Relay-ring signature decentralized automated wallet system

Nik Rykov nik@hns.is 12.07.2022

This document describes scheme of relay-ring 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 is good for security, but bad for flexibility: Like saying that trading isolation is good for country because then economical crisis can't affect this country. We already have some solutions to solve this isolation problem: centralized bridges, oracles, IBC.

Problem

The isolation problem of current existing blockchains is complexity of integration with other chains, security and flexibility of this integration. Bitcoin has the highest liquidity among cryptocurrencies, Ethereum has the best ecosystem around smart contracts, Arweave is the most scalable chain with scalable permanent data storage, handshake is the most successful blockchain-based naming system, Cosmos has best flexibility with the losest cost (PoS) among blockchains, allowing us to have progressive dApps like decentralized vpn (dVPN) and decentralized cloud compute (akash).

That's all is very cool! But the problem is that we can't use all at once: Can't buy handshake SLD from ethereum smart contract using BTC as currency, use arweave as DNS records storage for it, and can't place it for sale at stargaze (cosmos ecosystem's NFT marketplace) marketplace.

Relay-ring protocol acts like big decentralized gate between all these chains, allowing to deliver any possible interactions between any networks without centralization problem and without requirement to integrate protocol support on core blockchain level (letting us to bridge chains like bitcoin, which will unlikely integrate any external protocols to its core).

Current solutions

Oracles

Chainlink [2] 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.

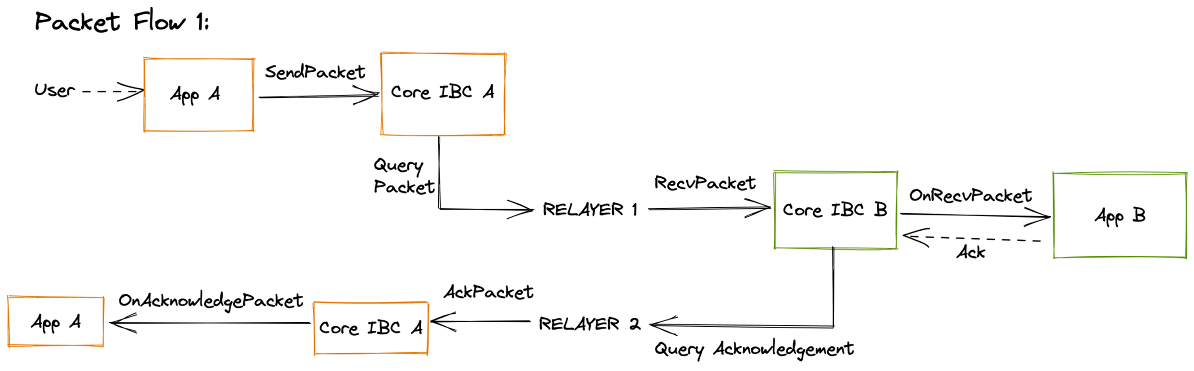
The main flaws of this scheme is that data is like under glass window: Contract can read data from other chain or other resource, but can't write anything. So contract on Arweave can't call contract on Ethereum and vica versa (if both contracts aren't having special communication library).

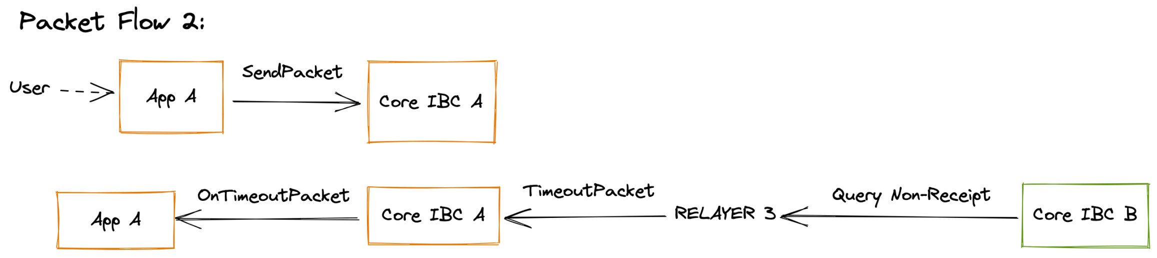
There of course exist solutions to call one chain's contract from another chain via oracles, but usually it requires special treatment from calee contract (contract that is called). To better understand it you can read about FCP (Foreign Call Protocol) [3]

Inter-Blockchain Communication (IBC)

IBC [4] can be good solution of isolation problem when you need to connect two (or more) chains, these chains are Proof of Stake based, chains are IBC-comaptible (IBC support is enabled in network, validators use software that allows IBC).

To use IBC, chain needs to integrate IBC Client, and relays between one chain and another should be rised. When one chain wants to interact with other chain, chains create *vouchers* that are delivered by relays and processed by validators of both chains.





You can read more about it here: ibcprotocol.org [4]

IBC is great way to solve isolation problem, but it requires too much from chain: To be proof-of-stake based and have own implementation of IBC protocol in chain software. So chains like Bitcoin will never have IBC support (If not using relay-ring protocol described in this document to enable IBC support)

Suggested solution

Relay-ring is one of possible chain-effortless (chains not requiring to do anything for relay-ring integration) writable oracle solutions.

Description of relay-ring signature vault system (RRSVS)

Relay-ring is writable 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 relay-sign protocol, relay operators have N of M multisignature accounts on each chain where oracle/bridge works. Where M=relays count in network, N = floor(M * (O/100)), and O is percent of required "fair relays", percent of signatures needed to relay interactions. So if network has 200 relay operators (M) and required signatures threshold (O) is 70, signature of 140 relay operators needed to write interaction to supported network from protocol account on this network.

Anyone can pay some coins to become a relay and get protocol coin inflation in return. 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 automated multisignature scheme allows protocol to write transactions to other networks in decentralized way (As decentralized as it is possible with Proof of Stake security scheme).

Technical description

Modularity

To allow adding support 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).

- 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.

For networks without builtin theshold signature accounts, protocol can use RSA Threshold Signatures [7]

Relay-ring writable oracle protocol was inspired by Thorchain vault system [5], Chainlink oracle system [2], and Nomic's Proof-of-Stake Bitcoin Sidechains system [6].

References

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

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

[3] Foreign Call Protocol Specification https://www.notion.so/Foreign-Call-Protocol-Specification-61e221e5118a40b980fcaade35a2a718

[4] Inter-blockchain communication protocol

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

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

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

[7] Implementation and Discussion of Threshold RSA https://courses.csail.mit.edu/6.857/2016/files/33.pdf