\mathcal{V}$
2. **Argument of Knowledge** $\forall$ PPT $\mathcal{P}$: $\exists\ \mathcal{E} \rightarrow w$
3. **Succinct.** $|\pi| << |w|$
4. **Zero-Knowledge.**

Proof Systems that are:

1. **Non-Interactive** single message from $\mathcal{P}$ to $\mathcal{V}$
2. **Argument of Knowledge** $\forall$ PPT $\mathcal{P}$: $\exists\, \mathcal{E} \rightarrow w$
3. **Succinct.** $|\pi| << |w|$
4. **Zero-Knowledge.**

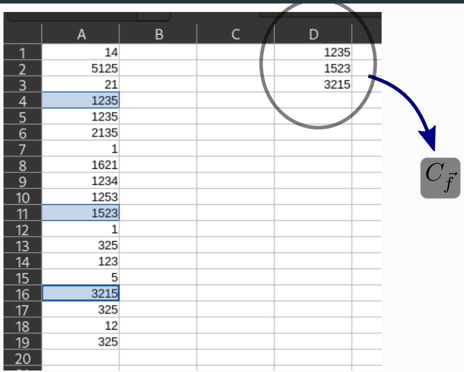
(1),(2),(3) without (4) is already cool, but with (4) is awesome.

Vector Commitment and Lookup Argument



	A
1	14
2	5125
3	21
4	1235
5	1235
6	2135
7	1
8	1621
9	1234
10	1253
11	1523
12	1
13	325
14	123
15	5
16	3215
17	31
18	12
19	325

- Binding
- Succicntness
- Hiding (or Not-Hiding),



	A	B	C	D
1	14			1235
2	5125			1523
3	21			3215
4	1235			
5	1235			
6	2135			
7	1			
8	1621			
9	1234			
10	1253			
11	1523			
12	1			
13	325			
14	123			
15	5			
16	3215			
17	325			
18	12			
19	325			
20				

Prove that col D (comm. as C_f) **subvector** of col A (committed as C_T)

State-of-Art:

- CQ **[EFG'22]** based on Cached Quotients (NEWS: Tue, 2nd Session, Track 2) $\Leftarrow$

  (Eagen, Gabizon and Fiore)
- Lasso **[STW'23]** for *"structured tables"*.

  (Setty, Thaler and Wahby)

State-of-Art:

- CQ **[EFG'22]** based on Cached Quotients (NEWS: Tue, 2nd Session, Track 2) $\Leftarrow$
  (Eagen, Gabizon and Fiore)
- Lasso **[STW'23]** for *"structured tables"*.
  (Setty, Thaler and Wahby)

Some Facts:

- Prove $\vec{f}$ commit'd as C_f is sub-vector of $\vec{t}$ commit'd as C_t.
- Proving Time **independent of** $|\vec{t}| = N$ after pre-computation"
  (we assume $|\vec{t}| >> |\vec{f}|$)
- Since we need pre-computation, we assume $\vec{t}$ is fixed.
- Based on KZG where $com_{\vec{t}} = g^{T(s)}$ and T poly interpolating values.

Our Contributions

- **Improve over CQ** along three directions:
    - Efficiency
    - Zero-Knowledge and *Fully* Zero-Knowledge.
    - Flexibility
- **Extend the notion** of Lookup Argument from vectors to matrices.
- **Application to Privacy-Preserving Machine Learning:** Zero-Knowledge Decision-Tree Statistics

