Relay-ring signature vault system

Nik Rykov nik@hns.is 10.07.2022

This document describes scheme of relay-ring signature vault network.

Introduction to problem

After invention of Bitcoin [1], there were created a lot of projects on blockchain technology. All these projects are unique, each chain is like unique dimension: Everything what happens there usually has no effect on other chains.

This has good side: If one chain is exploited, other chains remain unharmed. When Ethereum was exploited [2], other chains like bitcoin remained absolutely unharmed.

There's also bad example of how connected (via native IBC, we will return to it later) blockchain can influence other chains: Terra [3]. This blockchain was purposed as economical core for other blockchains. When peg of UST, their stablecoin, was ruined, connected blockchains suffered a bit: Kava's stablecoin USDX, that was backed mainly by UST, experienced peg movement too [4]. Thorchain and Osmosis experienced liquidity drain [5], and it highly affected cosmos-based coins: UST/<COIN> pools had heavy positions and big liquidity of coins, so coins pooled this way got heavy price decrease.

Problem

The problem of current existing blockchains is complexity of integration with other chains and control of this integration. We can't just merge Zcash with Bitcoin and Arweave now, and do it securely.

Current solutions

Oracles

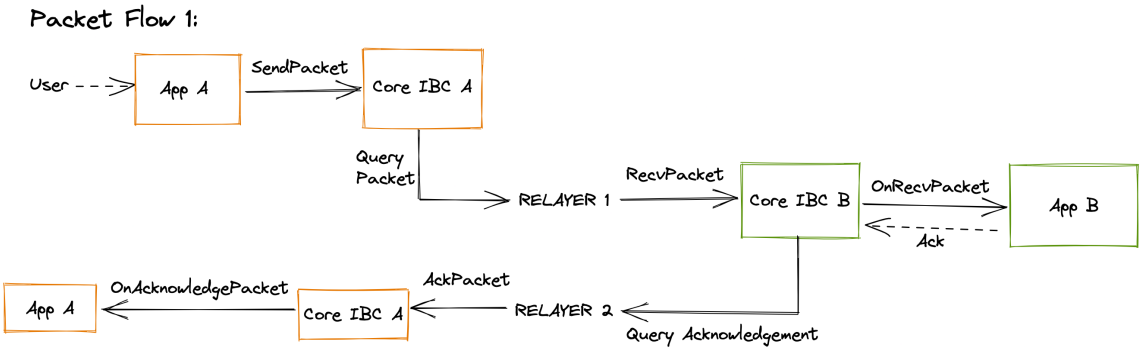
Chainlink [6] is first blockchain that made specially for running Oracles.

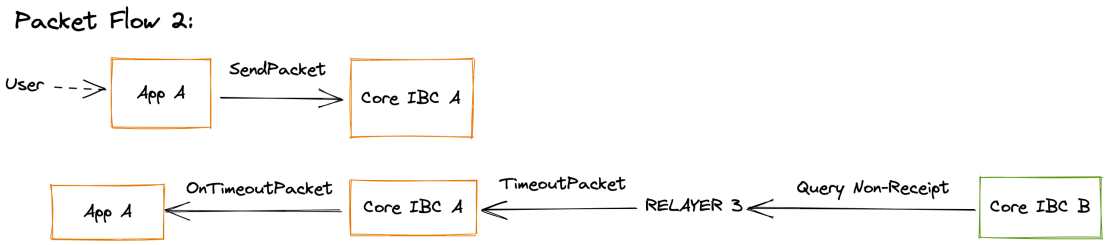
Oracles is first attempt to connect blockchain to outer world and other blockchains. It allows smart contracts on one chain to pull information from external sources via dedicated independent blockchain, relay-chain. This relay-chain's validators should pull data from external source, then reach consensus of what validator's data is valid, and then submit "reply" to smart contract that requested interaction via own smart contract set.

The main flaws of this scheme is that data is like under glass window: Contract can read data from other chain or other resource, but can't write anything. So contract on Arweave can't call contract on Ethereum and vica versa (if both contracts aren't having special communication library), it's not possible to transfer Solana token to Binance Smart Chain via oracles.

Inter-Blockchain Communication (IBC)

At first view, IBC seems like perfect solution of this problem. To use it, chain needs to integrate IBC Client, and dedicated relays should be risen. When one chain wants to interact with other chain, chains create *vouchers* that are delivered and processed by relays.





You can read more about it here: ibcprotocol.org [7]

IBC is great way to solve isolation problem, but it has several flaws: Firstly, IBC integration requires attention from chain that wants to be IBC-compatible. It of course broke isolation layer, but far away not for each chain and coin. Less likely that coins like Bitcoin, Ethereum or Monero will integrate it, as it requires full chain upgrade. At moment of writing (10.07.2022), IBC is limited only to chains based on Cosmos SDK and Tendermint PoS consensus. Second

flaw is control possibility of such scheme: If one chain is exploited, all contracts/tokens that use IBC connection with this chain somehow are in danger. This already happened with exploited Terra, when IBC DEX-es that had liquidities with UST were drained.

Suggested solution

The optimal solution can be found in relay-ring signature vault.

Description of relay-ring signature vault system (RRSVS)

Relay-ring is writable modulable oracle system that relies on relay operators that operate network for part of bridge coin inflation.

Relay-ring signature vault system can be only used if network supports multisignature accounts.

In RRSVS, relay operators have N of M multisignature accounts on each chain where oracle/bridge works. Where M=relays count in network and N = floor(M * 70%). So if network has 100 relay operators, signature of 70 relay operator needed to write interaction to supported network from protocol account on network.

