Imperial College London

MEng Individual Project

Imperial College London

Department of Computing

Exploring The Relationship Between Speed and Security of Cross-chain Transactions

Author:
Reece Jackson

Supervisor:
Dr. Arthur Gervais

Second Marker:
Dr. Mark Wheelhouse

June 20, 2022

Abstract

In recent years we have seen the monumental growth of capital investment in cryptocurrency projects, an overall growth in the number of projects being released and the growth of the applications that cryptocurrencies and blockchain provide us with. Yet , there is little to no accompanying strides in the connectivity. From a purely socioeconomic perspective, better efforts with the exchange of information, data and essentially value can create a more paralleled direction of growth for the blockchain and cryptocurrency rather than branching further and further away. An idea, perhaps trivial, is currently far from the conditions we require for mainstream usage. A big contributor to this is due to limitations of speed and security of the infrastructures we have in place today.

This paper formally attempts to quantifies security of cross-chain chain transactions under Proof-of-Work consensus and provides a speed metric comparison derived from the security mechanism aforementioned. Furthermore, in order to be comparable with the more recent paradigm shift to Proof-of-Stake consensus, utilising our cross-chain systematisation, we introduce the early stages for security quantification of PoS blockchains.

Our results indicate that there is in fact a positive correlation between the two to the quantifiers, since the transaction direction with the greater requirement of safe confirmations resulted in quicker speeds, partially due to network block propagation times.

Acknowledgements

I would like to take the chance to say a massive thank you to both my supervisor Dr. Arthur Gervais and second marker Dr. Mark Wheelhouse, my project wouldn't be the work it is today without their advice and guidance. I'd also look to extend my gratitude to Liyi Zhou and Kaihua Qin who would willingly take the time to answer and discuss any queries I had. Additionally, I want to thank my Mother, who sadly passed away this academic year. Mum, I hope this work makes you proud! Most of all I want to thank God because to be where I am today, given the difficulties and setbacks I've faced, is all thanks to him, Alhamdulillah.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 History

The birth of blockchain technology and cryptocurrency was first notably introduced in 2009, released under the pseudonym Satoshi Nakamoto, with a decentralised currency we know now as Bitcoin[1]. The currency is designed with a Proof-of-Work consensus (see more subsection 2.2.2) as its foundation, enabling peer-to-peer transaction validations to take place as opposed to traditional, centralised validation [1].

As the first-mover, Bitcoin paved the way for a new field of financial opportunities using blockchain technologies. In particular, Ethereum (see more subsection 2.2.1) with over $152 billion in TVL across 361 protocols [2] as well as its plethora of EVM sidechains (see more subsection 2.2.3). With the growing number of blockchains being developed year-on-year [3], communication between existing protocols is paramount.

1.2 Cross-chain and Interoperability

Cross-chain in essence, is the exchange of data, information and assets between two or more siloed blockchains. The key difficulty in fulfilling cross-chain ability with currently produced blockchains is that they were designed without a standardised inter-communication model in mind. Most chains have a specific use case and the limitations of one blockchain are the strengths of others. With cross-chain, the limitations can be alleviated as users can seamlessly switch between blockchains that deliver the best results for their usage [4].

For cryptocurrency's to go through widespread adoption, it is crucial that an interoperable system, similar to the centralised financial system, is produced. Economic agents such as consumers, businesses and governments require speed, security and ease before the notion of entering the space becomes enticing.

1.3 Motivation

The ultimate goal of any financial system or market is to solve the economic problem of efficiently allocating and distributing finite resources amongst the infinite wants. Cross-chain plays a big part in achieving this goal and therefore any research contributions to the field are valued.

On a more refined level, efforts to help understand and quantify security of cross-chain transactions has never been researched to such level, as far as I am aware. Furthermore, the majority of work that has gone into studying security of blockchains, simply at a base level, has primarily focused on proof-of-work blockchains, rather than the arguably prominent proof-of-stake consensus mechanism. To expand the usecases and mass-adoption of cryptocurrencies in wider society, security and speed are paramount to achieving this and we hope our contributions open up new opportunities for the DeFi space to grow.

1.4 Contributions

The primary focus of this paper will be to uncover the relationship between speed and security of cross-chain transactions and protocols.

This paper will make the following contributions.

1. **Nakamoto bridge security quantification and insights:** In order to provide the most representative study, we analyse bridging firstly with Nakamoto blockchains that make up 60.30% [5] of the cryptocurrency sector dominance. Studying the blockchains of Bitcoin and Litecoin, we provide the first formal security quantification methodology for cross-chain bridging between Nakamoto blockchains, extending work on secuirty quantification using stale blockrate [6].
2. **Identification of double-spend safe confirmations:** In this paper we identify the number of confirmations required for bridging to and from Nakamoto blockchains, in a safe manner against a double-spend adversarial attack strategy, whereby the transaction can be confidently, deemed as almost irreversible.
3. **Security and speed quantitative comparison:** We provide a methodology, extended from our security quantification results, for analysing the speed of a cross-chain transaction whereby we utilise the "safe" number of confirmations and block propagation times. This allows for numerical comparison between speed and security of cross-chain transactions for Nakamoto blockchains.
4. **Systematisation of PoS cross-chain protocols:** We produce the first systematisation of cross chain procedures for the 3 existing PoS blockchains of Cosmos, Multi and Wanchain. Moreover, we introduce early informal considerations of modelling PoS blockchains using MDP and the differences encountered against PoW blockchains.

Chapter 2

Background

This chapter will concentrate primarily on the knowledge required to understand points, arguments and results raised throughout the paper. Beginning with a top-down approach, going from more generalised concepts like what is a blockchain and drilling down into finer details such how consensus is reached. Lastly the chapter will close with an overview of the information regarding the framework we we later discuss for quantifying security.

2.1 DeFi

With the early success gained by Ethereum, it provided the avenue for many different and intricate applications to be deployed on the Ethereum blockchain. In recent years, we have seen them range from NFT projects such as Bored Ape Yacht club (current floor price 90 ETH approx $99k) [7] to Metaverse and gaming projects with Axie Infinity having a market cap of over a billion, making it a Top 50 coin in market cap [8]. However, one of the most prominent areas remains decentralised finance (DeFi). Take Uniswap, a decentralised exchange (DEX), for example, now facilitates $1bn+ in daily trades [9] along with its governance token UNI having all-time high market cap of $22.5bn [10].

2.1.1 Centralised Finance

Centralised or traditional finance is an umbrella term classifying investment vehicles whereby an intermediary is involved as part of the service to manage transactions and oversee usage. The Majority of the centralised investment activity occurs on centralised exchanges (CEX) such as Coinbase, Binance and FTX and the opportunities available span from buying and selling crypto, Lending, Borrowing, Margin trading etc.

CeFi provides its users with the benefit of seamlessly switching between Fiat and crypto and the added benefit of varying degrees of cross-chain support. However, some deem this is a pseudo-cross-chain solution as they are generated on individual blockchains rather than a transfer from one chain to another [11].

2.1.2 Decentralised Finance

Conversely, decentralised finance (DeFi) is finance with the absence of a middle man, that takes place using peer-to-peer financial services via permissionless blockchains. Thus, providing lower barriers to entry, total control and anonymity of your personal information as well as complete transparency of all transactions that take place.

Similarly to CeFi, there are multiple overlaps with the financial instruments that are available, such as lending, borrowing and purchasing etc. Although, blockchain transactions are subject to Gas which can become costly for active trading. Additionally, the free barriers to entry permit the freedom for high-risk, unregulated investments to occur. Furthermore, cross-chain on DEX is in its sunrise stage and rather underused, underdeveloped and convoluted, for the most part, currently [12].

2.2 Blockchain

A blockchain is a distributed ledger containing bundles of data chained together, individually known as blocks, but together represents the complete transaction history. Each block also contains a timestamp, Nonce value and the hash of the associated parent block [13].

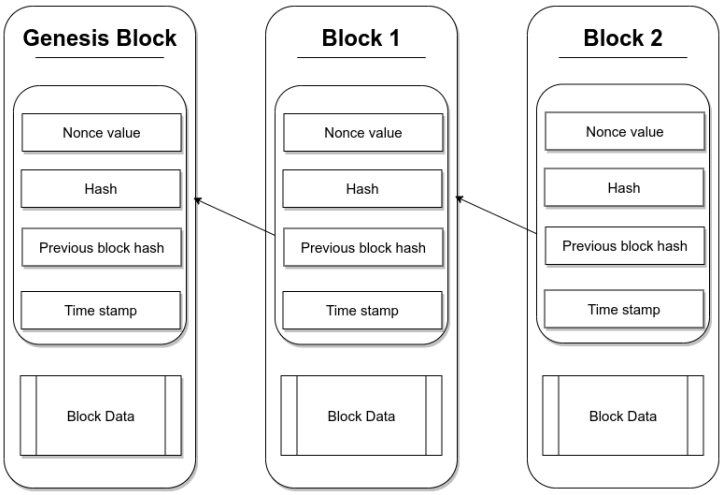

Figure 2.1: Blockchain Diagram

The block is then cryptographically validated and this model together asserts Blockchain integrity. If an individual wish to change the data within a block, then this would mean the entire block must be changed which would require agreement from the parent block and then in turn the parent's parent block etc. Only once a consensus agreement is reached, then a modification to the blockchain can be permitted [14].

2.2.1 Ethereum Blockchain

The Ethereum blockchain was first launched in July 2015, by founders Vitalik Buterin and Joe Lubin with the vision of becoming more than just a singular cryptocurrency [15]. The Ethereum platform, powered by the native Ether (ETH) token can support decentralised application code in the form of smart contracts with the aim of unifying multiple different applications [16].

2.2.2 Consensus Mechanisms

Blockchain validation is the process of concluding whether a transaction is legal on the blockchain. This happens via consensus of a certain number of nodes forming an agreement on a block of transactions being valid. Below are the most common consensus mechanisms that each blockchains of today uphold [17].

Proof-of-Work

This is a consensus mechanism that is the oldest of the four we will discuss. It is still used today by many coins including the two biggest market cap coins Bitcoin and Ethereum 1.0 and several other Nakamoto blockchains at the time of writing.

These Proof-of-Work (PoW) blockchains are secured by Miners to avoid malicious actions on the network, for instance, double-spending and fraudulent block creations and/or deletions. Miners race to solve maths problems to be rewarded the chance to update the blockchain with the latest verified transaction, in return, they receive a reimbursement in the native currency [18].

The strength of this model comes from the immense amount of computational power required

to overthrow the consensus mechanism (>50% of the mining network power). A benefit to this structure is that the longer the blockchain length becomes, the more difficult it is for users to behave maliciously [18].

Proof-of-Stake

With the expansion of the cryptospace, the PoW effectiveness has reached its ceiling in regards to the Ethereum network. Due to the number of various transactions on the network, scalability is an issue with PoW, resulting in spiking fees.

With this shortcoming in mind, other networks such as Cardano use a Proof-of-Stake (PoS) consensus mechanism. Rather than having miners in the system, instead, this model has validators, individuals who Stake the native token on a Node. The network randomly selects which validators have the privilege of ordering transactions and creating the next block. Validators are also subject to Slashings and, in more extreme cases, complete loss of stake for malpractice [18].

This validation paradigm doesn't eliminate the vulnerability of a 51% attack however, it does drastically disincentive one. Attackers would need to gain 51% of all the staked currency, which is an extremely costly affair, with most of the high market cap coins that implement PoS, such as the upcoming ETH 2.0. Thus, offers very little encouragement to hurt the value of a coin you own the majority of. But, on the other hand, this consensus mechanism is less energy-intensive and creates a more decentralisation process [19].

