Versioning & Changelog

Detailed Changelog of Secret Network can be found on the Github.

https://github.com/scrtlabs/SecretNetwork/blob/master/CHANGELOG.md ```

Copy

CHANGELOG

1.13.0

- Support DCAP attestation
- EPID attestation is still supported, until it'll be phased-out by Intel
- Migrate to SGX 2.20
- Bump github.com/cosmos/ibc-go/v4 from 4.5.1 to 4.6.0.

1.12.0

- Fix the hardcoded admins feature
- Add hardcoded admins according to proposals 269 (Shillables) & 270 (Sienna).
- Fix PFM to stop dropping packets of IBC contracts.
- This has always been a bug in PFM. It was introduced in v1.9 and was missed because of a bug in our CI system.
- Fixed the bug in PFM and updated the dependency.
- For more info see https://github.com/cosmos/ibc-apps/pull/105.
- Add `admin` to `WasmMsg::Instantiate` in cosmwasm-std (Thanks @luca992!).
- This allows contracts to specify an admin address when instantiating other contracts.
- See usage example here.
- Update IBC to v4.5.0

1.11.0

- Added ibc-hooks middleware by Osmosis.
- WASM hooks: allows ICS-20 token transfers to initiate contract calls, serving various use cases.
- Example: Sending tokens to Secret and immediately wrapping them as SNIP-20 token. For example, `ATOM on Hub -> ATOM on Secret -> sATOMS on Secret` (2 transactions on 2 chains) now becomes `ATOM on Hub -> sATOM on Secret` (1 transaction).
- Example: Cross-chain swaps. Using IBC Hooks, an AMM on Secret can atomically swap tokens that originated on a different chain and are headed to Secret. The AMM can also send those tokens back to the originating chain.
- Axelar GMP: Using IBC Hooks, a contract on Ethereum can call a contract on Secret and get a response back.
- Ack callbacks: allow non-IBC contracts that send an `IbcMsg::Transfer` to listen for the ack/timeout of the token transfer. This allows these contracts to definitively know whether the transfer was successful or not and act accordingly (refund if failed, continue if succeeded). See usage example here.
- Added an optional `memo` field to `IbcMsg::Transfer`, to ease to use of the IBC Hooks ack callbacks feature. See usage example here.
- Added contract upgrade feature.
- On init, the creator can specify an admin address.
- The admin can migrate the contract to a new code ID.
- The admin can update or clear the admin address.
- The admins of contracts that were instantiated before v1.10 are hardcoded according to proposal 262.
- Hardcoded admins can only be updated/cleared with a future gov proposal.
- When the new `MsgMigrateContract` is invoked, the `migrate()` function is being called on the new contract code, where the new contract can optionally perform state migrations. See usage example here.
- Fixed a scenario where the enclave's light client might fail a valid node registration transaction.
- Add support for uploading contracts that were compiled with Rust v1.70+.
- Update Cosmos SDK to v0.45.16
- Update Tendermint to CometBFT v0.34.29
- Update IBC to v4.4.2
- Update IAVL to v0.19.6
- Update Packet Forward Middleware to v4.1.0
- Fix initialization of x/vesting module
- Add `env.transaction.hash` to support SNIP-52
- SNIP-52: https://github.com/SolarRepublic/SNIPs/blob/3cc16b7/SNIP-52.md#notification-data-algorithms

- See usage example here.
- Flush the enclave's cache in a random order

1.10.0

Patch against SGX Downfall vulnerability. See v1.10 proposal for more info.

1.9.3

- Bump ibc-go from v4.3.0 to v4.3.1 (Huckleberry security patch)