Anyone can pay 0.3% of protocol coin supply to become a relay and get some of protocol coin inflation (optimal is 5% per year). In exchange for inflationary coin model, coin of this protocol can also act as Profit Sharing Token (PST), where holders receive part of protocol commissions (for bridging tokens from one network to another, interchain smart contract calls). Protocol should have commission to prevent overload abuse and reward holders of it's token, lowering or fully leveling coin inflation.

This multisig scheme allows our network to write transactions to other networks in decentralized way (As decentralized as it is possible with Proof of Stake security scheme).

Protocol should have governance, where 50%+ of protocol coin holders decide protocol software upgrades, adding and removing of new networks support, adding and removing of new tokens bridged, freezing of interaction with token or whole network.

Special attention should be paid to governance for freezing tokens and network's interactions: Freezing of token/network operations should be done as soon as proposal got 33%+ of protocol tokens, without waiting for proposal end time. It will allow community to suspend interactions with exploited networks/tokens as soon as possible, without significant loses. Relay-ring was inspired by Thorchain vault system [8], Chainlink oracle system [6], and Nomic's Proof-of-Stake Bitcoin Sidechains [9].

Technical description

Modularity

To allow fast adding of new networks support to protocol, protocol software should be highly modulable: If it's modulable and API is relatively easy, third party developers that are familiar with network specifications will be able to write module that will connect protocol with other networks.

Below is described one of possible module structures:

- API & Cryptography engine module

  > "The core", module that is responsible for providing base API for descending modules. All external things should be connected and used only here.

- Communications module

  > Module that is responsible for communications between relay nodes. Note: It's highly unrecommended to have pure ephermal p2p network as core of this module! If core is ephermal (interactions between nodes aren't stored for relatively long time), then it becomes more complex to detect "bad" relays (relays that are in downtime, purpose malicious multisig spends) and punish them. Variants like own Proof of Stake blockchain (we already have relays that locked tokens to validate transactions from multisig, so we can use it to validate blocks too) or lazy smart contracts (on cheap and scalable chain as base) are preferred.

- Internal network behavoir controller module

  > Module that is responsible for governance, protocol staking rewards, forms API for slashing and commission distribution to protocol coin holders. This module also calls "multisig

> migration" submodule in token modules. Also it controls all network configuration.

- External networks connector modules

  > Set of modules that is responsible for connection of external networks (like Arweave, Ethereum, Binance Smart Chain, Juno, Solana) to relays. Each module should be responsible for forming and managing multisig specially for module's network, fetching data from this network, providing data to internal network behavoir controller module to form read-oracles, serialization and signing data received from internal network behavoir module. Each network module should be easy suspendable: Main network should continue working when one of network modules is frozen by governance. This module also should provide "multisig migration API" submodule to internal network behavoir controller module. This submodule should move all access from one multisig (one set of pubkeys) to another multisig. It's required for cases when new relay is added to network or when one of current relays is jailed. Should also call interface module when network (via smart contract or explicit interaction) tries to query/interact with oracle or other network via protocol. May also provide submodule for making and managing tokens on this network (for cases when we want to bridge token from foreign network to this network via our protocol), but it's unnecessary (then only bridging *from* this network will be supported, but not *to*).

- Token bridge modules

  > Module that is child of one of network modules. Should provide basic API (transfer, acknowledge receiving, vault size, etc.) for specified token on network (for example to UNI on ethereum network). Should be disableble by governance without consenquences for upper modules (even for network module). Should provide "Token multisig migration API" submodule to network's multisig migration API submodule. This submodule should move all vault's tokens from old multisig to updated.

- Reading oracle module

> This module should provide API for querying smart contracts on it's network module to interface module. "Standard" oracle.

- Writing oracle module

> This module should provide API for submitting interactions with smart contracts from network's multisig on it's network to interface module. Note: Smart contract to whom interactions are addressed should explicitly identify that it supports interactions from protocol's network. Otherwise it opens doors for spending vault's tokens avoiding token module, where exploiter wants.

- Interface module

> Module that handles external interactions with protocol (Like governance activity, bridging some tokens from one network to other, querying and interacting with smart contracts on foreign networks via our protocol).

Such overmodularity will allow developers of foreign networks and tokens to integrate it as smooth as possible, will allow governance to freeze modules in emegrency situations, and will allow smooth forking of network when governance of current network not wants network/token to be integrated, but foreign network's community wants to integrate with other chains via relay-ring.

References

[1] Bitcoin: A Peer-to-Peer Electronic Cash System
https://bitcoin.org/bitcoin.pdf

[2] Understanding The DAO Attack
https://www.coindesk.com/learn/2016/06/25/understanding-the-dao-attack

[3] The Collapse of Terra's LUNA and UST: What Happened
https://coinmotion.com/terra-luna-and-ust-what-happened

[4] Kava USDX Suffers Devaluation as UST Continues Implosion
https://cryptoadventure.com/kava-usdx-suffers-devaluation-as-ust-continues-implosion

[5] Use Hardfork to Accelerate Proposals #222, #223, and #224
https://www.mintscan.io/osmosis/proposals/225

[6] ChainLink: A Decentralized Oracle Network
https://research.chain.link/whitepaper-v1.pdf

[7] The Interblockchain Communication Protocol: An Overview
https://arxiv.org/pdf/2006.15918.pdf

[8] THORChain: A Decentralised Liquidity Network
https://github.com/thorchain/Resources/blob/master/Whitepapers/THORChain-Whitepaper-May2020.pdf

[9] Proof-of-Stake Bitcoin Sidechains
https://gist.github.com/mappum/da11e37f4e90891642a52621594d03f6