Proof-of-History

Founded in 2017, Solana became the first blockchain to implement this new consensus method known as Proof-of-History, PoS. Suppose you plant a tree and take pictures at various stages of its development. The picture of the seedling would naturally come before the picture of branches forming, right? This is similar to how PoS works. PoS uses a secure, recursive delay function to hash transactions and supplies each with a count, together representing a function of real-time. Each node in the network has a clock that helps the network validate the events, without needing to communicate with other nodes [20]. This gives an ordering of transactions in an extremely fast manner without compromising security as an attacker would need 2^{128} cores to crack the hash function [21].

Zero-knowledge Proof

Although perhaps the least used consensus mechanism in applications currently, zero-knowledge proof (ZPK) has gained a lot of traction in the blockchain space [22]. ZPK is a methodology to achieve consensus whereby the prover is able to persuade a verifier that a statement is true without revealing any additional knowledge about the statement. This is facilitated through probabilistic assessments as opposed to revealing complete information. Cumulatively, these assessments gradually increase our confidence that the prover is indeed telling the truth, making it more efficient than verifying with the entire blockchain [23]. An example to help convey this is consider the scenario where the prover has a safe and wishes to prove to the verifier that they know the combination. A way to confirm that the prover is telling the truth, is by the verifier locking the message in the safe. If the prover can unlock the safe and tell the verifier the message, it is enough information to be relatively confident that the prover knows the combination [22].

2.2.3 Ethereum Virtual Machine

The Ethereum blockchain can be viewed as a distributed state machine with the, Turing-complete, Ethereum Virtual Machine (EVM) as its core. The EVM determines the rule set for valid state transitions and is also liable for execution of contract bytecode. Every single node on the Ethereum blockchain contains an immutable instance of the EVM [24, 25].

2.2.4 Smart Contracts

A smart contract is a written program that runs on the Ethereum blockchain. They declare an agreement between two parties and are self-executing, thus automating the agreement process

such that it is needless of third party involvement or delay. It delivers a trustful, yet completely anonymous transaction, between parties anywhere in the world [26].

2.2.5 Secure Multi-Party Computation (SMPC/MPC)

Secure multi-party computation is a cryptographic paradigm whereby a number of distinct, yet connected, parties carry out a joint computation in order to strengthen security against adversarial behaviour from individuals. Thus, it removes a single point of failure whilst maintaining privacy and correctness [27].

2.2.6 Threshold Signature Scheme (TSS)

A Threshold signature scheme is a schema consisting of 3 algorithms: KeyGen, Sign and Verify [28].

- The KeyGen algorithm outputs a public/private key pair. These are then passed to the other two algorithms. In this methodology, the basis is reliant on the principle of consensus and agreement between multiple parties. The algorithm involves a set of n parties who interactively generate an m-out-of-n the key i.e. like a jigsaw, whereby you need all m parts to formulate the picture. Thus no subset of fewer than m parties has any information about the key [28].
- The Sign algorithm receives the private key and uses this as well as a message to generate a signature. Similarly, this is generated interactively, whereby if and only if m-out-of-n parties agree on the message to sign, only then they are able to generate a signature [28].
- The Verify algorithm receives the public key, message, and signature, and performs verification of the signature [28].

2.3 Cross-chain Projects

Throughout this paper we will explore different DeFi cross-chain solutions. We will focus primarily on the following 5 protocols:

2.3.1 Polkadot (DOT)

The Polkadot blockchain is a blockchain with the vision of being a scalable, interoperable secure network protocol for Web3. Originally founded by former Ethereum CTO and co-founder Gavin Wood, technical director of Web3 Foundation, Peter Czaban and long-serving Rust community member, Robert Habermeier [29].

The DOT system allows for any arbitrary data to be transferred across different Parachain blockchains, termed as being a 'true multi-chain application environment'. Users will have the ability to transfer this data across public, private, Permissioned and Permissionless parachains [30].

Unlike Ethereum which uses Proof-of-Work and soon Proof-of-Stake, Polkadot uses a GRANDPA (GHOST-based Recursive Ancestor Deriving Prefix Agreement) consensus mechanism. Under positive network conditions, GRANDPA can finalise blocks nearly instantly, whereas, even under a Proof-of-Stake consensus, there is still the limitation of different blockchains competing to gather validators for their respective network. This hinders most projects security in the short run, yet the blockchain's security is aggregated across the Polkadot blockchain [30].

2.3.2 Cosmos (ATOM)

The Cosmos blockchain (ATOM) is a network of independent, parallel chains, whereby each uses the Tendermint consensus algorithm. The Tendermint consensus algorithm under-the-hood is a Byzantine Fault-Tolerance consensus mechanism. Based on the Byzantine Generals Problem, the fault-tolerance mechanism takes into account that inevitably, some nodes in the network will go down, misbehaviour, or even act maliciously, but as long as 2/3 of the network is still functioning,

the algorithm enables the network to still function [31].

The ATOM developers envisaged a blockchain that mimics the interconnectivity of the internet. Cosmos allows native blockchains to keep their sovereignty and uniqueness and still communicate with other blockchains in the ecosystem [32].

2.3.3 Blocknet (BLOCK)

Blocknet (BLOCK) is an open-source, layer two protocol designed, much like ATOM, to connect blockchains together into one network, similarly to how the internet connects computers. The aim is to integrate and synergise the utility of the multiple, different use cases each blockchain currently provides. This is achieved through the XRouter - layer for connecting DApps and the XBridge - layer for connecting siloed blockchains [33].

The Blocknet network, much like Cardano, is powered by nodes in a Proof-of-Stake consensus mechanism and it is self-funded as well as community governed. BLOCK operates as a Decentralised Autonomous Organisation whereby node operators, token holders and stakers have a say in the decision-making process for the protocol [33].

Furthermore, Blocknet provides Block DX, a trading DEX, allowing for free, permissionless listing and was officially the first DApp built on the Blocknet Protocol [33].

2.3.4 Wanchain (WAN)

Wanchain's mission is to promote blockchain interoperability through the use of fully decentralised bridges connecting various siloed blockchains. Last year (2021), they managed to launch the "world's 1st decentralised BTC-ETH direct bridge" [34]. To date, they now have bridges for Binance Smart Chain (BSC), Avalanche C-Chain (AVAX), Moonriver (MOVR), Dogecoin, Polkadot (DOT), Litecoin (LTC), Fantom (FTM), Arbitrum, Polygon (MATIC) and Ripple Ledger (XRP) [35].

Wanchain is validated through a modified version of PoS called Galaxy consensus. Prospective WAN node validators must register to participate in Galaxy consensus, with the idea being that registry is proportional to activity rate, overall improving network stability. Embedded into Galaxy is a triple ECDSA proxy signature scheme as a delegation policy, which intends to create the fairest possible environment so that even users with smaller amounts of WAN, can still participate. Moreover, the WAN team have developed a secure random number generation algorithm to support block proposer selection, introducing the notion of entropy, whilst asserting high network security [36].

2.3.5 Multichain (MULTI)

Multichain, formerly known as Anyswap (ANY), frames itself as being "the ultimate Router for Web3. Anyswap was founded in July 2020 with the view that, for the benefits of blockchain technology and services to reach wider consumers, the crypto scene would need a fast, secure, cost-efficient way to exchange and communicate value and information. Above all, intercommunication between the multiple, diverse blockchains. Through Multichain solutions, blockchain interoperability was accomplished for Ethereum like chains such as BSC, some layer 2 protocols like MATIC, a selection of parachain networks like Moonbeam (GLMR), a few Bitcoin comparable networks, LTC for example or even ATOM chains for instance Terra (LUNA) [37].

Mulichain uses a threshold distributed algorithm based on secure multi-part computation (SMPC). The multi-part computation network processes cross-chain requests in real-time between different chains. Node consensus is achieved when every node, independently, verifies the status of the original chain, communicated through the threshold distributed signature algorithm [38].

2.4 Reinforcement Learning

Reinforcement learning is a branch of machine learning that focuses on the sequential decision making process an agent make dependent on a given reward. It is the agent's primary objective to maximise their reward with each action. In this framework, learning is carried out through previous state experience as opposed to initial inputs like other machine learning algorithms.

2.4.1 Markov Decision Process (MDP)

An MDP is a methodology for simulating sequential decision-making processes. They are formed of 5 parts and work as the backbone for solving reinforcement learning problems.

2.4.2 Agents and Environment

In an MDP we have a decision-maker, formally known as an agent. The agent is placed and interacts within the simulation environment. These interactions are sequential.

2.4.3 States and Actions

At each time step, the agent will receive a snapshot of the environments state. Given this snapshot, the agent then makes the informed decision about the most appropriate action to next take. This is based on the action that yields to highest reward [39].

2.4.4 Reward

After an action is executed, we see the environment transition to a new state and the agent receives a corresponding reward. The goal of the agent is to maximise the cumulative reward of his/her actions, rather than the immediate reward per state [39].

Suppose we let the set of states be S, the set of actions be A and the set of rewards be R. At each time step $t = 0, 1, 2...$, the agent receives a snapshot of the environment's state $S_t \in S$. Based on this state S_t, the agent selects an action $A_t \in A$. This gives us a state-action pair (S_t, A_t). As we transition from time step t to time step $t + 1$, the new state is $S_{t+1} \in S$ and the agent receives reward $R_{t+1} \in R$ from taking action A_t in state S_t [39].

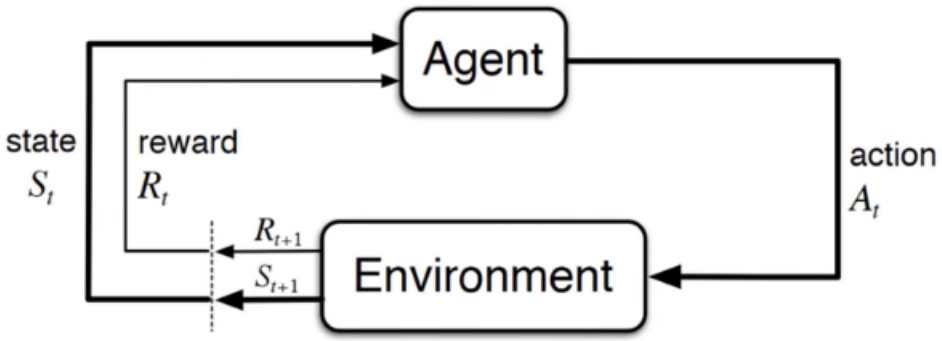

Figure 2.2: MDP Relationship Model [39]

2.4.5 Policy

The notion of a policy is the probability that an agent will select a specific action from a given state. If an agent follows policy π at time-step t, then $\pi(a|s)$ is the probability that the agent's action $A_t = a$ given the state $S_t = s$. For each state $s \in S$, π is the probability distribution over $a \in A(s)$ [40].

2.4.6 Value functions

As we are aware, the primary goal of the agent is to maximise the cumulative expected sum of rewards thus, in a finite setting of T time-steps [41]:

$$G_t = R_{t+1} + R_{t+2} + R_{t+3} + R_{t+4} + ... + R_T \tag{2.1}$$

However, not all problems are a finite task under continuous settings. We need to include the concept of discounting to our maximisation objective. The discount factor illustrates the present value of future rewards [41].

Thus agent's focus now shifts to selecting the action A_t to maximise the cumulative expected discounted return [41]:

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \gamma^3 R_{t+4} + ... \tag{2.2}$$

$$= \sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \text{ where } \gamma \text{ is the discount rate and } \gamma \in [1,0] \tag{2.3}$$

Value functions work as descriptors to help us quantitatively measure how good a specific action or state is for the agent within the environment [40].

