Secpk-Verifications Bloat

On the afternoon of Feb 21, 2022 the community started seeing impacted network performance stemming from the launch of the Shade airdrop. The performance degradation had multiple reasons and a lot of lessons were learned from this stress test. Some of these findings are documented here so to educate network participants.

The community came together quite quickly to solve as many problems as possible, the incident response can be found here:https://forum.scrt.network/t/discuss-network-issues-w-shade-airdrop-2-21-22/5475

What happened?

On Feb 21, 2022 at around 11pm UTC the Shade protocol airdrop was launched, drawing a lot of attention to Secret Network.

As a part of their airdrop mechanism, Shade heavily utilized secp256k1 signature verification in their contracts, which is very computationally expensive.

These transactions are causing blocks to slow down due to the time required to compute each block, causing the mempool to fill up which delays the execution of transactions. A further effect of blocks that take a long time to compute is that queries were slowed down as well, as at that time a node could not both compute a block and serve a request.

The network was clogged with transactions for multiple days on end because of the high demand and slow block times, after rate limiting the shade airdrop application the network picked back up again so that the rest of the applications were no longer affected.

Why Did This Happen?

The reason for the abnormal behavior is mostly due to nodes running an outdated WebAssembly engine, which does not handle long computations very efficiently. Also, gas calculations do not account for this inefficiency, which further compounds the issue.

To put it simply, these Secpk verifications took more computation power than they were paying for in Gas. A full block would therefore be several magnitudes more complex to compute which made it so that validators were not done in the normal 6 second time frame causing block time to become longer and longer. Longer block times means less space for transactions per second causing the network to be too slow to handle transactions.

A secondary reason as to why validator nodes were not able to meet the computational demand was not related to hardware, but to peering. Validators speak to their peers so to pass information about consensus along. Validators can set persistent peers but the network can decide to completely cut off certain validators when the network is being stressed. What happened is that during the chain congestion only certain groups of validators were talking to each other. When 67% of voting power was found the block would be signed leaving the rest of the validators to not sign at all. This created very spotty patterns in the nodes who signed blocks, some were signing every one and others were signing none even though they were both done before the block was committed.

Quick note: The shinobi protocol testnet launched slightly before the shade airdrop also caused significant slowdown of the network, blocks were full just from a few of their transactions at the same time.

it was later recognised that this was because of the same problem with Secpk verifications which Shinobi uses for light client verification's of cross-chain bitcoin transfers. What has been done

1. Firstly, a small upgrade was released which significantly improved the query node performance. This upgrade allows nodes to both serve many more requests, and lessen the impact of long block computations. This will help services like Keplr stay available during network-wide events. Reference:https://github.com/scrtlabs/SecretNetwork/releases/tag/v1.2.5
2. The execution performance for computationally expensive functions like secp256k1 verification ere changed to being exposed to contracts (instead of being executed inside the contract) which made them much more efficient. New APIs were released during Shockwave Alpha which brought 500x improvements to these transactions. Reference:https://github.com/scrtlabs/SecretNetwork/releases/tag/v1.3.0
3. Introduced Seeds for solving the validator peering issues. One can reference thevalidator documentation
4. to add these seeds to their peering list.
5. We are also replacing our WASM engine with a newer, more performant one. This item is still on theroadmap
6. and will help with long term scalability of Secret Network.
7. Lastly, we will also be re-evaluating gas calculation and pricing and try to adjust the gas to more accurately reflect the computational cost of each contract. This was a huge lesson learned, gas needs to equal the computational cost or nodes will not be able to handle the load.
8.

For some extra information on lessons learned from this event you can read this bloghttps://scrt.network/blog/scrt-labs-update-scaling-secrets