Risk assessment draft: Aave v3 on Scroll

1. Introduction

Scroll is a Layer 2 ZK (Validity) Rollup that “gains its security and speed by executing transactions off-chain, and also producing a cryptographic proof that the transactions were executed correctly.” Scroll Mainnet was officially launched on October 17th, 2023.

2. Methodology

Boilerplate - unnecessary to include here

3. Evaluation

3.1. Oracle Infrastructure

At the moment, the Aave protocol uses 3 types of oracles: prices, sequencer uptime, and Proof-of-Reserve.

Aave considers that having Chainlink providing oracles, especially for the most important type (prices), is a must for any deployment, with alternatives only considered really ad-hoc, given the additional complexity added on evaluation/integration.

3.1.1. Price feeds

Used by the Aave protocol to price assets listed.

Scroll **HAS** Chainlink price feeds available for the major assets on the network.

There are additional price feed oracles on Scroll, but they are unlikely to be needed by Aave.

3.1.2. L2 Sequencer Uptime feed

“flag” parameter indicating in real-time to the Aave protocol if a sequencer of a rollup (or any network involving some centralization on sequencing of transactions) is properly running.

[Scroll team to confirm with BGD when risk assessment starts]

3.1.3. Proof-of-Reserve feeds

The utility of Proof-of-Reserve on zk-rollups still needs further research, in what concerns the “official” bridge of the network. Therefore not having them available at the moment doesn’t have any influence on this analysis.

3.2. Blockchain explorer

For Aave, same as for any other blockchain project, block explorers like Etherscan are a fundamental component, specifically for the following:

1. Verifiable smart contracts code visualization.
2. Read/write interface with smart contracts.
3. Basic data analysis tool for misc aspects like token holders.

The official blockchain explorer of the Scroll network can be found at https://scrollscan.com/, and it is a white-labeled instance of Etherscan.

3.3. Compatibility with Ethereum RPC standard

Basic compatibility with the Ethereum nodes RPC de-facto standard (eth_, *web3_*) is quite an important requirement for Aave or any other protocol, given that it helps to have tools built for Ethereum (or other similar networks) working out-of-the-box just by plugging them to node, of Scroll in this case.

Scroll **HAS** complete compatibility, with the main node being Scroll L2geth Node, an adaptation of go-ethereum. Depending on the node provider there can also be support for non-standard endpoints (e.g. trace_*) in the future.

3.4 Compatibility with Ethereum account format (addresses)

Scroll **IS** fully compatible with the Ethereum account format.

3.5. RPC public endpoints and providers

Basic and reliable public RPC infrastructure is a must for Aave, as it is the way to connect to the network, both for data reading and transaction submission.

Scroll has the following:

- An 'official' public RPC endpoint at **https://rpc.scroll.io/** (subject to rate limits)
- Public and private options located here (https://chainlist.org/chain/534352)

3.6. Custom behavior (lack of) of the execution layer

Whenever a network has custom/extended behavior with respect to Ethereum, it is important to be aware of it and evaluate if it has any impact on the Aave protocol.

Examples of this potential behavior are the presence of new pre-compiles (compared with Ethereum or similar rollups like Optimism), EVM opcodes, native account abstraction/meta-transactions, chainId definition out of the norm, etc.

Even if not doing a full evaluation of the zkEVM implementation, the following should be checked, given that historically have been critical aspects:

- The `chainId` behavior is appropriate, with the id `534352` for Scroll not clashing with any other.
- Scroll has equivalence (address and logic) with the `ecRecover` and `identity` Ethereum pre-compiles, which covers the needs of Aave.
- Scroll has equivalence with Ethereum pre-1559, to be upgraded to post-1559 in 2024 (although Scroll also has functionality from the EVM *after* EIP-1559, with full list of opcode differences here).

Additionally, the Scroll team confirms that we have not implemented any additional behavior in the EVM, and that the limited changes are isolated and do not affect the general execution model of the EVM.

3.7. Support of wallet providers

Wallet products like Metamask, Ledger, Coinbase Wallet, and others, are fundamental pieces of the infrastructure for users to access the Aave protocol. So it is a strong requirement for a network to be supported by a subset of them.

Given its EVM compatibility in the context of this document, Scroll is transparently supported by many of the most common chain-agnostic wallets, like Metamask, Ledger, Rabby, etc. and in the process of integration with others such as Frame.

It is possible that other types of wallets (e.g. based on smart contracts) don't support Scroll, but it is something expected in a young network and doesn't have any negative consequence from the infrastructure perspective.

3.8. On-chain multi-signature infrastructure

The permissions on the Aave ecosystem are directly held by on-chain governance smart contracts or scheduled to be like that once cross-chain governance infrastructure can be applied across all the networks.

However, different protection/emergency mechanisms, like the capability of canceling cross-chain governance proposals, or pausing an Aave asset/pool, depend on the Aave Guardian, who is capable of acting faster than the governance process.

Consequently, having on-chain multi-signature contracts is a requisite to have Aave on a different network, with a high preference for industry-standard tools like Gnosis Safe.

Scroll **HAS** an instance of the Gnosis Safe contracts on-chain, but the user interface and server infrastructure are not the official Safe, but a fork on https://safe.scroll.xyz/welcome.

The Scroll team has confirmed the maintainer team (Protofire) is in close contact with the Safe team, and the upstream is fully aligned.

Additionally, an official Safe instance should be live in early 2024.

3.9. Transactions simulation infrastructure (fork)

Lately, a really important development experience component is the ability to execute test transactions (simulations) on forked production networks.