We have two types of value functions:

- State-value function (one arg)
- Action-value function (two args)

The state-value function for policy π, denoted as v_π, informs us how good any given state is for an agent following policy π, i.e. the value of the state under π [40].

The value of state s under policy π is the expected return from beginning at state s at time t and continuing to follow policy π [40]:

$$v_\pi(s) = E[G_t \mid S_t = s] \tag{2.4}$$

$$= E[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \mid S_t = s] \tag{2.5}$$

Similarly, the action-value function for a policy π, denoted by Q_π, informs us how good any given action is from a given state for the agent at hand, following policy π, i.e. the value of an action under π [40].

The value of action a in state s under policy π is the expected return from beginning at state s at time t, taking action a and continuing to follow policy π [40]:

$$q_\pi(s,a) = E[G_t \mid S_t = s,\ A_t = a] \tag{2.6}$$

$$= E[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \mid S_t = s,\ A_t = a] \tag{2.7}$$

2.4.7 State-value and Action-value Optimality

With the ability to express the value of different states and actions under a certain policy π, we can reformulate the goal to look for the optimal policy leading to the greatest return.

A policy π is considered to be better or equal to the policy π' if and only if the expected return of π is greater than or equal to the expected return π' for all states i.e [42].

$$\pi \geq \pi' \text{ iff } v_\pi(s) \geq v_{\pi'}(s) \text{ for all s} \in \text{S [Optimal policy]} \tag{2.8}$$

The optimal policy will have an associated optimal state-value function, denoted by v_*, and defined as [42]:

$$v_*(s) = \arg\max_\pi v_\pi(s) \text{ for all } s \in S \tag{2.9}$$

The optimal state-value function provides us with the largest expected return achievable by any policy in the state space.

Similarly, the optimal policy has an associated optimal action-value function, denoted by $q_*(s)$, and defined as [42]:

$$q_*(s,a) = \arg\max_\pi q_\pi(s,a) \text{ for all } s \in S,\ a \in A(s) \tag{2.10}$$

The optimal action-value function provides us with the greatest expected return achievable by any policy π for each possible state-action permutation [42].

2.4.8 Bellman Optimality Equation

A fundamental property of q_* is that it must satisfy the following equation, known as the Bellman equation [42]:

$$q_*(s,a) = E[R_{t+1} + \gamma \arg\max_{a'} q_*(s',a')] \tag{2.11}$$

We use the bellman equation to find the optimal q function. Once we have the optimal q function, we can in turn find the optimal policy. This is found through using a reinforcement learning algorithm to identify the action a that maximises the output of $q_*(s,a)$ for that given state [42].

The Bellman optimality equation declares that for any state-action pair (s,a) at time t, the expected return from starting in state s, selecting action a and continuing to follow the optimal policy is the same as the expected reward we get from taking action a, in state s (yielding reward R_{t+1}), plus the maximum expected discounted return that can be achieved from any possible next state-action pair (s',a') [42].

Conclusion

We have now looked at the cryptocurrency space from a top down approach, starting with DeFi, transitioning to blockchain structures and consensus mechanisms and finally, ending with a framework that utilises reinforcement practices that later reoccur. With this information, we now present how we will combine this knowledge with cross-chain structures to achieve our main objective of quantifying cross-chain security. This will test the relationship with cross-chain transaction speed.

Chapter 3

Systematisation of Cross-chain and Bridging Protocols

3.1 Cross-chain Protocol Models

In this chapter we discuss in greater technical detail the methodologies deployed by 3 different projects in order to facilitate successful, cross chain transactions. Moreover, towards the end of the chapter we showcase an analytical breakdown of the consensus mechanism employed by the 3 cross-chain protocols.

3.1.1 Multichain Cross-chain mechanism

Multichain utilises SMPC nodes to collectively sign transactions to assist cross-chain services such as the MULTI Bridge and the MULTI Router.

MULTI Bridge

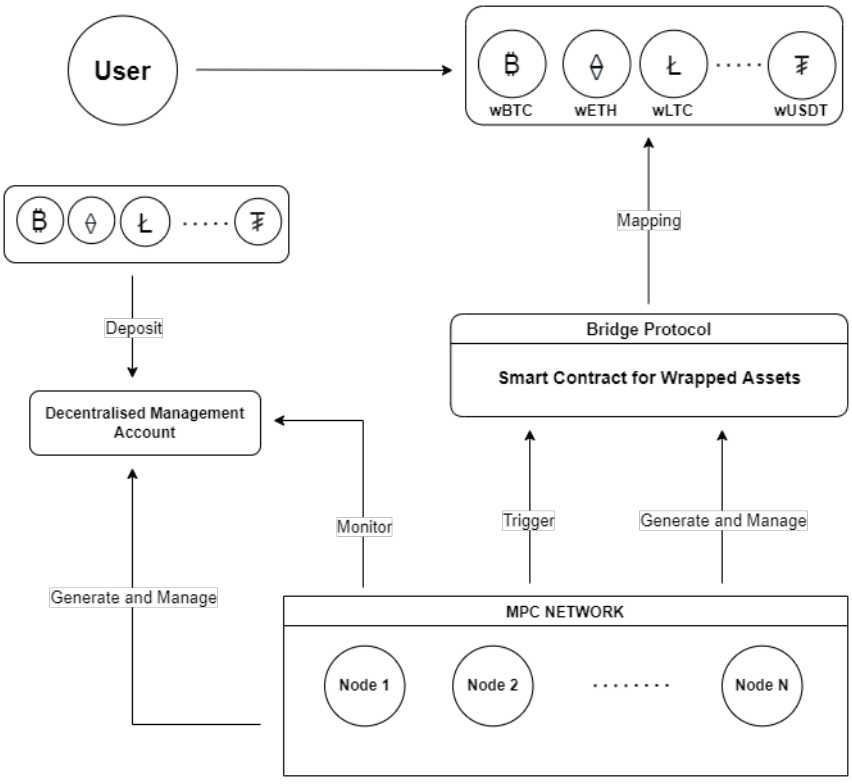

Figure 3.1: Multichain Deposit Diagram [43]

The MULTI Bridge links together two blockchains. Upon bridge creation, the SMPC nodes generate a Decentralised Management Account. This is a secured address used to send assets to when a user wishes to bridge assets from the origin chain, completely controlled by SMPC nodes only [43].

The SMPC nodes monitor this account and are notified when a new asset arrives. This in turn triggers the Wrapped Asset smart contract on the destination chain to mint 1:1 wrapped tokens pegged to the original token value.

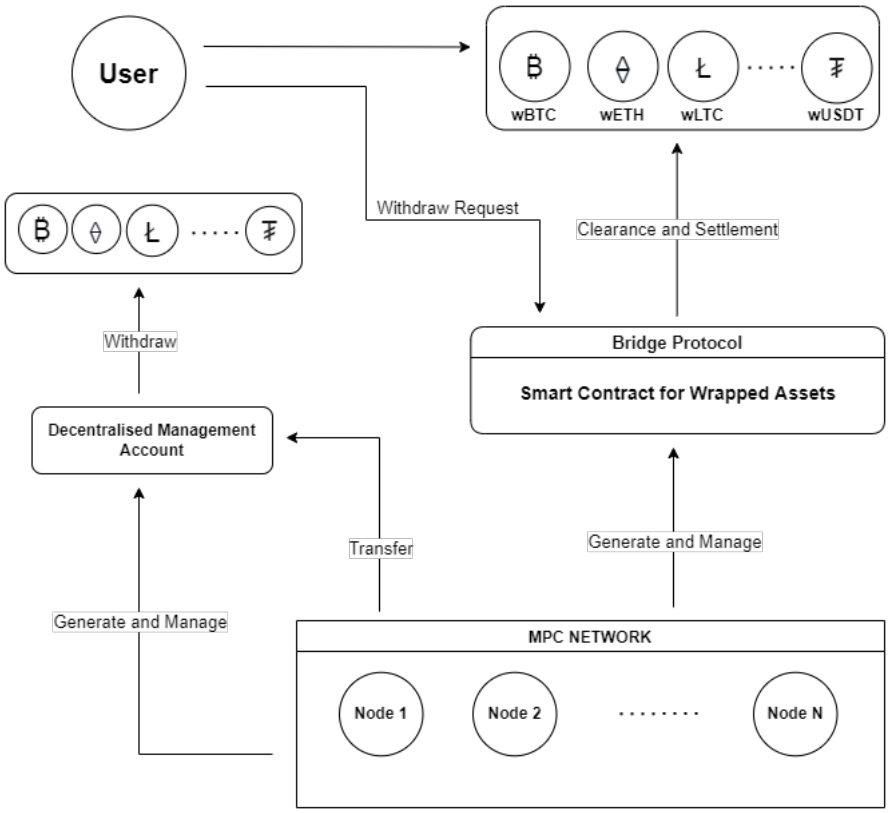

Figure 3.2: Multichain Withdraw Diagram [43]

In the reverse case, supposing the user wishes to move back to the origin chain, the Wrapped Asset smart contract is triggered by the SMPC nodes to burn the tokens. The assets from the Decentralised Management Account are then released by the SMPC nodes, then sent to the user [43].

MULTI Router

However, bridging is only an applicable option for some coins but not all, for instance USDC is a token that exists in its native form on multiple blockchains [43].

Therefore, for native coins and hybrid native coins, Multichain uses a different approach for moving assets between chains. This is carried out through the use of the MULTI Router and liquidity pools.

A liquidity pool is a collection of token funds, sourced from either the team, individuals or Multichain, locked in a smart contract. Thus providing users a means to access tokens when they move between different chains.

Suppose a user wishes to move N lots of token Z from chain α to chain β. The user's current

N lots of Z tokens are withdrawn to the liquidity pool for said tokens on chain α (now available for any users who wish for token Z on chain α). Now N lots of $anyZ$ are minted on chain α. As a result, N lots of $anyZ$ are also minted on the destination chain, chain β. This is done for completeness, to identify the number of X tokens the user has forgone on chain α and receive on chain β. The $anyZ$ on chain α is then burned to confirm finality of this stage. Provided that the liquidity pool for token Z on chain β is greater than the N lots of $anyZ$ we now have on chain β, this is automatically converted to token Z and withdrawn to the users wallet on chain β with the $anyZ$ on chain β burnt as well. In the event a shortage occurs, the user keeps the N lots of $anyZ$ and this represents the proportion of token Z needing to be redeemed later when liquidity is available [44].

Furthermore, some cryptocurrencies are hybrid native/bridged assets such as FTM, whereby they live native on some chains but are bridged on others. Thus if a user wishes to move this type of asset across to the native chains, then liquidity pools are required, if it's bridged to other chains then the 1:1 wrapped assets are utilised [44].

3.1.2 Cosmos Cross-chain mechanism

The cosmos network consists of many independent blockchains, known as zones with each zone powered by Tendermint consensus. The first zone on Cosmos is called the Cosmos Hub with the primary token of the Cosmos Hub being ATOM [45, 46].

The hub and zones of the Cosmos network communicate with each other via an inter-blockchain communication (IBC) protocol.

