24.11.05 Merkle Tree with Selector

Xun Zhang Wuyun Siqin Bingsheng Zhang
Zhejiang University, CHN
22221024@zju.edu.cn 3210101763@zju.edu.cn bingsheng@zju.edu.cn

November 5 2024

1 Merkle Tree with Selector

In our previous implementation, there is no selector in the circuit. Which means that we "swap" the inputs fo Poseidon hash function manually.

In this version, the index of each public key has not been imported as a witness.

So we add the this part of constraint into the circuit:

- $\mathsf{left} = gate.select(\mathsf{hash}, \mathsf{path_i}, \mathsf{index_i})$
- $\mathsf{right} = gate.select(\mathsf{path_i}, \mathsf{hash}, \mathsf{index_i})$
- $\mathsf{inputs} = [\mathsf{left}, \mathsf{right}]$
- $\mathsf{hash} = poseidon.hash(\mathsf{inputs})$

And we need the index value to do the later implementation.

2 Benchmark

We also benchmark the new version of merkle tree.

Here is the benchmark results:

degree	num_aggregation	num_origin	proof_time	proof_size	verify_time
19	16	32	5.6890s	704	6.5756ms
19	32	64	7.6071s	1056	4.4714ms
19	64	128	13.4401s	2112	7.6402ms
19	128	256	23.2960s	3872	8.1395ms
19	256	512	48.6304s	8448	11.6570ms
19	512	1024	100.0534s	17600	13.6171ms
19	1024	2048	213.2567s	38016	20.3142ms

Table 1: Merkle Tree Path Benchmark with stake

The benchmark results shows that the merkle path with selector takes about 5% to 8% longer proving time than previous version.

3 Relations of Index

There is another relation that we have not discussed before, that is the unequal relationship between indexes.

- $\forall i : \mathsf{index}_i \leq m$ and $\forall i \neq j : \mathsf{index}_i \neq \mathsf{index}_j$.

It maybe be difficult to prove, because we need the boolean value of every path's indexes, but in the mithril, the index is a integer. We plan to prove it in following circuit:

- $\mathsf{index\_bigint} = gate.bit\_composition(\mathsf{index}[])$
- $0 = gate.equal(\mathsf{index\_bigint_i}, \mathsf{index\_bigint_j})$

Or to prove it as:

- $\mathsf{index}[] = gate.bit\_decomposition(\mathsf{index\_bigint})$
- $0 = gate.equal(\mathsf{index\_bigint_i}, \mathsf{index\_bigint_j})$

We will adjust the proving logic according to the engineering practice.

4 Discussion About $\phi(\mathsf{stake})$ In the Merkle Tree

We mentioned that we can optimize the merkle tree by replacing the stake by $\phi(\mathsf{stake})$.

However, ev is a natural in [0,2^{512}], while ϕ is a floating point in [0, 1]. It is hard to represent a float number in circuit. How to address this problem ?

This is the original code implementation in Mithril:

- $w = \mathsf{stake_i}/\mathsf{total\_stake}$
- $p < 1 - (1 - \phi_f)^w, with \quad p = ev/2^{512}$.
- let $q = 1/(1 - p) \quad and \quad c = ln(1 - \phi_f)$
- $q < exp(-w * c)$

And they use Taylor expansion to compare the two values, iterate 1000 times. Note that q and $exp(-w * c)$ are all float numbers.

We try to modify the comparison process into a integer version, still need discussion and investigation. Or we can just use a simple comparison:

- For $i \in \{1 \dots k\}$: $ev_i \leq \phi(\mathsf{stake}_i)$.

But it seems different from the original code implementation.