1.9.2

- Fix the v1.9.0 upgrade

1.9.1

- An atempt to fix the v1.9.0 upgrade

1.9.0

- New Feature: Randomness injection for secret contracts.
- Eliminates the need for contracts to bootstrap their own entropy pool.
- Unique for every contract call.
- Useful in lotteries, gaming, secure authentication protocols, protocols where unpredictable outcomes are essential for fairness and security, and much more. For more infomation on how to use this feature, see the documentation
- New Feature: FinalizeTx.
- Contracts can force the transaction to finalize at a certain point, otherwise revert.
- Example: protect against sandwich attacks and potential transaction rollbacks.
- Example: protect against cheating in gaming applications, where a malicious player could try to rollback a transaction in which they lost.
- IBC: Updated ibc-go from v3.4.0 to v4.3.0.
- New IBC Feature: Added packet-forward-middleware by Strangelove.
- Other chains would be able to more easily route SCRT in the interchain. For example, sending SCRT from Osmosis to Hub now becomes a single transaction from `Osmosis -> Secret` rather than a transaction from `Osmosis -> Secret`, then a transaction from `Secret -> Hub`.
- New IBC Feature: Added IBC fee middleware.
- Creates a fee market for relaying IBC packets.
- New IBC Feature: Added IBC panic button.
- Quickly shut down IBC in case of an emergency.
- New Feature: Evaporate & Check Gas APIs The new Check Gas and Evaporate APIs allow contract developers to create contracts that consume a constant amount of gas, independently of their code path. This helps harden contracts against information leakage from the amount of gas consumed by a contract.
- Bug Fix: Fixed an issue where nodes would sometimes stop if failing to enter SGX enclave
- Bug Fix: Fixed a bug where stopping and restarting a node would often cause the node to apphash
- Bug Fix: Fixed an issue where storing and deleting a key from storage in the same msg would cause it not to be deleted

1.8.0

Fixed a critical bug in 1.7.0 that prevented new nodes from joining the network and existing nodes from restarting their secretd process.

1.7.0

- Added the ability to rotate consensus seed during a network upgrade
- this will be executed during this upgrade
- Added expedited gov proposals
- Initial params (can be amended with a param change proposal):
- Minimum deposit: 750 SCRT

- Voting time: 24 hours
- Voting treshhold: 2/3 yes to pass
- If an expedited proposal fails to meet the threshold within the scope of shorter voting duration, the expedited proposal is then converted to a regular proposal and restarts voting under regular voting conditions.
- Added auto-restaking - an opt-in feature that enables automatic compounding of staking rewards
- Added light-client validation for blocks
- Protects against leaking private data using an offline fork attack
- Enables trusted block heights and block time to be relied on by contracts

1.6.0

- Fixed issue causing registrations to fail
- Changed internal WASM engine to Wasm3 from Wasmi. Contract performance increased greatly
- ~~Added seed rotation. On upgrade the network will fetch a new seed and use it for derivation of keys and encryption~~
- Seed rotation has been delayed to the next network upgrade

- Changed default grpc-concurrency to false (from true). For nodes that serve API requests, we highly recommend setting this manually to true as described in the release notes for version 1.5.1

- Changed peer rejected tendermint log to debug so it's hidden by default (from info)

- Increased tendermint query limit to 1000 (from 100)
- Bumped to tendermint 0.34.24
- Bumped to cosmos-sdk 0.45.11

- Changed base gas prices:

- Default instruction cost 1 -> 2

- Div instruction cost 16 -> 2
- Mul instruction cost 4 -> 2
- Mem instruction cost 2 -> 2
- Contract Entry cost 100,000 -> 20,000
- Read from storage base cost 1,000 -> 10

- Write to storage base cost 2,000 -> 20

- SecretJS 1.5 has been released, and uses GRPC-Gateway endpoints. Check it out: https://www.npmjs.com/package/secretjs or https://github.com/scrtlabs/secret.js

- Add check-hw tool that returns patch-level and compatibility information for hardware

1.5.1

29/11/2022 - Startup fix due to TCB recovery - startup validation on 1.5.1 does not account for SW_HARDENING_NEEDED including INTEL-SA-00615 in it's response. Registering using this binary will not work, however restarting your node can be done using the _startup_bypass packages.

Fix for GRPC-gateway concurrency. This will greatly improve performance on nodes serving queries to GRPC-gateway requests (REST requests going to v1beta1/blah/blah)

Note that concurrency is toggled from the "concurrency" flag under GRPC in app.toml - see an example below. (max-send-msg-size and max-recv-msg-size are also new, but are less important)

In this version the default is set to true for ease of deployment - We currently recommend for validators to not update to this version, or if you do, manually set concurrency to false. This update is for node runners that provide API services to increase performance of nodes that serve queries.