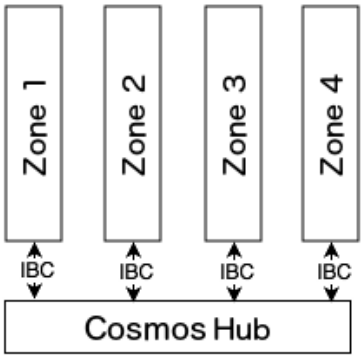

Figure 3.3: Hub-IBC-Zone relationship [46]

The IBC encompasses a constant stream of recent block commits posted from other zones to the hub, enabling the hub to remain informed about the state of each zone. Likewise, each zone is able to keep up with the state of the hub, although zones do not keep up to date with each directly. Instead, packets of information are then communicated from one zone to another by posting Merkle-proofs as evidence that the information was sent and received [46].

Tokens can be transferred from one zone to another in a special IBC packet called a "coin packet". The Cosmos hub is responsible for preserving the global invariance of the total amount of each token across the zones, whilst isolating each zone from the failure of another [46].

Cosmos Network communication

To understand the bridging process, we would need to understand how information is exchanged between different blockchain zones. Consider three blockchains, "Zone1", "Zone2", and "Hub", and we wish for "Zone1" to produce a packet destined for "Zone2" going through "Hub" [46].

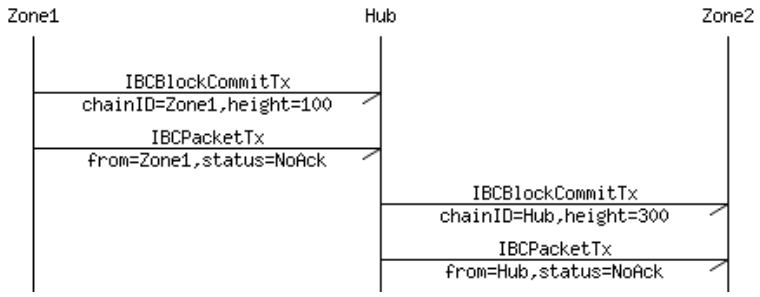

Figure 3.4: Blockcommit and packet communication [46]

In order to transfer a packet of data from one blockchain to another, a proof is submitted, stating that the sending chain has published a packet for the destination chain. The IBC protocol utilises two types of transactions: an *IBCBlockCommitTx* transaction and an *IBCPacketTx* [46].

The *IBCBlockCommitTx* transaction allows for a zone to prove and present its most recent block-hash to any other zone or observer [46].

The *IBCPacketTx* transaction allows for a zone to prove to any other zone or observer that the given packet was indeed published by the sender's application, via a Merkle-proof to the recent block-hash [46].

Bridge-zones

The Cosmos network also contains these special zones called "bridge-zones". "Bridge-zones" that are within the Cosmos network remain informed with ongoing cryptocurrencies also including the Hub itself. Each validator of a bridge-zone runs a Tendermint-powered blockchain as well as a full-node of the "origin" blockchain [46].

The bridge-zone validators come to an agreement on committed blocks when new blocks are mined on the origin blockchain. They will then sign and share their respective local view [46].

Once sufficient confirmations are agreed, the bridge-zone can confirm a receipt of payment on the origin which will create a corresponding account with that balance on the bridge-zone [46].

On the origin chain, token holders can send their tokens to the bridge-zone via the bridge-contract. However, tokens cannot be withdrawn once it has been received by the bridge-contract unless an appropriate IBC packet is received by the bridge-contract from the bridge-zone [46].

Tokens on the bridge-zone can be transferred to and from the Hub. To prove that the transaction occurred on the bridge-zone, an IBC packet is submitted to the chain's bridge-contract allowing for the token to be withdrawn [46].

IBC Transfer-In and Transfer-Out Process

The revolutionary IBC protocol used by Cosmos has the unique property of providing an unprecedented level of blockchain interoperability. The IBC protocol can be added or incorporated into almost any PoS blockchain. Once a blockchain has enabled and adhered to the IBC protocol guidelines, this opens a pathway for the blockchain to transfer assets and information between any other IBC-enabled blockchains. This special property of the IBC mechanism makes it difficult to up-scale such a methodology, nonetheless we can consider the IBC mechanism as follows:
https://interchainacademy.cosmos.network/academy/ibc/token-transfer.html

Given two blockchains A B, both IBC-enabled PoS chains. For a user that wishes to transfer tokens from blockchain A to B, the methodology begins with first understanding whether the tokens are currently on the source chain (origin) or sink chain (non-origin) [47].

If the token was on the source chain, funds are sent and locked in an escrow account, the receiver will then mint tokens and attach the prefix $\{Port\}/\{Channel\}/\{denom\}$ [47] with the port likely to be *transfer*, *channel* illustrating the IBC connections between blockchains that the token as travelled through, and finally the *denom* indicating the token that you wish to transfer. Thus *transfer/channel-40/atoms* defines the transfer of the Atom token via *channel-40* minted on the destination chain. The transfer of the tokens can continue to many more blockchains other than A and B, by repeating the methodology and modifying the *channel* to accommodate for the journey the token has taken [47].

Now suppose the case whereby the token wasn't originally on the source chain, but instead is being sent from a sink chain to a source chain. Since the transaction is going back to the source chain, the prefix is removed and the source chain will un-escrow the tokens releasing the funds and the sink chain will burn the tokens that previously contained the prefix [47].

3.1.3 Wanchain Cross-chain mechanism

WAN Bridge provides an foundation for the transference of assets between different blockchains, despite the lack of standardised blockchain infrastructure [48].

Wanchain's cross-chain approach is to set up one or more decentralised WAN Bridges in order to connect blockchain pairs. Each WAN Bridge has a Storeman Group consisting of 21 Storeman Nodes [48]. Storeman nodes are used to verify crosschain transactions with nodes that work together securely using MPC and TSS to withhold revealing any private keys. Storeman Node operators must pledge a certain amount of collateral to be put up as stake [49].

To better understand a two-way bridge, we should first learn about one-way bridges. As a consequence of different blockchains having different consensus mechanisms and structures, not all cross chain asset-transfers are supported. A one-way bridge is a bridge that supports the transfer of a native asset in one direction. BTC for example is relatively simple to transfer to the Ethereum blockchain, however it is rather complex to transfer ETH from Ethereum blockchain to the Bitcoin blockchain due to lack of smart contract support [49].

Similarly to the structuring of Multichain, there are two types of assets, native, i.e. generated from its source blockchain, and transformed/wrapped i.e. minted after locking assets in its parent blockchain [49].

Bridging-in

Since the two-way bridge is composed of two one-way bridges each way, we take a look at how a one-way bridge is formulated [49].

To illustrate the transfer-in process, consider the scenarios whereby:

- ➢ Assume Alice has an account on Ethereum
- ➢ Assume Bob has an account on Wanchain
- ➢ Alice wishes to transfer 10 ETH to Bob who's on Wanchain

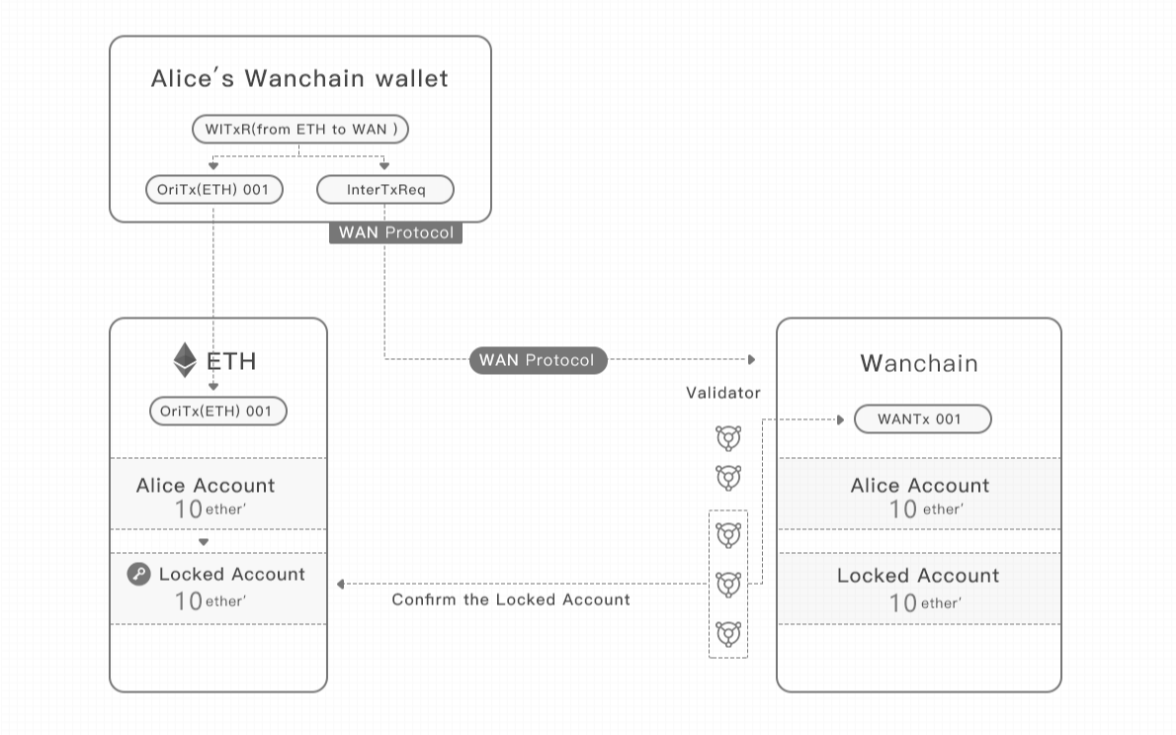

Figure 3.5: Transfer-in High-Level Diagram [49]

Step 1:

To send the asset OriAssetID to Wanchain's locked account, Alice would need to trigger a transaction OriTx (in this case ETH) with her OriAccount on the original chain (Ethereum). This will broadcast the crosss-chain transaction request InterTxReq to Wanchain [49].

Step 2:

Confirmation of the transaction on the original chain occurs via the Token Locked Flag (TLF) to indicate when the cross-chain transaction node, i.e. Voucher, has received the InterTxReq request [49].

Step 3:

OriTx is confirmed once the Voucher collects consensus on the TLF results with TLF=true being confirmation [49].

Then TLF=true result is received by the Wanchain's validator node. For uniqueness purposes, this validator checks whether the asset to be transferred is registered with Wanchain [49].

New assets will be registered and added into the registry. The public WANAccount issues a transaction WANTx and deploys a smart contract for the new assets. Distribution of tokens of value will be in the smart contract. Whereas for a registered asset, the transaction WANTx distributes tokens of Value for OriAccount directly into the existing asset contract [49].

Alice will receive the successful receipt of the cross-chain transaction by the Validator once WANTx is confirmed. In the event that this isn't confirmed, the storeman node will trigger a transaction on the original chain to transfer Alice's locked asset back to her account OriAccount [49].

