Access list incentives mean transactions should rarely throw an ACCESS_LIST_EXCEPTION

by accessing state outside of the access list. By having users provide two transaction signatures (only one of which is included onchain) we can save from having onchain access lists whenever an ACCESS_LIST_EXCEPTION

is not thrown. This reduces onchain transaction sizes, saves gas, and possibly improves privacy.

Construction

Let T

be an unsigned transaction (without the access list) and let A

be the corresponding unsigned access list. Currently a user sends T

, A

and sig([T, A])

to validators. We suggest instead that the user sends the following:

- Unsigned transaction

: T

- Unsigned access list

: A

- Optimistic signature

: sig(T)

- Exceptional signature

: sig([T, A])

Validators execute T

relative to A

. Two cases may arise:

1. Optimistic

: No ACCESS_LIST_EXCEPTION

is thrown, in which case only T

and sig(T)

must be included onchain.

1. Exceptional

: An ACCESS_LIST_EXCEPTION

is thrown, in which case T

, A

and sig([T, A])

must be included onchain.