Improve over CQ

Haböck's Logaritmic Derivatives Lemma and CQ

$\vec{f}$ subvector of $\vec{t}$ iff $\exists$ sparse $\vec{m} \in \mathbb{N}^N$

$$\sum_{i=1}^{N} \frac{m_i}{t_i + X} = \sum_{i=1}^{n} \frac{1}{f_i + X}$$

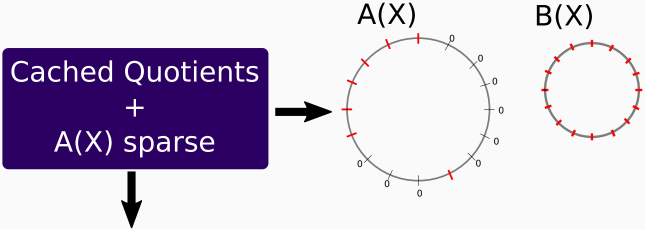
- $A(X)$ interpolates $\frac{m_i}{t_i+\beta}$, $B(X)$ interpolates $\frac{1}{f_i+\beta}$ (random β)

Haböck's Logaritmic Derivatives Lemma and CQ

$\vec{f}$ subvector of $\vec{t}$ iff $\exists$ sparse $\vec{m} \in \mathbb{N}^N$

$$\sum_{i=1}^{N} \frac{m_i}{t_i + X} = \sum_{i=1}^{n} \frac{1}{f_i + X}$$

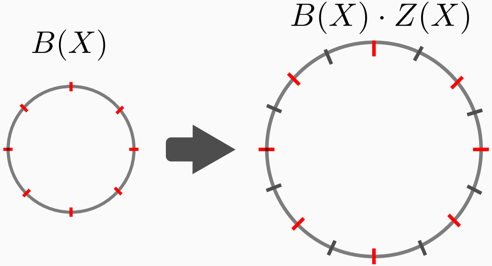
- $A(X)$ interpolates $\frac{m_i}{t_i+\beta}$, $B(X)$ interpolates $\frac{1}{f_i+\beta}$ (random β)



- 2 Sum-Checks Protocols to prove $\sum A(\omega_N^i) = \sum B(\omega_n^j)$.

$\{\mathsf{CQ}^+, \mathsf{zkCQ}^+\}$: From two sum-checks to one

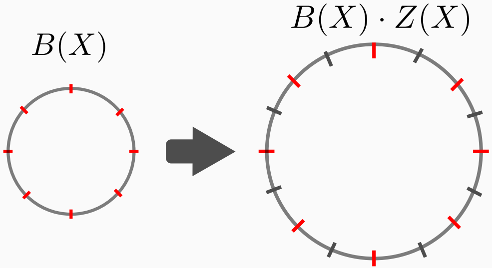
- If the (interpolation) subgroups $\langle \omega_n \rangle \subset \langle \omega_N \rangle$ then there exists $Z(X)$:



$$\sum_{i \in [n]} B(\omega_n^i) = \sum_{i \in [N]} B(\omega_N^i) \cdot Z(\omega_N^i)$$

$\{\mathsf{CQ}^+, \mathsf{zkCQ}^+\}$: From two sum-checks to one

- If the (interpolation) subgroups $\langle \omega_n \rangle \subset \langle \omega_N \rangle$ then there exists $Z(X)$:

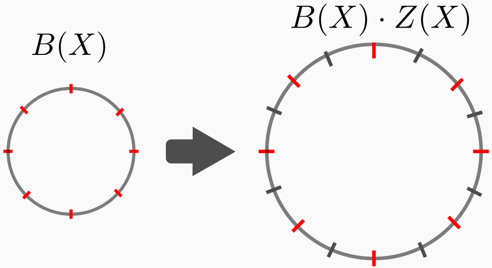


$$\sum_{i \in [n]} B(\omega_n^i) = \sum_{i \in [N]} B(\omega_N^i) \cdot Z(\omega_N^i)$$

- Now, we can batch Sum-Checks together!

