GGX

Interchain Infrastructure Protocol

Matthew Doty Viktor Ihnatiuk Yehor Butko
Artur-Yurii Korchynskyi Timofei Sokolov
Danyil Poliakov

1 Introduction

The blockchain ecosystem is currently undergoing a transformative phase characterized by a shift from a predominantly centralized paradigm to a decentralized and multi-chain landscape. Market dynamics and a growing demand for scalability and innovation within the blockchain space have fueled this evolution. Bitcoin (BTC) and Ethereum (ETH), two of the most influential blockchain networks, are at the forefront of this transformation, collectively commanding more than 60% of the market capitalization. However, despite their significance, both Bitcoin and Ethereum face inherent limitations, such as their modest transaction throughput, slow release cycles, and scalability challenges, which have spurred a surge in the development of Layer 1 and Layer 2 blockchain solutions.

The proliferation of these diverse blockchain platforms brings a pressing need for efficient cross-chain communication tools, particularly in *Decentralized Finance* (DeFi) applications. This burgeoning multi-chain ecosystem offers immense potential but also poses a complex set of challenges. One crucial challenge is establishing practical liquidity routing mechanisms to facilitate seamless interactions among these disparate chains. Moreover, the nascent chains in this ecosystem often suffer from limited accessibility, impeding broader community participation and imposing substantial market friction. Indeed, empirical evidence[6, 8] suggests that market friction remains closely tied to liquidity in the cryptocurrency landscape.

Of paramount significance within this evolving landscape is the imperative to solve the interchain communication problem, a matter of substantial economic consequence. Billions of dollars are currently locked within cross-chain custodial contracts, exemplifying the magnitude of the challenge. The industry's heightened interest in this area has given rise to extensive research, yet numerous questions still need to be answered.

In response to these challenges, we introduce the GGX blockchain—a novel solution poised at the crossroads of the Bitcoin, Cosmos, and Ethereum communities. GGX addresses the pressing need for efficient cross-chain communication and offers innovative solutions for decentralized custody and DeFi liquidity provisioning. To comprehend the significance of GGX, it is crucial to first delve into the transformative developments within the Bitcoin ecosystem, notably the Bitcoin Taproot upgrade.

The Bitcoin Taproot upgrade, along with the introduction of ordinals, enabling the creation of fungible tokens in the form of BRC-20s and Non-Fungible Tokens, has unlocked exciting opportunities within the DeFi space. Furthermore, this upgrade has ushered in the era of Taproot assets[11], enabling tokens to be issued and traded on the Bitcoin Lightning Network, offering unparalleled scalability. However, to fully harness the potential of these developments, a decentralized custody solution for Bitcoin is paramount, especially for enhancing liquidity on other blockchain platforms such as Ethereum and the Cosmos ecosystem.

The inadequacies of the current custodial solution, Wrapped Bitcoin (WBTC), are evident from its historical challenges, including episodes where it temporarily lost its peg, notably during the Alameda meltdown[3]. GGX serves as a groundbreaking remedy to this custodial pain point, bringing a decentralized solution for providing Bitcoin liquidity to the broader blockchain ecosystem. This achievement is made possible by leveraging the recent Bitcoin Taproot upgrade, which enables threshold Schnorr signatures[17, 12], thereby facilitating decentralized custody of Bitcoin assets.

Safety-critical interchain messaging is at the core of GGX's architecture. It combines the InterBlockchain Communication Protocol (IBC) with a decentralized Ethereum Oracle, laying the foundation for trustless cross-chain communication. Implementing a trustless Ethereum Oracle is part of GGX's commitment to security and decentralization. By incorporating an Ethereum light client at its consensus layer, GGX eliminates the need for reliance on centralized infrastructure providers for Ethereum data. This dual-layered approach, utilizing both threshold Schnorr and threshold Secp256k1 signatures for Ethereum smart contracts, establishes a robust and crypto-agile security architecture, ensuring resilience against potential security breaches.