In this release we also include a special version for nodes that serve queries from legacy apps (those that use secretjs 0.x, for example Keplr) - this version contains an unstable patch that serves legacy LCD and rpc requests much faster. This fix has been found to be unstable at scale, and should be used sparingly until secretjs 1.5 is released and apps upgrade to the latest version

```toml [grpc]

# Enable defines if the gRPC server should be enabled.
```

enable = true

Address defines the gRPC server address to bind to.

address = "0.0.0.0:9090"

The default value is math.MaxInt32.

max-recv-msg-size = "10485760"

The default value is math.MaxInt32.

max-send-msg-size = "2147483647"

Concurrency defines if node queries should be done in parallel.

This is experimental and has led to node failures, so enable with caution.

The default value is true.

concurrency = true ```

1.5.0

- Fix IBC contracts bug (#1199)
- Fix creating accounts using ICA (#1215)
- Fix node registration (#1221)
- Fix nested attributes in contracts reply (#1241)
- Fix state sync (#1243)
- Fix some protobuf type names (256d9b)
- Update cosmos-sdk from v0.45.9 to v0.45.10
- Update Tendermint from v0.34.21 to v0.34.22
- Update ibc-go from v3.3.0 to v3.3.1

1.4.1-patch.3

- Patch againt the IBC Dragonberry vulnerability
- Update cosmos-sdk from v0.45.5 to v0.45.9
- Update Tendermint from v0.34.19 to v0.34.21

1.4.0

- CosmWasm v1
- Bump WASM gas cost:
- Base WASM invocation 10k -> 100k
- WASM storage access 2k per access
- Support MetaMask pretty signing
- Ledger support for x/authz & x/feegrant
- Revert Chain of Secrets tombstone state and restore slashed funds
- Update ibc-go from v3.0.0 to v3.3.0

1.3.1

- Use all available cores to serve queries.
- Mainnet docker image with automatic node registration & state sync (docs).
- Mempool optimizations (Thanks @ValarDragon!). For more info see this.
- Fix missing libsnappy1v5 dependency for rocksdb deb package.
- Update {LCD_URL}/swagger/ for v1.3 and add {LCD_URL}/openapi/.

1.3.0

- Bug fix when calculating gas prices caused by queries. This is will increase gas prices for contracts that use external queries, and will more accurately reflect resources used
- Update cosmos-sdk from v0.44.5 to v0.45.4
- Add the secretd rollback command
- Add the ~/.secretd/.compute directory to state sync
- Full changelog: cosmos-sdk/v0.44.5...v0.45.4
- Update tendermint from v0.34.16 to v0.34.19
- Fix registration failure for Intel Xeon 23xx-series processors (icelake still unsupported)
- Floating point checks no longer ran on execute (only on init)
- Update ibc-go from v1.1.5 to v3.0.0
- Added support for ICS27 - default host messages include voting, delegate/undelegate and voting
- Full changelog: ibc-go/v1.1.5...v1.3.0
- Backport API from CosmWasm v1: -ed25519_verify() -ed25519_batch_verify() -secp256k1_verify() -secp256k1_recover_pubkey()
- Add new secret CosmWasm API:

-ed25519_sign() -secp256k1_sign()

- Registeration service has been reworked. Registering a new node automatically now no longer requires a node to function properly. It also includes built-in support for the pulsar-2 testnet with the --pulsar flag.

- Secretcli now automatically appends either port 80 or port 443 when not providing any port using secretcli config if the node address starts with http:// or https://

1.2.6

Highlights

This version only a bug fix in the 1.2.5 release

1.2.5 - deprecated

Highlights

Architecture now split into query nodes and validator nodes. Query nodes contain optimizations that may not be entirely safe for validators and greatly improve querying performance. In addition, contracts are now served by two different enclaves: Query enclaves and execute enclaves. This will allow upgrading query enclave and improving performance without consensus-breaking changes. Lastly, rocksdb support is enabled. We are releasing binaries for each supported Database. Rocksdb is recommended for performance, but requires a resync of any nodes currently running goleveldb.