$\{\mathrm{CQ}^+, \mathrm{zkCQ}^+\}$: From two sum-checks to one

- If the (interpolation) subgroups $\langle \omega_n \rangle \subset \langle \omega_N \rangle$ then there exists $Z(X)$:

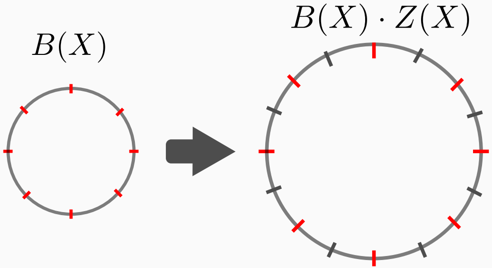


$$\sum_{i \in [n]} B(\omega_n^i) = \sum_{i \in [N]} B(\omega_N^i) \cdot Z(\omega_N^i)$$

- Now, we can batch Sum-Checks together!
- Zero-Knowledge: using ZK-SumCheck from Lunar [CFFHQ19].
- Fully ZK: privacy for both (big) table and (sub) vector.

$\{\mathsf{CQ}^+, \mathsf{zkCQ}^+\}$: From two sum-checks to one

- If the (interpolation) subgroups $\langle \omega_n \rangle \subset \langle \omega_N \rangle$ then there exists $Z(X)$:



$$\sum_{i \in [n]} B(\omega_n^i) = \sum_{i \in [N]} B(\omega_N^i) \cdot Z(\omega_N^i)$$

- Now, we can batch Sum-Checks together!
- Zero-Knowledge: using ZK-SumCheck from Lunar [CFFHQ19].
- Fully ZK: privacy for both (big) table and (sub) vector.
- Shorter proofs: $\{\mathsf{CQ}^{++}, \mathsf{zkCQ}^{++}\}$ using tricks from [LSZ22] (Lipmaa, Siim, Zajac)

Matrix Lookup Arguments

Matrix Commitment and Matrix Lookup

	A	B	C	D	E
1	14	42	84	133	45
2	423	465	507	45	549
3	21	63	105	46	147
4	122	164	206	47	248
5	133	175	217	48	259
6	244	286	328	49	370
7	1	43	85	50	127
8	1621	1663	1705	51	1747
9	1234	1276	1318	52	1360
10	1253	1295	1337	53	1379
11	1523	1565	1607	54	1649
12	1	43	85	55	127
13	325	367	409	56	451
14	123	165	207	57	249
15	5	47	89	58	131
16	3215	3257	3299	59	3341
17	325	367	409	60	451
18	12	54	96	61	138
19	325	367	409	62	451
20					
21					

$C_{\vec{t}}$

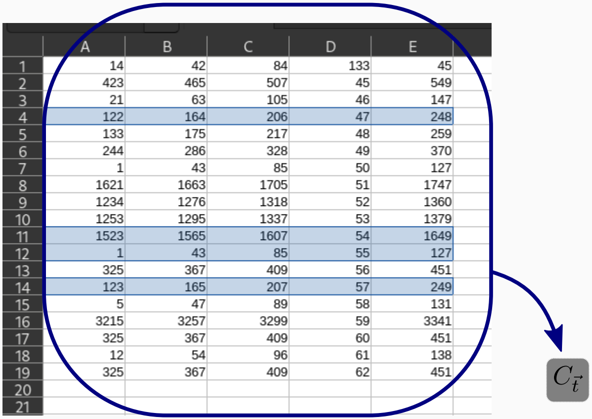
Matrix Commitment and Matrix Lookup

	A	B	C	D	E
1	14	42	84	133	45
2	423	465	507	45	549
3	21	63	105	46	147
4	122	164	206	47	248
5	133	175	217	48	259
6	244	286	328	49	370
7	1	43	85	50	127
8	1621	1663	1705	51	1747
9	1234	1276	1318	52	1360
10	1253	1295	1337	53	1379
11	1523	1565	1607	54	1649
12	1	43	85	55	127
13	325	367	409	56	451
14	123	165	207	57	249
15	5	47	89	58	131
16	3215	3257	3299	59	3341
17	325	367	409	60	451
18	12	54	96	61	138
19	325	367	409	62	451
20					
21					

$C_{\vec{t}}$

- A sub-matrix as rows PROJECTION [We also cover row + column]

