Private Batch Swaps using Aztec and SUAVE

During Flashbot's DEX Day in London,@ferranbt (SUAVE Engineer), Reaal (Independent builder) and I (PM at Aztec) spec'd out how Private Batch Swaps could work with SUAVE and Aztec

Special thanks to Reaal and@dmarz for the review.

First let's explore how would Private Batch Swaps + Suave work on a generic public chain

[

image

716×800 46.3 KB

](https://collective.flashbots.net/uploads/default/original/2X/5/5453f41265353664ecb6a582f921601a7b026d25.jpeg)

1. User on a chain "X" creates an Intent containing: their address, the swap's input and output tokens, the input amount, slippage parameters (on or offchain based), and a secret key for claiming.
2. All intents go to a Gateway, a solidity smart contract on Suave which consolidates all user intents (across a sufficiently large constant time or variable volume-based window) into a batch. It auctions the batch to a group of solvers and selects the best one.
3. The Solver is a participant that takes a batch and makes HTTP requests and onchain calls to query various liquidity sources. Only liquidity sources on chain X can actually be used for the swap, whereas all other sources can only be queried.
4. The kettle is the SGX server where the gateway and all the solvers reside. Any user's inputs are encrypted with the key in the kettle and therefore the individual intents are hidden to everyone except the gateway in the kettle.
5. The winning solver provides transaction calldata and targets to settle the aggregate trades on chain X.
6. The gateway then builds a multicall transaction for the escrow contract that: transfers in the pre-approved input tokens from all users; performs the solver's swaps; and assigns hashes of the clients' secrets to each output token allocation.
7. After the gateway simulates and submits the transaction, users can claim their swap outputs from the escrow contract to new addresses using their secret keys.

What's private?

In the current system, the gateway can see the individual trades (and user identities), solvers and everyone else only see aggregated trades.

Users' approvals and transfers to the escrow are public, so input addresses + input amounts + input tokens are public.

The transaction on escrow contract is settled publicly. Individual output amounts are also public, but users can redeem their outputs privately.

The gateway also aggregates the input transfers and the output secret->token,amount mappings, this enables using multiple intents to split and combine swaps for increased privacy:

Users can split one input token/amount into multiple output tokens and/or amounts claimable with different secrets.

Users can combine multiple input addresses/tokens/amounts into one output token claimable with one secret.

The only potential for toxic MEV would be the transactions that would make use of AMMs on the chain X.

What does Aztec add?

Approvals at Aztec would reside in the solver rather than the escrow so they would be private. This would mean input token amounts are hidden from the public along with user's address.

The map of secret to output token is still public. As with the flow, the user would supply their secret in public to redeem their output amount and then "shield"/"claim" it into the private domain.

The redeem-claim flow is currently 2 transactions. This can be further optimized. In Aztec, private state exists as UTXOs, while public state exists in Ethereum-like account state model. While creating the intent, in private, the user can create a note with their output token but leave output amount blank (as they do not know what their output amount will be). When the solver executes the swap, it would emit the secret with the amount in public. User's wallet scans for such events and then adds the amount to your note.

Using Aztec would increase privacy (hide input intents, output address, make it easier to split/batch) and also optimize on the with a superior privacy set that the entire Aztec network would share, as opposed to public chains where each app would have to bootstrap their own privacy set.

Aztec Alpha Programme

If you have read on so far, you deserve some alpha!

Aztec is running an incentivized guided ecosystem programme to explore how DEXs would look on the Aztec network! The programme runs from April 7 - May 4th, has a prize pool of $15,000 and provided dedicated technical and marketing support for the builders.

- More info here.
- If you are keen to participate, there is a1-minute survey waiting for you

You will build on the Aztec Sandbox, which is a local development environment (similar to Anvil). Aztec is not EVM compatible and has a custom execution environment with its own rust-inspired smart contract language, built using Noir