In conclusion, the significance of cross-chain messaging in the DeFi landscape cannot be overstated. As the blockchain ecosystem evolves, GGX is situated as a DeFi gateway for Bitcoin, offering innovative solutions for interchain communication, decentralized custody, and liquidity provisioning. GGX represents a pivotal step towards realizing the full potential of a multi-chain blockchain ecosystem, enabling greater accessibility, reduced market friction, and enhanced DeFi opportunities for the broader community.

2 Threshold Signatures

Within GGX Chain, the secure management of cryptographic signatures is a cornerstone of trust and integrity within the blockchain ecosystem. Threshold signature schemes, a cryptographic paradigm, are prominent in enhancing decentralized networks' security and reliability. These schemes empower distributed entities to generate a single cryptographic signature from a collectively held distributed private key. Termed *threshold signatures*, these cryptographic protocols offer a unique advantage: any subset of decentralized participants can collaboratively produce a valid signature, provided that their numbers exceed a predetermined threshold established during the distributed key generation phase. For example, in a network comprising 30 participants, a threshold set at 20 during key generation means that any combination of 20 or more participants can collectively create a legitimate signature. This inherent flexibility in signature generation enhances security and fosters the resilience and availability of cryptographic operations within distributed systems.

In the blockchain ecosystem, where nodes may exhibit faulty or malicious behavior, an additional layer of security becomes imperative. This layer is embodied in the "identifiable abort" concept, a critical feature in threshold signature schemes, especially in a blockchain setting. Identifiable abort empowers the distributed key generation process to terminate whenever a bad actor or malfunctioning node is detected. Given the potential presence of adversarial or unreliable nodes within blockchain networks, the ability to pinpoint and address issues during the distributed key generation phase becomes paramount for safeguarding the network's integrity and the cryptographic operations upon which it relies.

GGX Chain leverages two such cutting-edge schemes, each offering unique advantages in the context of distributed ledger technology:

1. Komlo and Goldberg's *FROST: Flexible Round-Optimized Schnorr Threshold Signatures*[9]: This scheme represents a significant advancement in Schnorr threshold signature protocols. FROST optimizes the network overhead during signature generation, reducing the communication rounds among signers. It introduces innovative techniques to protect against forgery attacks and provides the ability to safely perform signing operations in a single round without constraining the concurrency of signing processes. Furthermore, FROST incorporates an abort mechanism to handle misbehaving participants, a practical consideration for real-world deployment scenarios.

2. Genero and Goldberg's *One Round Threshold ECDSA with Identifiable Abort*[4]: This protocol is used for generating threshold secp256k1 signatures. It addresses the unique challenges posed by threshold ECDSA signatures, particularly relevant in cryptocurrencies. It introduces a highly efficient and non-interactive online phase, enabling asynchronous participation by players without the need for simultaneous online presence. Importantly, as previously discussed, it offers identifiable abort capabilities, allowing the protocol to halt when a misbehaving participant is detected. This feature minimizes the risk of catastrophic failures caused by dishonest actors within distributed settings.

In practice, we use threshold Schnorr signatures for interacting with Bitcoin and both signature schemes when interacting with Ethereum.

3 Communication & Liquidity Protocols

3.1 Overview

There are different mechanisms for communication and liquidity routing used by GGX:

1. Threshold Signature Bitcoin Custody
2. Incentivized Message Delivery Protocol (IMDP)
3. Interblockchain Communication Protocol (IBC)

The mechanisms differ in implementation, guarantees provided, and transaction fees. GGX supports connections via different mechanisms to a specific external chain.

3.2 Threshold Signature Bitcoin Custody

GGX Chain has been designed to offer a secure and efficient solution for distributed Bitcoin custody. One of the fundamental challenges in the blockchain space is to strike a balance between security, decentralization, and efficiency. GGX Chain employs threshold Schnorr signatures to achieve this balance, ensuring that users' assets are securely held in a distributed manner while maintaining efficiency.

Threshold Schnorr signatures are at the core of GGX Chain's distributed Bitcoin custody solution. The threshold is set such that validators representing 66% of the network stake can always construct a valid Schnorr signature. This design ensures that a supermajority of network participants must cooperate in signing transactions, enhancing security.