Matrix Commitment and Matrix Lookup



	A	B	C	D	E
1	14	42	84	133	45
2	423	465	507	45	549
3	21	63	105	46	147
4	122	164	206	47	248
5	133	175	217	48	259
6	244	286	328	49	370
7	1	43	85	50	127
8	1621	1663	1705	51	1747
9	1234	1276	1318	52	1360
10	1253	1295	1337	53	1379
11	1523	1565	1607	54	1649
12	1	43	85	55	127
13	325	367	409	56	451
14	123	165	207	57	249
15	5	47	89	58	131
16	3215	3257	3299	59	3341
17	325	367	409	60	451
18	12	54	96	61	138
19	325	367	409	62	451
20					
21					

- A sub-matrix is a rows PROJECTION [We also cover row + column]

Our Matrix Lookup

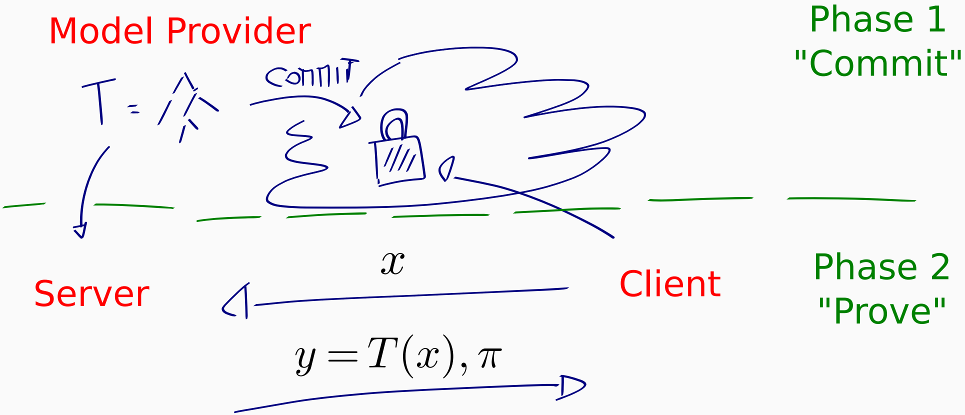
	A	B	C	D	E
1	14	42	84	133	45
2	423	465	507	45	549
3	21	63	105	46	147
4	122	164	206	47	248
5	133	175	217	48	259
6	244	286	328	49	370
7	1	43	85	50	127
8	1621	1663	1705	51	1747
9	1234	1276	1318	52	1360
10	1253	1295	1337	53	1379
11	1523	1565	1607	54	1649
12	1	43	85	55	127
13	325	367	409	56	451
14	123	165	207	57	249
15	5	47	89	58	131
16	3215	3257	3299	59	3341
17	325	367	409	60	451
18	12	54	96	61	138
19	325	367	409	62	451
20					
21					

$C_{\vec{t}}$

- Matrix Commit $\vec{F}$ = Vectorize $\vec{F} \to \vec{f}$ + Vector commit.
- Generic compiler from any homomorphic Vector Commitment (read it as KZG)
- Matrix Lookup for table with few columns is easy.
- Prove that $\exists \vec{r}, \vec{c}$: such that: (1) $(\vec{r}, \vec{c}, \vec{f})$ sub-vector of $(i, j, t_{i,j})_{i,j}$ and (2) tensor structures, $\vec{r} = \vec{r}' \otimes \vec{1}, \quad \vec{c} = \vec{1} \otimes (A, B, \dots, E)$.