Secretd

- Added Rocksdb support (currently Ubuntu 20.04 only)
- Added new query node setup

SecretCLI

- Changed default behaviour to not print help on errors. Use -h if you miss it:)
- Added support for Ledger using Secret Network coin type (529). Creating keys using secretcli keys add x --ledger will use this by default. To create keys compatible with the Cosmos ledger app continue to use --legacy-hd-path (thanks @SecretSaturn)

References

-#879 Enclave multithreading + dedicated query enclave #881 Added telemetry measurements to compute module #881
-#882 Shutting up usage help by default in CLI #882 -#884 Bumping cosmos sdk version to v0.44.6 and added rocksdb

support

1.2.3

SecretCLI

- Fixed creating permits with Secretcli

1.2.2

Secretd

- Fixed issue where queries would try to access the Enclave in parallel from multiple threads, causing `SGX_ERROR_OUT_OF_TCS` to be returned to users when a node was under sufficient load. Queries now access the enclave one-at-a-time again.

1.2.1

This is a minor non-breaking version release.

SecretCLI

- Migrate the `secretcli tx sign-doc` command from v1. See this for more info.

1.2.0

Version 1.2.0 has been released - the Supernova upgrade!

Highlights

- Upgraded to Cosmos SDK 0.44.3. Full changelog can be found here
- Gas prices are lower - as a result of performance upgrades and optimizations, gas amounts required will be much lower.
- GRPC for cosmos-sdk modules in addition to legacy REST API. See API here
- New modules:
- Fee Grant - allows an address to give an allowance to another address
- Upgrade - Allows triggering of network-wide software upgrades, which significantly reduces the amount of coordination effort hard-forks require
- Auto Registration - The new node registering process is now automated via a new command `secretd auto-register`

API and endpoints

Registration module

- The endpoint `/reg/consensus-io-exch-pubkey` has been changed to `/reg/tx-key` and now returns `{"TxKey": bytes }`
- The endpoint `/reg/consensus-seed-exch-pubkey` has been changed to `/reg/registration-key` and now returns `{"RegistrationKey": bytes }`

GRPC and REST endpoints

GRPC endpoints have been added for cosmos-sdk modules in addition to legacy REST APIs, which remain mostly unchanged.

GRPC endpoints for the registration and compute modules will be added in a future testnet release

SecretCLI and Secretd

Unlike other cosmos chains, we chose to maintain the differentiating CLI and Node runner executable differences. SecretCLI still contains the interface for all user-facing commands and trying to run node-running commands using SecretCLI will fail. Secretd now contains both node-running and user-facing commands.

As a result of cosmos-sdk upgrade, some CLI commands will have different syntax

Secretd nodes now run the REST API (previously named LCD REST server) by default on port 1317. You can change this behavior by modifying /home/\<account>/.secretd/config/app.toml and looking for the `api` configuration options

SecretJS

Version 0.17.3 has been released! SecretJS has been upgraded to support the Supernova upgrade. All APIs remain unchanged, although the versions are NOT backwards compatible.

For compatiblity with 1.2.0+, use SecretJS 0.17.x. For compatiblity with 1.0.x (legacy), use SecretJS 0.16.x

CosmWasm

Secret-CosmWasm remains in a version that is compatabile with the v0.10 of vanilla CosmWasm, and previous versions compatible with secret-2 will still work with this upgrade.

A new feature has been added - plaintext logs. To send an unencrypted log (contract output), use `plaintext_log` instead of `log`. This allows contracts to emit public events, and attach websockets to listen to specific events. To take advantage of this feature, compile contracts with `cosmwasm-std = { git = "https://github.com/scrtlabs/SecretNetwork", tag = "v1.2.0" }`

Known Issues

- SecretCLI still uses /home/.secretd to store configuration and keys
- Signatures other than secp256k1 are unsupported for CosmWasm transactions.
- snip20 CLI commands not working
- IBC commands not yet working
- Fee grant messages not supported by CosmWasm
- SecretCLI incompatible on M1 Mac
- /reg/registration-key returns malformed data
- To register a new node the environment variable SCRT_SGX_STORAGE should be set to "./" or the registration process might fail
- SecretCLI/Secretd default gas prices are set to 0 while nodes default to 0.25uscrt

``` Last updated23 days ago On this page Was this helpful?Edit on GitHub Export as PDF
```