GGX requires users to transfer their Bitcoin tokens to a specific TapScript address to enable the utilization of threshold Schnorr signatures. These addresses use a 62-character-long bech32m format, slightly longer than the traditional 26-34 character-long Bitcoin addresses. TapScript addresses are necessary to enable the distributed ownership and secure key management features provided by GGX Chain.

GGX Chain employs a periodic process known as *sweeping* to consolidate Unspent Transaction Outputs (UTXOs) into a single UTXO suitable for key rotation. This consolidation process is initiated when transaction fees on the Bitcoin network are low, but it generally involves a fee that is charged on Bitcoin liquidity transfers. It is important to note that crucial rotation also requires transaction fees, which must be sourced from interchain transfers.

Key rotation is a crucial aspect of the GGX Chain's security model. Even though GGX Chain operates as an open proof-of-stake blockchain, it enforces that when validators decide to rotate, at least 66% of the existing validators must remain the same. This requirement ensures continuity and security during the transition.

The key rotation process involves transitioning from the old signature scheme to a distributed key controlled by the new validators. This transition is orchestrated so that the new validators can construct a valid signature for the old signature scheme and rotate the signatures to the distributed key controlled by the new validator set. This approach ensures a smooth and secure transition of custody control while maintaining the security and integrity of the GGX Chain.

In conclusion, GGX Chain's use of threshold Schnorr signatures, TapScript addresses, UTXO sweeping, and a carefully designed key rotation protocol collectively contribute to efficient, succinct distributed ownership of Bitcoin within the blockchain network. These features allow users to securely manage their assets while benefiting from the security and decentralization of GGX Chain.

3.3 Incentivized Message Delivery Protocol

GGX uses a novel *Incentivized Message Delivery Protocol* (IMDP) for efficient cross-chain communication. IMDP uses a network of *couriers* to deliver messages. Any computer may run a Courier node. Courier nodes run GGX light clients. Courier nodes pay gas fees for communicating messages from GGX. In exchange, they earn GGX token rewards. While validators could run Courier nodes if they wished, the blockchain and courier network are designed to be distinct and decentralized.

The lifecycle of a message demonstrates the role of the different Golden Gate components play. We depict this lifecycle in Figure 1.

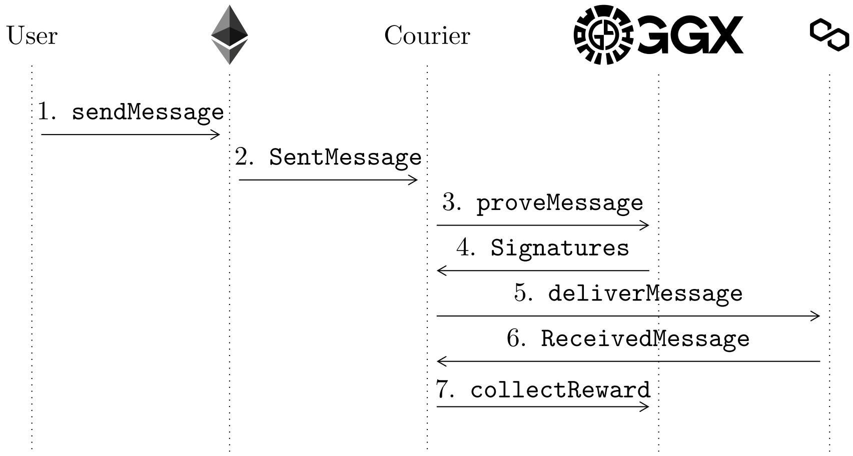

Figure 1: Lifecycle of a User's call to GGX's interchain communication smart contracts

1. To start, a User sends a message by invoking the `sendMessage` method on a designated smart contract on blockchain ♦ (note that ♦ is distinct from GGX). The User specifies a destination chain and address when they submit a message. In this case, the destination chain is denoted by ∞.

2. Next, the Courier detects the user's message via a light client for ♦, listening for a `SentMessage` event. Using ♦'s RPC, it constructs a Merkle proof that the `SentMessage` was included in the block where it was emitted.