Zero-Knowledge Decision Tree

The Model (Simplified)

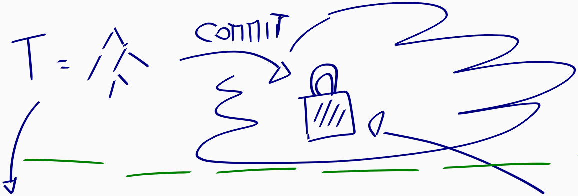


The Model: ZK Decision Tree "Statistics"

Model Provider

Phase 1
"Commit"

$T =$



Server

$(x_i)_{i \in [n]}$

Client

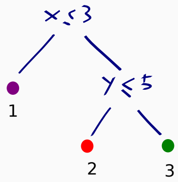
Phase 2
"Prove"

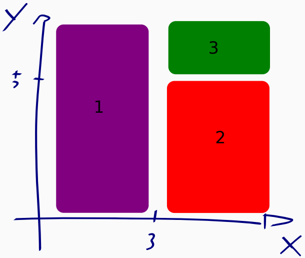
$f(T(x_i)_i), \pi$

Efficiency: $\Theta(|f| + nd)$

"Universal"

Standard Encoding



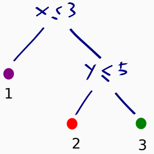


Box Econding

1 | (0,0),(3,B) |
2 | (3,0),(B,5) |
3 | (3,5),(B,B) |

Commit Phase

Standard Encoding



Box Econding

1 | (0,0),(3,B) | ●
2 | (3,0),(B,5) | ●
3 | (3,5),(B,B) | ●

Matrix Commit

$\Rightarrow C_T$

Prove Phase

T((4,2))= ●

1 | (0,0),(3,B) | ●
2 | (3,0),(B,5) | ●
3 | (3,5),(B,B) | ●

$\Rightarrow C_M$

$\Pi =$ Ct commit to T
Cm commits to M
and M submatrix of T

$\Pi =$ (4,2) lies inside the box defined by M
and the last column of M is ●

A Simple Attack and Our Fix

Box Econding

Matrix Commit

1 | (0,0),(3,B) | ●

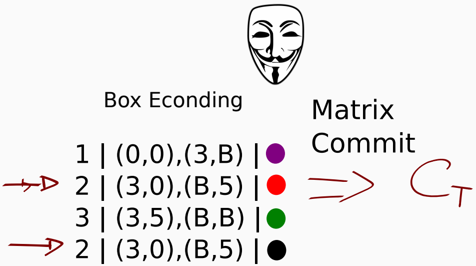
2 | (3,0),(B,5) | ● $\implies C_T$

3 | (3,5),(B,B) | ●

2 | (3,0),(B,5) | ●

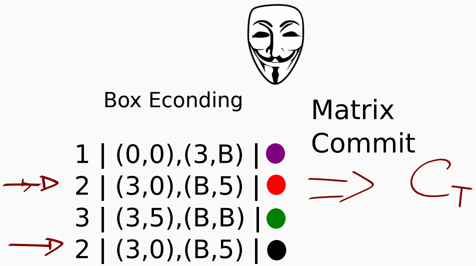
- The attacker can claim $T((3,2)) =$ ● and $T((3,2)) =$ ●

A Simple Attack and Our Fix



- The attacker can claim $T((3,2)) = \textcolor{red}{\bullet}$ and $T((3,2)) = \bullet$
- Fix: Prove that C_T commits to a valid *box encoding*

A Simple Attack and Our Fix



- The attacker can claim $T((3,2)) = \bullet$ and $T((3,2)) = \bullet$
- Fix: Prove that C_T commits to a valid *box encoding*
- We give algebraic constraints (read it linear/hadamard constraints) for validity

A Simple Attack and Our Fix

Box Econding

Matrix Commit

1 | (0,0),(3,B) | ●
2 | (3,0),(B,5) | ●
3 | (3,5),(B,B) | ●
2 | (3,0),(B,5) | ●

$\Rightarrow$ C_T

- The attacker can claim $T((3,2)) =$ ● and $T((3,2)) =$ ●
- Fix: Prove that C_T commits to a valid *box encoding*
- We give algebraic constraints (read it linear/hadamard constraints) for validity
- Using technique from **[ZGKMR22]** we get $\Theta(N)$ proving time.

  (Zapico et al)

- New Lookup Argument with Fully ZK
- Generic compiler to Matrix Lookup
- zkSNARKs for decision tree inference and statistics

https://ia.cr/2023/1518


Mandaang guwu!