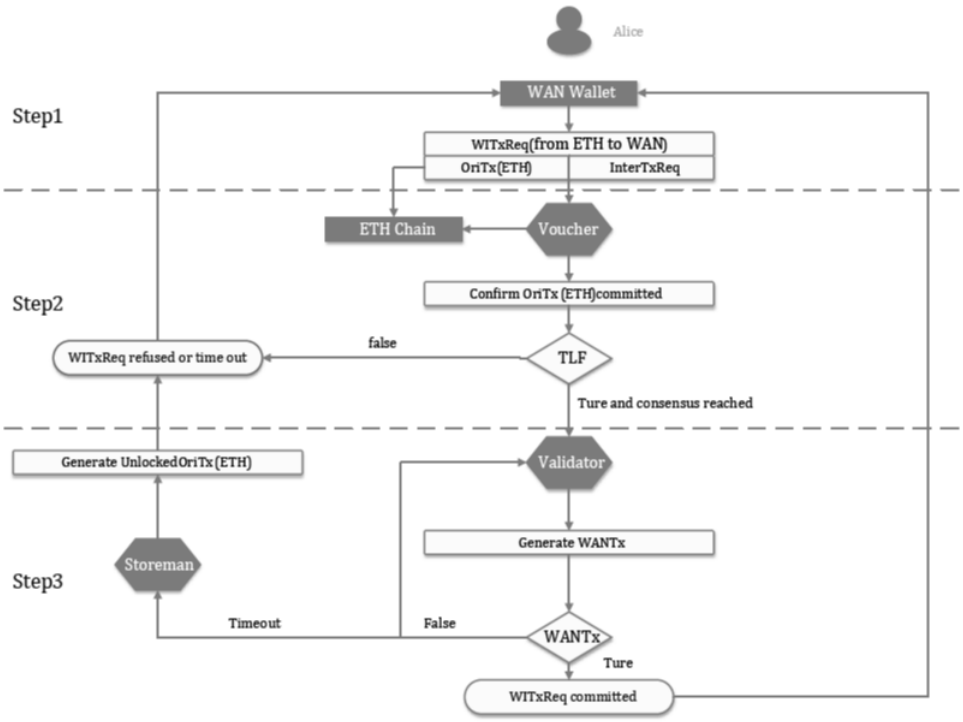

Figure 3.6: Transfer-in Low-Level Diagram [49]

Bridging-out

To illustrate the transfer-out process, consider the scenarios whereby [49]:

- ➢ Again assume Bob has an account on Wanchain
- ➢ Assume Chris has an account on Ethereum
- ➢ Bob wishes to send the 10 ETH he received from Alice to Chris

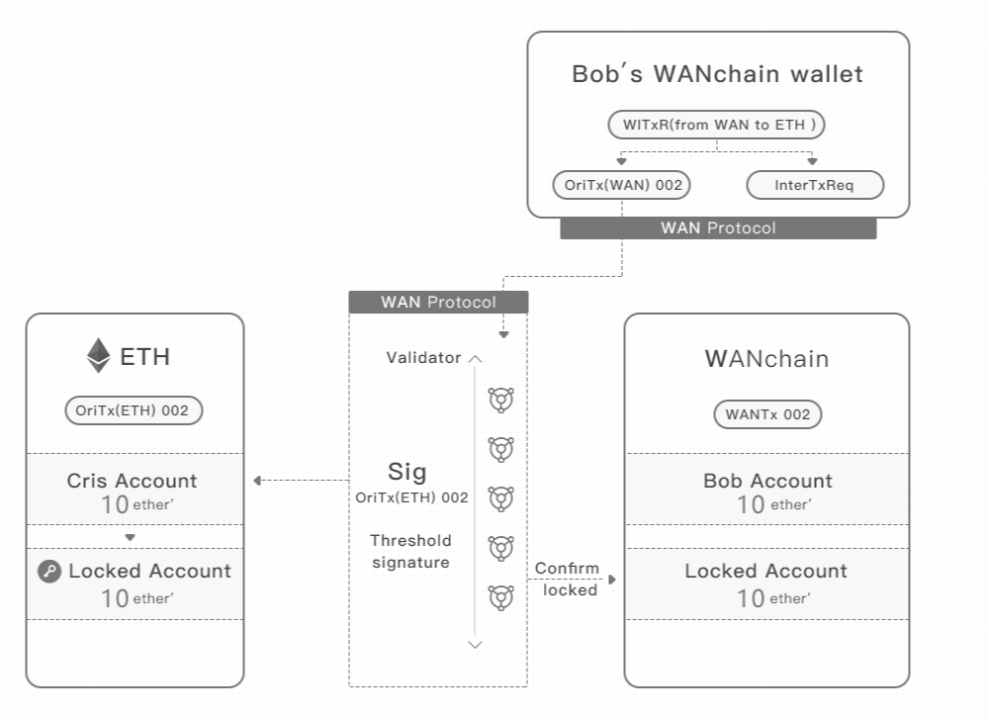

Figure 3.7: Transfer-out High-level Diagram [49]

Step 1:

Another InterTXReq request is produced subsequent to Bob using his Wanchain account to send the transaction WANTx (in our case ETH) to Chris on Ethereum. This is because it initiates the asset transfer-back function of the smart contract corresponding to the asset of the original chain [49].

Step 2:

Verification of this request triggers the Validator to call the smart contract. The Voucher checks the results of the contract execution and collects consensus. The Storeman node broadcasts the request of the transaction of the now Locked account if TFL=true. The confirmation of this transfer is additionally checked by the Voucher, gathering consensus on the confirmation flag TLF similar to before [49].

Step 3:

If the Token Unlocked Flag (TUF)=false the Storeman nodes will re-initiate the transaction, indicating that the transaction was not confirmed on the original chain. However, if true, the Validator will clear the assets locked in the smart contract on Wanchain [49].

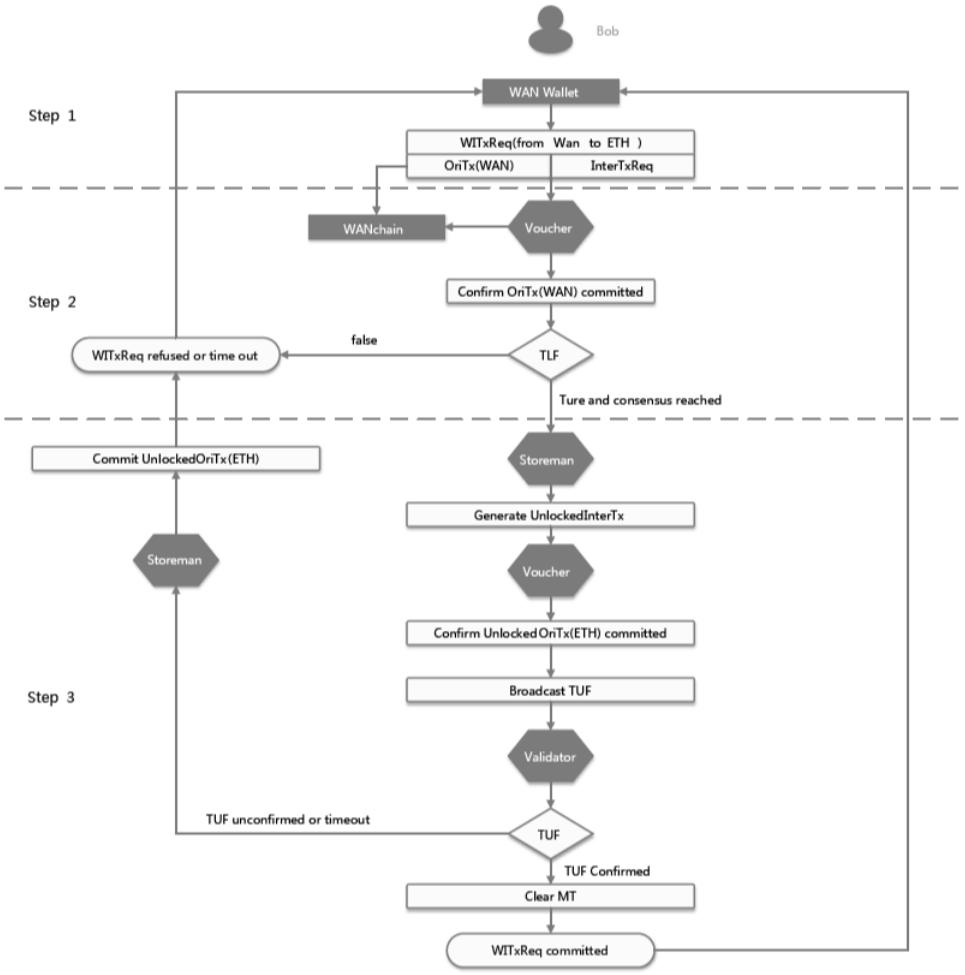

Figure 3.8: Transfer-out Low-level Diagram [49]

3.2 Systematisation Of Cross-chain Mechanisms

From the 3 studied protocols we observe that the main cross chain methodology adopted by almost all the protocols is via locked asset accounts mechanism. We formalise main existing mechanism in the following subsection.

3.2.1 Asset-locking mechanism

A high-level abstraction of the asset-locking cross-chain mechanism observes the following structure:

Transfer-in

1. User, wishing to move tokens from source chain to the destination chain, sends the tokens to an asset account
2. Nodes verify the receipt of the tokens from the user on the source chain and lock the account
3. Minting of wrapped assets with 1:1 value on the destination chain takes place
4. User receives the equivalent wrapped assets now on the destination chain

Transfer-out

1. User, wishing to move tokens back to the original chain, triggers a transfer back request from the smart contract

2. Once confirmed, the wrapped tokens are burned

3. The assets in the locked accounts are unlocked and distributed corresponding to the value of tokens burned

As we have identified, structurally, there may be differences in implementation, naming convention and terminology each project uses, despite that, the over-arching design for achieving cross-chain ability is relatively identical.

3.3 Studied Cross-chain Protocols Consensus Mechanisms

Finally, now we will examine specifically how each of the previously discussed protocols achieve consensus amongst block generation to enable network usage.

3.3.1 Multichain - Hierarchical Hybrid Consensus Mechanism (HHCM)[†]

The HHCM framework is reflected in the fact that the transaction packaging and generation of blocks are split into two phases; a hierarchy that is one after another [51].

The first layer implements the application's execution and submits the results to the second layer of the hierarchy. Virtual groups of physical nodes make up this layer. Each group of nodes will jointly process one of all the transactions assigned to the group. This grouping is established via a function $f(\alpha, \beta) \pmod X$, with α being the value of the previous block's hash and β being the public address and X the number of groups [51].

The block generation layer is the second layer. The nodes are mapped from one of the virtual groups in the first layer and they pack all the results submitted by the first layer into a new block record on the chain [51].

The hybrid aspect of the HHCM framework is portrayed in the different consensus methods used within each section. Across the first tier, a PoS Consensus is used but PoW is used to generate the final block in the second layer [51].

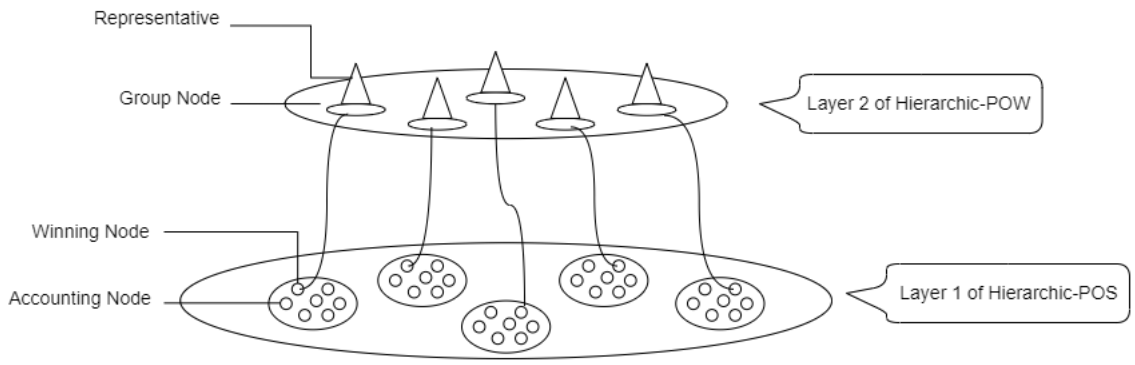

Figure 3.9: HHCM Diagram [51]

[†]Anyswap's brand upgrade to Multichain only involves changes to the brand name and relative aspects. It does not involve any change in the operating entity and will not have any impact on the services and products currently in use [50].