3. The Courier proves the message was included in a block to GGX by calling the `proveMessage` RPC call.

4. GGX validates the proved message against its own consensus layer light-client. GGX then generates a two signatures for the Users' message. One signature is a threshold secp256k1 signature while the other signature is a threshold Schnorr signature (see the discussion in §2). These threshold signature schemes are discussed in §2. GGX's blockchain keeps a permanent record of the signed messages for retrieval. GGX puts the signatures

on an event stream. A Courier picks up the `Signatures` on GGX's event stream.

5. The Courier delivers the signed message to the destination chain by calling `deliverMessage` on the interchain communication contract on ∞.

6. The Courier detects a `ReceivedMessage` event from ∞ using its light client. As in step 2, the Courier uses ∞ to prove the event was included in a block.

7. The Courier calls `collectReward` on GGX and provides proof it delivered the message. After validating the proof GGX distributes a reward.

3.4 IBC Protocol

The *Inter-Blockchain Communication Protocol*[5] (IBC) is designed to facilitate communication of sovereign replicated ledgers that share only a minimum requisite common interface. IBC handles authentication, transport, and ordering of opaque data packets relayed between modules on separate ledgers can be run on solo machines, replicated by many nodes running a consensus algorithm, or constructed by any process whose state can be verified. The protocol is defined between modules on two ledgers, but designed for safe simultaneous use between any number of modules on any number of ledgers connected in arbitrary topologies. IBC requires certain functionalities and properties of the underlying ledger. It requires *finality signatures*, cheaply-verifiable consensus transcripts, and a simple key/value store. On the network side, IBC requires only eventual data delivery — no authentication, synchrony, or ordering properties are assumed.

While IBC may be implemented on any blockchain with smart contracts[7], in practice the gas costs are too high. This is because IBC-supporting blockchains such as Cosmos use Ed25519 signatures. Ethereum does not have a precompiled contract for checking these signatures[10]. As a consequence, it costs 500,0000 gas to check an Ed25519 signature, and millions to run a light client.

GGX includes IBC support via the IBC substrate palette[15]. GGX parachains can also opt into IBC support via the same palette.

IBC does not provide specific protocol-level provisions for compute-level or economic-level flow control. The Courier network is expected to have compute throughput limiting and flow control mechanisms of their own such as gas markets.

We demonstrate the workflow of GGX's IBC protocol below and depict how messages flow in Figure 2.

1. To start, the User makes a `transaction` calling a designated smart contract on GGX, which emits an event `relayPacket`. GGX saves all information about the connection, channel and message. The User specifies a destination chain and address when they submit a transaction. In this case the destination chain is denoted by ⚛.

2. The Courier listens to GGX for a `relayPacket` event.

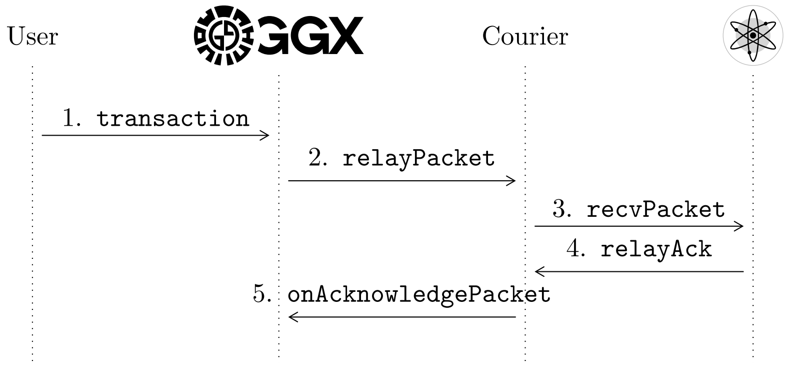

Figure 2: GGX Chain's IBC workflow

3. The Courier sends message to a designated smart contract on ⚛ invoking a `recvPacket` event.

4. Smart contract on ⚛ relays an acknowledgement to the Courier using a `relayAck` method.

5. The Courier[1] sends an acknowledgment to GGX calling an event `onAcknowledgePacket` on a designated smart contract.

4 Virtual Machine

GGx provides a flexible environment for running decentralized applications (dApps). GGx supports the *Ethereum Virtual Machine* (EVM) as well as *WebAssembly* (WASM). EVM support means that developers can write in either Solidity or Vyper. On the other hand, WASM support allows authors to write in Rust or any language that compiles to WebAssembly. GGx accomplishes this by using AStar's XVM pallet[2].