A good part of the tooling around Aave depends on simulations by using different libraries/frameworks like Hardhat, Foundry, or Tenderly. This way, it is possible to rapidly prototype new developments, get extra assurances on governance proposals and protocol upgrades, change risk parameters, etc.

It is possible to do fork simulations on Scroll with Foundry and Hardhat.

Currently, Tenderly is not integrating Scroll, but the Scroll team is working on a potential integration.

3.10. Chain data/indexing solutions

For different projects and entities integrating Aave, and even if not a blocker for deployment, it is important that solutions like TheGraph or Dune are operating on the candidate network, to avoid building from scratch data pipelines.

Scroll is supported on TheGraph.

Scroll is not supported by Dune.

3.11. Bridging infrastructure: assets, messages

Given the central role of Ethereum in the DeFi and Aave ecosystems, proper bridging infrastructure to/from is a must for any candidate network.

There are 2 types of bridging affecting Aave: assets and generic messaging. In the case of Scroll, the state of these 2 types is as follows:

- **Assets.** sub-use case the generic messaging infrastructure supported by Scroll. For end users, it is possible to bridge assets via https://scroll.io/bridge.

  Regarding the security of the ERC20 smart contracts for bridged assets, we have checked the main tokens share the same implementation: a simple ERC20 based on the OZ version, with burn/mint capabilities by an entity defined as “bridge”.
- **Generic messaging.** Scroll supports bi-directional generic message passing.

In addition to Scroll’s default bridging mechanism, there are other providers available, but this is not so important for Aave at the moment, as a.DI will use mainly the canonical one.

3.12. Commitment in security-incidents

Having proper mechanisms and procedures to prevent and react to security incidents is something quite fundamental for any platform and application, and networks like Scroll are no exception:

- On the prevention side, Scroll has an Immunefi bug bounty campaign running.
- Scroll has a sizeable in-house team of security experts.
- Scroll collaborates with external thread monitoring platforms such as Chainalysis in aspects like bridge security.

- If any incident happens, Scroll has confirms immediately react and execute at least the following measures:
    - Act as fast as possible to protect against damage.
    - Contact the technical side of Aave.
    - Engage independent security experts to assess the security problem and the reaction to it.
    - Properly community the Aave community about the incident, and the next steps.
- A private channel of communication will be kept between the Scroll team and the assigned technical team of the Aave community (e.g. BGD), for any necessary update concerning the network and consequently, Aave on Scroll.

3.13. Network security/technical model

At the core of any candidate network analysis are its morphology (which type of network it is) and security/operational models (how it works and which parties are involved in the control over the network; decentralization degree).

Regarding its morphology/type, Scroll is a zk/validity roll-up, meaning that it leverages zk-knowledge technology (specifically, zk-SNARKs) to provide high scalability for blockchain transactions.

Regarding its security/operational model, there are multiple aspects to analyze.

3.13.1. Transactions flow

A detailed explanation can be found HERE, but to summarize:

1. A user interacting directly with Scroll builds a transaction and submits it to the JSON RPC API endpoint of a network node (referred to as the execution node, and architecturally part of the L2 sequencer).
2. The sequencer validates the transaction and stores it in its local transaction pool. *Additionally, users will be also able to submit transactions to the Scroll bridge contract on Ethereum (forced batches), which will introduced in the Pool. This mechanism is not active yet.*
3. The Scroll sequencer periodically starts a new mining job. It pulls the L1 messages from the `L1MessageQueue` contract and transactions in the L2 mempool and seals a

block. Once a transaction is included in a L2 block, its status becomes `Confirmed`.

4. The rollup node collects new L2 blocks and packs them into chunks and batches (see more details in **Transaction Batching**). Periodically it sends a *Commit Transaction* that posts the data of a batch of transactions to the L1 `ScrollChain` contract. After the Commit Transaction is finalized in a L1 block, the status of the transactions in this batch becomes `Committed`. At this time, users can reconstruct L2 state themselves completely based on the committed data from the L1 contract.
5. After the validity proof is generated, the rollup node sends a *Finalize Transaction* including the validity proof and the new state root after this batch for onchain verification. Once the Finalize Transaction succeeds and confirmed in a L1 block, the status of the L2 transactions in this batch becomes `Finalized`. The new state root can be used by third parties trustlessly.3

3.13.2. Data availability

On Scroll, all the transaction data is submitted to Ethereum, so data availability boils down to Ethereum, which can be considered the highest standard at the moment.

3.13.3. Upgradeability and control model

The main centralization point of the network at the moment is the Sequencer component, in charge of gathering and sending to proof the transactions to be included in the rollup.

It could be possible to skip the Sequencer by using so-called forced transactions, but currently, this mechanism is not active.

For self-verification of smart contract roles, the major Scroll contract addresses can be found HERE.

3.13.4. Security audits

The list of Scroll security audits of can be found HERE.

3.13.5. Network upgrade procedures

Currently there is a `ScrollOwner` contract that allows a fine-grained control of the Scroll contracts. Under normal situations, it is configured to mandate a 14-day delay for contract upgrades and 1/7/14-day delay for administrative contract methods based on the risks. During an emergency situation, the contract allows Scroll multisig to pull an immediate pause of the bridge and rollup to contain the loss. Scroll multisig currently also has the ability to bypass the delay for an instant upgrade in an emergency event. Once the system has matured and proven its stability, Scroll will establish a security council and transfer the bypass privilege to its multisig.

4. Summary

To be decided by BDG