3.3.2 Cosmos - Tendermint Consensus Mechanism

Tendermint is a Byzantine Fault Tolerant (BFT), deterministic and mostly asynchronous, consensus protocol [52].

Validators are participants in the protocols that alternate in proposing blocks of transactions and voting on them. These blocks are committed on a chain, with one block at each height. The protocol however will move to the next round if a block fails to commit. In this case, a new validator obtains the proposal of a block for that height. There are two stages of voting required to successfully commit a block called pre-vote and pre-commit. More than $\frac{2}{3}$ of validators must pre-commit for the same block in the same round for a block to be committed [52].

A polka is when more than two-thirds of the validators pre-vote for the same block. In the same round, justification by a polka is required for every pre-commit [52].

Failure to commit a block by validators occurs for numerous reasons. One is that validators vote to move to the next round without receiving a complete proposal block from the proposer. This reliance on a timeout is what makes Tendermint a weakly synchronous protocol, rather than an asynchronous one [52].

Once a validator pre-commits a block, it is locked on that block and is required to pre-vote for that block in which it is locked on. It can only unlock and pre-commit for a new block if there is a polka for that block in a later round. These locking rules are introduced by Tendermint which guarantees that validators do not violate safety, that is, they will never commit conflicting blocks at the same height [52].

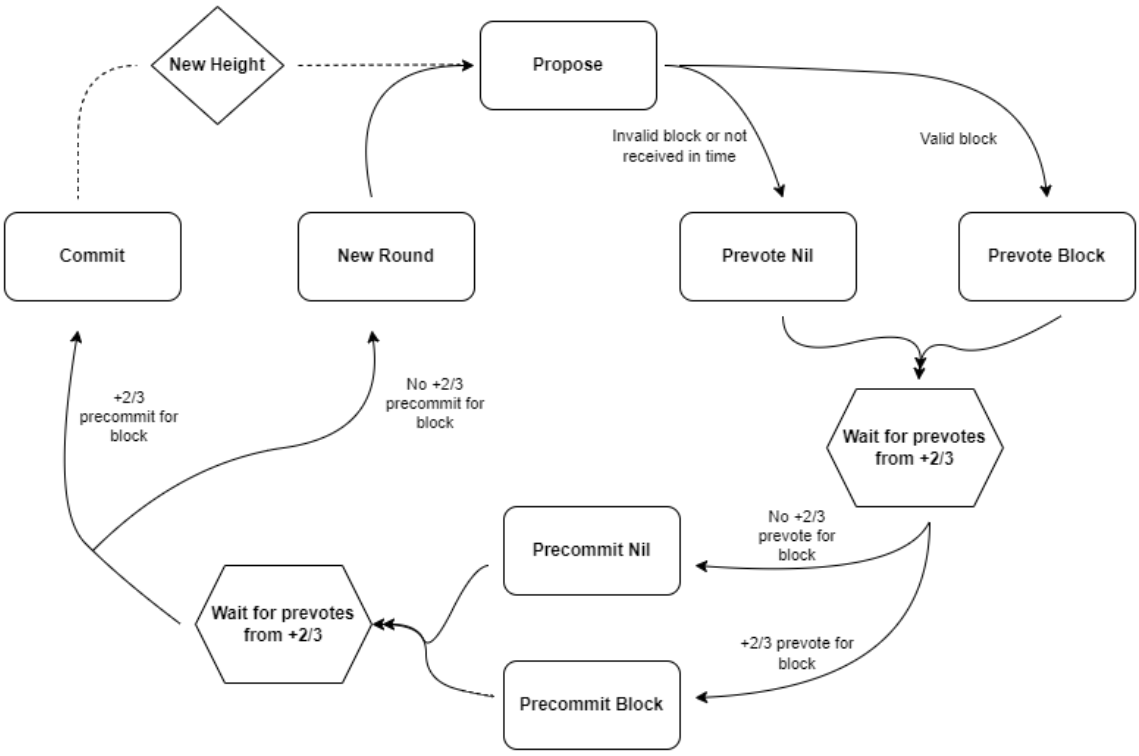

Figure 3.10: Tendermint Diagram [52]

3.3.3 Wanchain - Galaxy Consensus mechanism

In terms of validator nodes used in PoS consensus, Wanchain partitions these into two distinct types based on whether or not they can receive delegations. Delegations is an instrument employed by the Wanchain team to allow users with a smaller holding of WAN to still participate in consensus. Through triple ECDSA, these users can delegate their rights to validators to participate in consensus on their behalf. Non-delegating Validator nodes, as the name suggests, are consensus nodes which cannot accept delegation, however there is no difference in how the two nodes participate in consensus [53].

In Wanchain Proof of Stake, participating nodes are divided into two groups based on their different tasks: the RNP (Random Number Proposer) group and the EL (Epoch Leader) group [53].

A slot is the generation time of a block, that is, a new block is generated in each slot, and an epoch is a time period composed of many consecutive slots. [bar for bar] Staking power of a node is given by a combination of both the amount of WAN pledged and the total lock time that the WAN is secured for. The more the time the WAN remains locked, the greater the associated staking power. This information will be come paramount to understanding the continual cycle of consensus mechanism [53].

According to their staking power, participating nodes are selected to become a member of the RNP group. This group of nodes are responsible for generating the random numbers to be used by the protocol [53].

Similarly, according to their staking power, Epoch Leader (EL) nodes are selected. This group of nodes are responsible for collecting transactions and packaging them into blocks through two cycles of work. The first cycle comprises of secret message array generation. The second cycle takes the secret message array from the first cycle along with the random number generated by the RNP group to decide who has the authority to propose the next block along with the slot time period they must do so within [53].

EL nodes are sorted in ascending order following a hashing of their public keys and the random number. The result produced is now used as an ordering for the allocation subsequent block production rights for this epoch [53].

Whenever a new block is proposed, all the remaining nodes in the EL group are required to include their own proof of validity, public and accessible by anyone [53].

Conclusion

Now moving forward, we are equipped with both a specific and generalised insight into how cross-chain is facilitated as well as the consensus mechanisms used by 3 different protocols. This will aid us in modelling agent behaviour now that we are aware of case-specific actions and states available in the environment.

Chapter 4

Security Methodology

In order to compare security between different cross-chain protocols, we need a security quantification schema. Since blockchain technology is still a vastly expanding sector, there isn't a industry standard to rigorously achieve this to high degrees of accuracy in relation to a real world environment. This paper will use an adaptation of a methodology first proposed in 2016 [6], to achieve a closely-related objective.

To narrow our focus, we will concentrate on the quantifying the security of two blockchain consensus mechanisms. We will cast a majority outlook at Nakamoto to Nakamoto cross-chain transactions and later consider the alterations we would need to make with the purpose of modelling both Cosmos's Tendermint consensus and the Nakamoto blockchain of Bitcoin's PoW consensus together. Studying this will enable us to effectively examine and evaluate the security of any nakamoto-based network.

In this work, we primarily consider the adversarial attack strategy of double spending through cross-chain analogous to the standard double spending attack. This is facilitated by the attacker first publishing an honest transaction T_h, but then privately continues to mine on his own chain, formed from prior to transaction T_h with a conflicting following transaction T_c [6]. Similarly, we will assume that the operational costs of failing such attack are negligible and the adversary is rational [6], thus would wait until the block with the transaction has reached full confirmations on the target chain.

Furthermore, we will also present findings under a premise that the bridge nodes all work in an honest manner and do not help the single adversary confirm malicious blocks. This diminishes the problem space to focus on the double spend attack direct being the transaction confirmed on the target chain but unconfirmed on the source chain. In all other cases, this would break a preconceived premise, for example, the case whereby the transaction is confirmed on both source and target chain is the honest trivial case. The case where the source chain confirms the transaction but it is unconfirmed on the destination chain would mean the bridge has behaved in a dishonest manner which goes against the assumption of an honest bridge.

4.1 Two-stage MDP - Methodology

Unlike traditional MDP structuring, we cannot solve the single-player MDP directly due to the fact that the reward function is non-linear and we use linear iteration methods to find the ideal policy. Instead we utilise a two step methodology by condensing down the reward matrix using a mapping function [6]. The mapping function utilises the concept of eclipsing which is characterised by when nodes are unable to receive information from the honest network and in turn can be mislead into preforming malicious actions [54]. We can model this using ω as the proportion of mining power of the total network that has been eclipsed [6]. Additionally, we will make the same assumption that eclipsed miners will not produce stale blocks whereas the honest network will affected by the stale block rate [6].

Letting ρ symbolise the value of the objective function such that $\rho \in [0, 1]$, thus we represent

the previously discussed mapping function as ω_ρ: $\mathbb{N}^2 \to \mathbb{R}$ [6]

We will express the cross-chain model as a decision problem $M := \langle S, A, P, \omega_\rho(R) \rangle$ with S being the possible states, A is the action space, P is the transition matrix and R the reward matrix, now mapped [6].

4.2 Two-stage MDP - Setup

In this model the action space available to the adversary consists of the following actions:

Adopt

This action resembles when the adversary accepts the honest network's chain and discards their own private chain. The impact of an adopt state can be viewed as restarting of the game, since we begin at an initial honest state, therefore this action is only desirable to a rational adversary when no longer believes they can catch up to the honest network's chain [55, 6].

Override

This action represents the scenario whereby the adversary publishes their blocks from their private chain containing the conflicting transaction T_c, causing an override of any conflicting blocks. This case is only applicable to the rational adversary once their chain length, l_a, is longer than the honest network's chain length l_h [6]. Moreover, it is optimal for the adversary to only publish at least ($\geq l_{h+1}$) of his conflicting blocks if and only if the transaction is confirmed on the target chain but unconfirmed on the source chain. Ultimately replacing the honest network on the source chain with his own private chain.

Match

The match action entails the situation in which, following a block published at height h' the adversary publishes a conflicting block at height h' likewise, causing a race for which block is adopted to the honest network, rather than a more abrupt *Override*. Note: the adversary's block has to have been pre-made ahead of time in order to successfully implement this [55, 6].

Wait

A wait action portrays the circumstance where the adversary continues to privately mine on his own chain without publishing blocks to the honest network until a new block is found [55, 6].

Exit

Made specifically for double spending, the exit action implies a successful double spend has taken place with k confirmations. This action is only applicable under the condition whereby the $l_a > l_h$ and $l_a > k$ since this indicates that the length of the adversary chain is greater than the number of blocks that have been accepted since the block that contained the conflicting transaction [6].

In a similar manner, the state space for this model is constructed as a four-element tuple of the following elements:

- $\boldsymbol{l_a}$ - As previously mentioned, this is the length of the adversary's chain [6]
- $\boldsymbol{l_h}$ - Likewise, this is the length of the honest network's chain [6]
- $\boldsymbol{b_e}$ - This represents the number of blocks mined by the eclipsed victim [6]
- $\boldsymbol{fork}$ - A 3-selection value labelled *irrelevant*, *relevant*, *active* [55, 6] signifying:
  - *Irrelevant* - This resembles the notion whereby the last block has already reached the honest participants when the adversary has only just found the block, e.g. the state $(l_a - 1,\ l_h,\ b_e,\ irrelevant)$. Since this would mean the adversary chain is shorter in length than the honest network, then the *match* action is unattainable [55, 6]
  - *Relevant* - This resembles the notion whereby the last block has only just been found by the honest network and if the length of the adversarial chain is longer than (or equal to) the honest networks chain, this would indicate that a *match* action is possible [55, 6]
  - *Active* - Finally, this resembles the notion whereby the adversary had just performed a match action, thus causing an acceptance race [55, 6]

In the model we use, every state transition, except *Exit*, leads to a new block being created and thus in turn, rewards provided to the honest network, adversary or eclipsed victim for mining of all non-stale blocks [6].