GGx supports multiple VM environments for better decentralized finance software. For example, suppose a Decentralized Autonomous Organization (DAO) wanted to air-drop a token with an exponential vesting schedule. The DAO would have 50% of the token vested in 2 years, 75% in 4 years, etc. Because blockchain VMs do not support floating-point, these calculations require fixed-point arithmetic. While a fixed-point library exists in Solidity for use with the EVM[16], Rust has higher-quality libraries for fixed-point arithmetic[13]. In other circumstances, EVM support is preferable.

[1] note that in this step the Courier may have changed and the protocol would still function properly.

5 Summary

GGX Chain's architectural sophistication, predicated on advanced technologies and protocols, fosters harmonious integration between Bitcoin and Ethereum-based smart contract chains and facilitates interchain compatibility via the IBC protocol within the Cosmos-based blockchain network.

GGX Chain achieves a paradigm shift by channeling the potential of the Taproot upgrade, meticulously elucidating the mechanics of distributed Bitcoin custody using threshold Schnorr signatures. This scientific exposition underscores the rigor applied to Bitcoin liquidity enhancement within the GGX Chain ecosystem, demystifying the intricacies of secure asset transfer and utilization while unequivocally preserving the quintessence of security.

GGX Chain introduces a pioneering security paradigm characterized by its multifaceted crypto-agile approach. Emphatically, it substantiates the architectural prudence of employing a dual-tier security model featuring both threshold secp256k1 and threshold Schnorr signatures. This methodological dualism advances the defense of user assets and amplifies the veracity of communication channels with smart contract platforms, underpinning an academic discourse on holistic blockchain security.

The GGX Chain acknowledges the intricate nuances of the ever-evolving blockchain ecosystem, providing an incisive analysis of communication protocols. GGX Chain's astute approach to interchain communication thrives on malleable communication protocols within the kaleidoscopic backdrop of coexisting blockchain networks. This research underscores the profound realization that the future landscape of interchain communication remains uncharted, and no singular protocol shall monopolize this domain.

GGX Chain emerges as an exemplar of academic exploration in the blockchain domain. It forges new frontiers by harmonizing Bitcoin liquidity, propelling interchain communication to new dimensions, and advancing multi-layered security. By synergistically embracing innovative technologies, such as Taproot and threshold Schnorr signatures, GGX Chain occupies the vanguard of blockchain intercommunication, poised to navigate the intricate labyrinth of a decentralized, interconnected future.

6 Bibliography

[1] Handan Kilinc Alper. *BABE*. Research at W3F. Oct. 8, 2021. URL: `https://research.web3.foundation/en/latest/polkadot/block-production/Babe.html` (visited on 11/28/2022).

[2] *Astar-Frame*. Astar Network, Feb. 6, 2023. URL: `https://github.com/AstarNetwork/astar-frame` (visited on 02/21/2023).

[3] cryptorank. *WBTC Wrapped Bitcoin Loses Peg*. Cryptorank News. Nov. 28, 2022. URL: `https://news.cryptorank.io/wbtc-wrapped-bitcoin-loses-peg/` (visited on 09/25/2023).

[4] Rosario Gennaro and Steven Goldfeder. *One Round Threshold ECDSA with Identifiable Abort.* 2020. URL: https://eprint.iacr.org/2020/540 (visited on 05/03/2023). preprint.

[5] Christopher Goes. *The Interblockchain Communication Protocol: An Overview.* June 29, 2020. DOI: 10.48550/arXiv.2006.15918. arXiv: arXiv:2006.15918. URL: http://arxiv.org/abs/2006.15918 (visited on 12/30/2022). preprint.

[6] Kewei Hou and Tobias J. Moskowitz. "Market Frictions, Price Delay, and the Cross-Section of Expected Returns". In: *The Review of Financial Studies* 18.3 (2005), pp. 981–1020. ISSN: 0893-9454. JSTOR: 3598084. URL: https://www.jstor.org/stable/3598084 (visited on 01/04/2023).

[7] Jun Kimura. *IBC-Solidity.* Version v0.2.4. Hyperledger Labs, Dec. 27, 2022. URL: https://github.com/hyperledger-labs/yui-ibc-solidity (visited on 12/29/2022).

[8] Gerrit Köchling, Janis Müller, and Peter N. Posch. "Price Delay and Market Frictions in Cryptocurrency Markets". In: *Economics Letters* 174 (Jan. 1, 2019), pp. 39–41. ISSN: 0165-1765. DOI: 10.1016/j.econlet.2018.10.025. URL: https://www.sciencedirect.com/science/article/pii/S0165176518304361 (visited on 01/04/2023).

[9] Chelsea Komlo and Ian Goldberg. "FROST: Flexible Round-Optimized Schnorr Threshold Signatures". In: *Selected Areas in Cryptography.* Ed. by Orr Dunkelman, Michael J. Jacobson, and Colin O'Flynn. Vol. 12804. Lecture Notes in Computer Science. Cham: Springer International Publishing, 2021, pp. 34–65. ISBN: 978-3-030-81651-3 978-3-030-81652-0. DOI: 10.1007/978-3-030-81652-0_2. URL: https://link.springer.com/10.1007/978-3-030-81652-0_2 (visited on 09/25/2023).

[10] Tobias Oberstein. *EIP-665: Add Precompiled Contract for Ed25519 Signature Verification.* Ethereum Improvement Proposals. Mar. 25, 2018. URL: https://eips.ethereum.org/EIPS/eip-665 (visited on 12/29/2022).

[11] Olaoluwa Osuntokun. *TAP: Taproot Assets Protocol.* GitHub. Dec. 10, 2021. URL: https://github.com/Roasbeef/bips/blob/b831047bcdcdf625ea41a53aec4303c12c0d6fc3 bip-tap.mediawiki (visited on 09/25/2023).

[12] C. P. Schnorr. "Efficient Identification and Signatures for Smart Cards". In: *Advances in Cryptology — CRYPTO' 89 Proceedings.* Ed. by Gilles Brassard. Vol. 435. Lecture Notes in Computer Science. New York, NY: Springer New York, 1990, pp. 239–252. ISBN: 978-0-387-97317-3. DOI: 10.1007/0-387-34805-0_22. URL: http://link.springer.com/10.1007/0-387-34805-0_22 (visited on 10/07/2023).

[13] Trevor Spiteri. *Fixed.* Version 1.22.0. Feb. 19, 2023. URL: https://gitlab.com/tspiteri/fixed (visited on 02/21/2023).

[14] Alistair Stewart and Eleftherios Kokoris-Kogia. *GRANDPA: A Byzantine Finality Gadget*. July 3, 2020. DOI: 10.48550/arXiv.2007.01560. arXiv: arXiv:2007.01560. URL: http://arxiv.org/abs/2007.01560 (visited on 11/28/2022). preprint.

[15] *Substrate IBC Pallet*. Version 3.0.0. Octopus Network, Dec. 12, 2022. URL: https://github.com/octopus-network/substrate-ibc (visited on 12/12/2022).

[16] Mikhail Vladimirov. *ABDK Libraries for Solidity*. ABDK, Feb. 21, 2023. URL: https://github.com/abdk-consulting/abdk-libraries-solidity (visited on 02/21/2023).

[17] Peter Wuille, Jonas Nick, and Tim Ruffing. *BIP: 340 – Schnorr Signatures for Secp256k1*. GitHub. Jan. 19, 2020. URL: https://github.com/bitcoin/bips/blob/7004ad1a825a0422b78bbf1a96bf748d5e380569/bip-0340.mediawiki#L3 (visited on 09/25/2023).

A Datasheet

A.1 Account Model

A private key represents an account. The account model supports the following private keys: Ed25519, Sr25519, and Secp256k1. In addition, Golden Gate offers multi-signature and proxy accounts that we will discuss in depth alongside a classic account.

The Golden Gates uses a Substrate-based SS58 format. The address consists of two parts:

- Prefix - determines how to validate the address and shows the address network.
- Address - address data

A.2 Account Balances

Each account consists of several balances that give the user relevant info about his activities and token usage.