The following are parameters also captured by the extension of our modified MDP model:

- Stale block rate r_s [6]
- Mining power α being the proportion of mining power the adversary controls [6]
- Mining costs c_m being the expected operational mining costs for the adversary, $\in [0, \alpha]$ [6]
- No. block confirmation k [6]
- Propagation ability γ being the proportion of the network receiving the adversary's blocks when both the adversary and honest network release blocks at time t [6]
- Minimum double spend value v_d indicates the critical point at which double spending becomes more profitable than honest mining [6]

4.3 State transition and Reward Matrices

Given these parameters, actions and states, the overall objective for the agent (the adversary) will remain the same as a standard MDP - to maximise the cumulative expected reward. The adversary will always have the option of honest mining available thus the most meaningful result we can gather from the model would be the v_d. [6]

More precisely [6]:

$$P = (\alpha, \gamma, r_s, k, \omega, c_m) \tag{4.1}$$

$$v_d = min\{v_d \mid \exists\ \pi \in A : R(\pi, P, v_d) > R(honest\ mining, P)\} \tag{4.2}$$

The state transition:

State x Action	Resulting States	Probability	Block Reward
$(l_a, l_h, b_e,)$, *adopt*	$(1, 0, 0, i)$	α	$(-c_m, l_h)$
	$(1, 0, 1, i)$	ω	$(-c_m, l_h)$
	$(0, 1, 0, r)$	$(1-\alpha-\omega)\cdot(1-r_s)$	$(-c_m, l_h)$
	$(0, 0, 0, i)$	$(1-\alpha-\omega)\cdot r_s$	$(-c_m, l_h)$
$(l_a, l_h, b_e, \cdot)$, *override*	$(l_a - l_h, 0, b_e - \lceil (l_h+1)\frac{b_e}{l_a} \rceil, i)$	α	$(\lfloor (l_h+1)\frac{l_a-b_e}{l_a} \rfloor - c_m, 0)$
	$(l_a - l_h, 0, b_e - \lceil (l_h+1)\frac{b_e}{l_a} \rceil + 1, i)$	ω	$(\lfloor (l_h+1)\frac{l_a-b_e}{l_a} \rfloor - c_m, 0)$
	$(l_a - l_h - 1, 1, b_e - \lceil (l_h+1)\frac{b_e}{l_a} \rceil, r)$	$(1-\alpha-\omega)\cdot(1-r_s)$	$(\lfloor (l_h+1)\frac{l_a-b_e}{l_a} \rfloor - c_m, 0)$
	$(l_a - l_h - 1, 0, b_e - \lceil (l_h+1)\frac{b_e}{l_a} \rceil, i)$	$(1-\alpha-\omega)\cdot r_s$	$(\lfloor (l_h+1)\frac{l_a-b_e}{l_a} \rfloor - c_m, 0)$
(l_a, l_h, b_e, i), *wait*	(l_a+1, l_h, b_e, i)	α	$(-c_m, 0)$
	(l_a+1, l_h, b_e+1, i)	ω	$(-c_m, 0)$
	(l_a, l_h+1, b_e, r)	$(1-\alpha-\omega)\cdot(1-r_s)$	$(-c_m, 0)$
	(l_a, l_h, b_e, i)	$(1-\alpha-\omega)\cdot r_s$	$(-c_m, 0)$
(l_a, l_h, b_e, a), *wait*	(l_a+1, l_h, b_e, a)	α	$(-c_m, 0)$
	(l_a+1, l_h, b_e+1, a)	ω	$(-c_m, 0)$
	$(l_a - l_h, 1, b_e - \lceil (l_h)\frac{b_e}{l_a} \rceil, r)$	$\gamma\cdot(1-\alpha-\omega)\cdot(1-r_s)$	$(\lfloor (l_h)\frac{l_a-b_e}{l_a} \rfloor - c_m, 0)$
	(l_a, l_h+1, b_e, r)	$(1-\gamma)\cdot(1-\alpha-\omega)\cdot(1-r_s)$	$(-c_m, 0)$
	(l_a, l_h, b_e, a)	$(1-\alpha-\omega)\cdot r_s$	$(-c_m, 0)$
$(l_a, l_h, b_e, \cdot)$, *exit*	*exit*	1	$(l_a - b_e + v_d, 0)$

Table 4.1: The state transition table used in [6]. With $\alpha =$ is the mining power of the attacker, ω is the mining power of the eclipsed node, b_e is the number of blocks in the attacker chain that were mined by the eclipsed node, γ is the fraction of nodes that an attacker can reach faster than the honest network, r_s is the stale block rate and v_d is the value of the double-spend. The fork label is denoted by i, r and a for irrelevant, relevant and active respectively.

4.4 Informal PoS modifications for MDP

Studies within the space neglect focus on MDP in regards to a PoS context. In terms of the action space, it will largely be the same, regardless of consensus, the option to *adopt* will always be present since it simply states the adversary's intent to give up with the attack since they don't believe they can catch up to the chain length of the honest network. The option to *wait* will also always be present as we can see from an example like Tendermint consensus, although selection is randomised, thus the adversary doesn't have a complete control upon selection, he does however have control over proposing the new block and thus the wait would simply be to not propose the next block and Tendermint's time threshold would be broken, resulting in a NIL prevote and precommit. However with a *match* action this will firstly be different due to the two stage design of Tendermint. Between a prevote and precommit there is a locking that ensure a validator can only precommit on the block that they have prevoted on, thus to achieve a successful match action, the adversary must win the propagation race for $\geq 2/3$ of votes for prevote (then most likely consequently the precommit) and ensure a polka doesn't happen in a later round. For similar reasons, the *override* action would be different also.

Furthermore, one extra concept that needs to be captured for most PoS consensus mechanisms is

the concept of slots and epochs. The concept of time under PoW is determined by how computationally difficult the problem is to solve for miners, but in other PoS systems including Ethereum and Wanchain, epochs and slots are concepts to declare time. A slot is, in essence, a time-slice or slither (fixed integer unit of time) and an epoch contains a set number of slots. With a random validator for each slot selected to propose next block and a random group of validators to then verify each block in each slot within the epoch [19]. The epoch system is used in a way that resembles a bookmark, as the first block in each epoch is marked and validators vote on whether pairs of these blocks are valid. These blocks then go through an upgrade process until they are deemed finalised [19]. Structuring in this way would make PoS chains like Ethereum difficult to model in the same way because of adversarial behaviour can occur in many different stages and with the notion of finalisation, this can be viewed essentially as a reset at finality, thus our model would almost certainly need to track this as a parameter.

4.5 Technologies Used

For consistency, comparability and adherence to prior work, we use similar modules in our experiments, focusing on the respected cross-chain transactions aforementioned. Based on this, we used Arthur Gervais's *pow_mdp* repository [56] as a template for the code produced to achieve our results.

4.5.1 Python

We chose to use Python as the primary language to facilitate the MDP modelling due to the fact it is a language I am familiar. Furthermore, there are a plethora of additional libraries and modules that would support the objective in question, of which we will discuss further, later.

We explored the possibility of using either Matlab or R as languages to use to assist with achieving our goal, however we ultimately opted to use Python for 2 reasons:

1. The major swaying factor for using Python over R or Matlab specifically was the available modules for reinforcement learning. Two packages in particular which we had anticipated using was Numpy and openAI Gym. Most reinforcement learning algorithms that utilise machine Q-learning for example, work seamlessly with openAI gym. Moreover, libraries such as Tensorflow, Pytorch and Keras ensure we would always remain well supported if the direction of the project went into deeper learning or neural networks.
2. This is a limited time project, thus in order to maximise my time effectively, we thought it was ideal to work with a language we was familiar with. Python is a language we have used inside and outside of university, thus the learning curve wouldn't be as steep as learning an entire new programming language, instead we would just need read and understand the documentations of any new modules unfamiliar to me.

Mdptoolbox

The most popular module for Markov Decision work on Python is Mdptoolbox. Our selection of using this stems from both it's popularity, being an indication of it's support, and the inbuilt functions make solving descrete-time MDP's relatively easy. Additionally, the Mdptoolbox package provides clear examples to aid formation of valid MDPs.

Matplotlib

Matplotlib is the primary graphical library for Python. This will allow us to translate results into graphical illustrations directly, thus we can easily transfer the graphs to my report, again maximising our time available.

4.5.2 Software quality

a Although this is smaller coding project, it doesn't necessarily mean that code quality should suffer as a result. Using this rationale, we have decided to implement the following conditions to maintain a consistent code style, aiding the overall management of the project.

Formatters and Linters

All Python files used are formatted via Autopep8 and this project makes use of Pylint as our linter. The formatter corrects area such as extra, unnecessary white space, indentation errors and deprecation. The linter instead focuses on bugs and actual code mistakes as well as optimisations in code.

Git Hooks

We utilise (pre-commit, commit-msg, pre-push) git hooks. This meant that the linting and formatting checks were enforced. In addition to this, it provides a standardisation for commit messages supporting message readability.

GitHub

The code for the project is on GitHub, and we introduced multiple CIs for the repository, compilation checks, formatting and linting. This provides us with an automated process in regards to a range of vital checks on any git-push actions, letting me primarily focus on project development.

Conclusion

To close, we have now crafted the picture of how we can model adversary behaviour on a blockchain in order to quantify how robust a cross-chain transaction is against a double-spend attack. In addition to this, we are also aware of the tech stack and modules used to accommodate the modelling process.

Chapter 5

Security Insights

In this chapter, we explain the focus of model including the blockchains and parameters used, and additionally review the results produced. Furthermore, we conclude with an insight about the direction of travel between the tokens in question, target blockchains and the impact it has on the safe number of confirmations for cross-chain transaction.

5.1 Nakamoto-to-Nakamoto Bridging

For the remainder of this paper, Nakamoto-to-Nakamoto bridging will be dealt with a focus on Litecoin to Bitcoin. Although the cryptocurrency market has seen a sizeable growth, our investigations revealed no relevant data that reflects that growth phenomenon in the cryptocurrency market due to market dynamics change over market cycles, which could hinder the usefulness of past data for current/future findings. This is why parameters take into account more recent values in terms of the average block size of the past 6 months and compare this alteration with previously gathered results in order to reproduce newly updated stale block rates. All proceeding findings derived from stale block rate r_s are under the assumption.

	Bitcoin	Litecoin
Block interval (2016 study)	10 min	2.5 min
Public nodes (2016 study)	[redacted]	807 [58]
Public nodes (currently)	15425 [57]	1202 [58]
Mining pools (2016 Paper)	16	12
Mining pools (currently)	30 [59]	10 [60]
t_{MBP} (2016 Paper)	[redacted]	1.02s
t_{MBP} (currently)	0.085s [61]	1.02s[‡]
s_B (2016 Paper)	[redacted] KB	6.11 KB
s_B (currently)	1161.2 KB [62]	147.8 KB [63]
r_s (2016 Paper)	0.41%	0.273%
r_s (currently, remodelled)[‡]	3.17%	1.13%

Table 5.1: Updated parameters for Bitcoin and Litecoin adjusted from [6]

[†]Block interval values have remained the same from 2016 to the time of writing this paper
[‡]Due to time and availability of data constraints, we maintain the t_{MBP} value from the 2016 study for Litecoin
[‡]Estimated from the bitcoin simulator [64] using the new values of s_B

5.1.1 Results

We ran several different tests, using a various chain cutoff points in order to gather as many comparative results. Since [6] illustrates the affect that parameters such as stale block rate and propagation ability. Alternatively, we focus on the number of expected of blocks needed to successfully perform a double-spend given different chain lengths, how changes to the double-spend reward affect strategy and how cost of mining affects agent behaviour.