We require the user to have at least one token to prove that the account is active. We suspend and remove accounts with a balance of less than one token.

Each account contains different types of balance depending on account activity:

Balance type	Description
Total	Tokens in the account. The balance does not represent available funds
Transferable	Tokens available for use
Vested	Tokens sent to the account. GoldenGate will release tokens after some verification time controlled in blocks
Bonded	Tokens locked for staking
Democracy	Tokens locked for governance activity
Redeemable	Tokens available for unlocking. These tokens have passed the lock period and are available for usage
Locked	Tokens frozen for on-chain activities. The locks do not stack, so Golden Gate suspends the biggest lock. The difference between the current lock and the next biggest became redeemable when Golden Gate unlocks funds
Reserved	Tokens locked and not relevant to governance, staking, or vesting

A.3 Proxy Accounts

Any account can create a proxy account with limited or full access to the main one. The proxy account can do transactions on behalf of the creator with a limitation or not.

This approach helps to limit transactions made by the primary account and keeps it more secure. The primary account can remove or change the proxy if the user cannot access it anymore. This helps to come up with more granular security practices.

The bare lock deposit for identity consists of two constants:

- The `ProxyDepositBase` is the default deposit for setting up proxy accounts.
- The `ProxyDepositFactor` is the deposit for each used proxy.

The Golden Gate will unlock funds once the user removes proxies.

Golden Gate supports the following proxy types:

Proxy type	Transaction types
Any	All. Gives all rights to the proxy account. It does not give any security benefits, so the user should avoid it
Non-transfer	All, except balance transfers and vested transfers
Governance	Governance-related
Staking	Staking-related
Identity judgement	Transactions for registrars to judge an account's identity
Auction	Transactions for participation in para-chain auctions and crowd loans
Time-delayed	Time-delayed transactions. The proxy will announce its intended action and wait for the number of blocks defined in the delay before executing it. Also, it will include the hash of the intended function call in the announcement. The user can cancel the intended action by primary account or cancel-proxy within this time window
Cancel	Transactions for accounts to reject and remove any time-delay proxy announcements

A.4 Multi-Signature Accounts

Golden Gate supports multi-signature accounts. The multi-signature account consists of one or more addresses and the threshold. The threshold defines how many approvals Golden Gate requires to sign a transaction.

The account should have a locked deposit to participate in the network. The deposit consists of two constants:

- The deposit base is the deposit for the usage of a multi-sig account.
- The threshold deposit is for each approval required to sign a transaction.

A.5 Transaction Model

In Golden Gate, we call Transactions/state changes that are included into the block extrinsic.

There are three different extrinsic transaction types:

- Signed transaction - must include the signature of an account sending it; signed user should pay an execution fee.
- Unsigned transaction
- Inherent transaction - a particular case of unsigned transaction. The creator node is the only one that can add information to a block. Most of the data inserted by this type of transaction are assumed to be valid without validation.

A.6 Block Structure

A Golden Gate block consists of a header and a body.

The block body contains a list of extrinsic transactions included in the block.

The block header contains the following data:

Field name	Description
Parent hash	32-byte Blake2b hash of the parent block
Block height	An integer representing a block. The genesis block is a 0
State root	Merkle tree root hash after applied transactions
Staking	Staking-related
Transaction root	Cryptographic digest of the transaction series
Digest	Any chain-specific auxiliary data. It contains consensus-related data, including the block signature

A.7 Consensus

Golden Gate is a substrate-based chain that leverages hybrid consensus. Block production is separated from finalization to achieve fast chain growth with secure finality. Fast block production is achieved using round robin, although Golden Gate may switch to probabilistic block production with `BABE`[1] in the future. For efficient and secure block finalization, we are using the `GRANDPA`[14] protocol. The key principle behind this design is to get security guarantees level similar to instant-finality consensus while having block production speed similar to probabilistic safety consensus.

Estimated consensus parameters for this approach:

Parameter	Value
Time to finalization	12-60s
Block production time	6s
Block size	< 5 MiB
Estimated TPS	1000