Figure 5.1: Bitcoin Expected No. blocks given different hashrates, parameters used: gamma=0.1, kmax=5, stale = 3.17%, cost=1

Figure 5.2: Litecoin Expected No. blocks given different hashrates, parameters used: gamma=0.1, kmax=5, stale = 1.13%, cost=1

Figure 5.3: Litecoin Double-spend value for different hashrates given an eclipsed node hashrate, paramaters used: gamma=0.1, stale = 1.13%, cost=1

Figure 5.4: Bitcoin Double-spend value for different hashrates given an eclipsed node hashrate, paramaters used: gamma=0.1, stale = 3.17%, cost=1

The strategy for deciding a safe number of confirmations is formed from the preliminary notion of a safe number of confirmations on each siloed PoW blockchain stems from our assumption that the bridge is completely honest and that there is no delay in creating the transfer out transaction, once the transfer-in has been safely confirmed. Thus, for some pair of PoW blockchains x and y, the safe number of confirmations $k_{safe}{}_{x,y}(x, y)$ is given by the minimum k confirmations whereby k is greater than $k_{safe}(x)$ which is the number of confirmations required to safely transfer into the bridge from blockchain x. This results in the following:

$$k_{safe}(x,y) = min\{k \mid \exists\, x,y \in PoWblockchains : k > k_{in}(x)\} \tag{5.1}$$

Extending our results for the stale blockrates recently recorded in Table 5.1 for Bitcoin and Litecoin, we can identify the following given $\alpha = 0.3$:

For the number of safe confirmations required when bridging from Bitcoin to Litecoin

$$k_{in} = (\quad) \tag{5.2}$$

$$k_{safe}(Bitcoin, Litecoin) = \tag{5.3}$$

Similarly, recorded results for the reverse transfer:

For the number of safe confirmations required when bridging from Litecoin to Bitcoin

$$k_{in} = \tag{5.4}$$

$$k_{safe}(Litecoin, Bitcoin) = \tag{5.5}$$

As you can see, interesting insight is immediately evident, that contrary to popular belief, direction matters. There is a difference in the k_{safe} for two blockchains x and y depending on the direction of transfer. This implies that k_{safe} is in fact non-commutative and therefore, with Bitcoin and Litecoin, greater security needs to be taken for the backword (Litecoin->Bitcoin) direction.

FULL VERSION AVAILABLE ON REQUEST

For the cross-chain transaction of Litecoin to Bitcoin

$$k_{safe}(Litecoin, Bitcoin) = 8 \tag{6.6}$$
$$t_{MDP}(Bitcoin) = 0.73 \ seconds \tag{6.7}$$
$$\phi(Litecoin, Bitcoin) = 5.84 \ seconds \tag{6.8}$$
$$\tag{6.9}$$

These results share the same relationship that the direction of the transfer has an affect on now both the security and the speed of the transaction. Moving further, rather unexpectedly we can see the relationship between the speed and security results with the backword (Litecoin->Bitcoin) transaction being the more safer transaction (by number of confirmations), but on the other hand, the backword (Litecoin->Bitcoin) transaction also has a faster transaction time ϕ. Although our data suggests a positive relationship between the speed and security, we are reluctant to conclude that this result is generalised and can be condensed into being simply a pairing. This is because we would be extrapolating the data to make conclusions for other PoW blockchains, let alone PoS blockchains. Moreover, studies such as [65] take an in-depth look at how the factor of block interval affects the propagation rate, which may, in turn, present different rationale for findings.

FULL VERSION AVAILABLE ON REQUEST

FULL VERSION AVAILABLE ON REQUEST

FULL VERSION AVAILABLE ON REQUEST

Glossary of Terms

Blockchain integrity Referring to the maintenance and assurance of block completeness, consistency, accuracy, 9

Borrowing The process whereby users can receive a loan of cryptocurrency assets from individuals at a specified fee or interest rate for repayment after a specified time. 8

Byzantine Generals Problem A game theory problem highlighting the difficulty individual players face trying to reach consensus without the support of a trusted third party. 11

double-spending An attack where attackers overwrite objective of revenue maximisation through attempts to spend the same coin more than once. 9

ECDSA Elliptic Curve Digital Signature Algorithm is a public key encryption algorithm that utilises the mathematical properties of elliptic curves for digital signatures. 12

Fiat A currency which has been declared legal tender, typically by Government monetary policy backing, e.g. the US Dollar $, the Great British Pound £, the Euro € etc. 8

Gas The measurement unit used to reflect the amount of computational resources required to perform a transaction, as based on supply and demand of computational resources available against the number of transactions awaiting to be processed . 8

Lending The process whereby users can temporarily lend cryptocurrency assets to individuals to on an exchange for a fee or interest payment. 8

Margin trading A trading method whereby the user has access to greater funds for investment, and thus potential profits, provided by third party individuals or organisations in return for interest on their provided capital. 8

Merkle-proof A light-weight method of proof to verify whether data belongs in the tree (data structure) without actually storing or reading the entire tree. 19

Miners Individuals assemble and verify unconfirmed transactions into organised blocks, with rights earned through solving computationally difficult puzzles. 9

Nakamoto blockchains Blockchains that utilize the PoW consensus originally proposed under the pseudonym "Satoshi Nakamoto". 9

Node A computer or hardware that stores the entire blockchain history to enforce rules. 10

Nonce value "Number Only used Once" refers to a counter used during the process of block mining. 9

Parachain The paralleled running of siloed blockchains on the Polkadot system. 11

Permissioned A distributed ledger that require access approval to be apart of. 11

Permissionless A distributed ledger that is open to all. 11

Slashings A penalty to the validator's rewards, usually triggered by double-signing and node downtime. 10

Stake The process whereby an individual commits their cryptocurrency assets to the protocol to support the network. 10

Turing-complete The notion that, under no guarantees of time or memory limitations, the system can run any program and produce a result analogous to a universal Turing machine. 10

Appendix A

First Appendix

Figure A.1: Quantity of cryptocurrencies as of Feb 2013 [3]

Figure A.2: Litecoin [mining pools] [60]

Figure A.3: Bitcoin block propagation [61]

Figure A.5: Litecoin Average block size over the past year [76]

Bibliography

[1] Nakamoto S. Bitcoin: A peer-to-peer electronic cash system; 2008. Available from: https://blockchain.com/btc/bitcoin.pdf.

[2] Malwa S. Terra becomes second-largest defi protocol surpassing binance smart chain; 2021. Available from: https://www.coindesk.com/markets/2021/12/21/terra-becomes-second-largest-defi-protocol-surpassing-binance-smart-chain/.

[3] Statista. Number of cryptocurrency coins 2013-2022; 2022. Available from: https://www.statista.com/statistics/863917/number-cryptocurrencies/.

[4] PPIO. Cross-chain: How blockchains communicate with each other; 2021. Available from: https://medium.com/ppio/understanding-cross-chain-technology-59c0cfaf3.

[5] CryptoCurrency. How does PoS work?; 2022. Available from: https://cryptocurrency.com/cryptos/proof-of-stake.

[6] Gervais A, Karame GO, Wüst K, Glykantzis V, Ritzdorf H, Capkun S. On the security and performance of proof of work blockchains. In: Proceedings of the 2016 ACM SIGSAC conference on computer and communications security; 2016. p. 3-16.

[7] Opensea. BoredApeYachtClub Opensea page; 2022. Available from: https://opensea.io/collection/boredapeyachtclub.

[8] CoinGecko. Axie Infinity Price in USD: Actual Price Charts & News; 2022. Available from: https://www.coingecko.com/en/coins/axie-infinity.

[9] Kaal WA. Everything you need to know about DeFi; 2022. Available from: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4044906.

[10] Uniswap (Uni) price today, chart, market cap & news; 2022. Available from: https://www.coingecko.com/en/coins/uniswap.

[11] Simplified B. Centralized finance (Cefi) vs decentralized finance (Defi) — the battlefield of cryptocurrencies; 2021. Available from: https://medium.com/blockchain_simplified/centralized-finance-cefi-vs-decentralized-finance-defi-the-battlefield-of-cryptocurrencies-bd5250f55c.

[12] Coinbase. What is DeFi?; 2022. Available from: https://www.coinbase.com/learn/crypto-basics/what-is-defi.

[13] Nofer M, Gomber P, Hinz O, Schiereck D. Blockchain. Business & Information Systems Engineering. 2017;59(3):183-.

[14] Blockchain explained; 2022. Available from: https://www.investopedia.com/terms/b/blockchain.asp.

[15] Wood G, et al. Ethereum: A secure decentralised generalised transaction ledger. Ethereum project yellow paper. 2014;151(2014):1-32.

[16] What is Ethereum and how does it work?; 2022. Available from: https://cointelegraph.com/ethereum-for-beginners/what-is-ethereum-a-beginners-guide-to-eth-cryptocurrency.

[39] Deeplizard. Markov Decision Processes (MDPs) - Structuring a Reinforcement Learning Problem; 2018. Available from: https://deeplizard.com/learn/video/my207WNoeyA.

[40] Deeplizard. Policies and value functions - good actions for a reinforcement learning agent; 2018. Available from: https://deeplizard.com/learn/video/eMx0GwbdqKY.

[41] Deeplizard. Expected return - what drives a reinforcement learning agent in an mdp; 2018. Available from: https://deeplizard.com/learn/video/a-SnJtmBtyA.

[42] Deeplizard. What do reinforcement learning algorithms learn - optimal policies; 2018. Available from: https://deeplizard.com/learn/video/rP4oEpQbDm4.

[43] Cross-chain bridge; Available from: https://docs.multichain.org/getting-started/how-it-works/cross-chain-bridge/.

[44] Cross-chain router; Available from: https://docs.multichain.org/getting-started/how-it-works/cross-chain-router/.

[45] Introduction to cosmos hub;. Available from: https://hub.cosmos.network/main/hub-overview/overview.html.

[46] Cosmos network - internet of blockchains; Available from: https://v1.cosmos.network/resources/whitepaper.

[47] Team C. IBC Fungible Token Transfer | Cosmos Developer Portal; Available from: https://interchain.cosmos.network/ibc/transfer.html.

[48] Team W. Chapter 1 - Overview of Wanchain 5.0 - Connecting the DeFi World With Cross-chain Bridge; 2020. Available from: https://medium.com/wanchain-foundation/chapter-1-overview-of-wanchain-5-0-connecting-the-defi-world-with-cross-chain-bridge-[illegible]9f69[illegible]af.

[49] Team W. Building Blocks for Financial Markets for the New Digital Economy. Wanchain white paper. 2017.

[50] Multichain Team. Faq; Available from: https://docs.multichain.org/getting-started/faq/.

[51] Wanchain. An Inclusive Cryptofinance Platform Based on Blockchain. Wanchain white paper. 2017.

[52] Cosmos Team. What is tendermint | tendermint core; 2021. Available from: https://docs.tendermint.com/v0.35/introduction/what-is-tendermint.html.

[53] Wanchain Team. Wanchain - introduction; Available from: https://www.explorewanchain.org/#/technology/privacy-consensus.

[54] Gemini. Eclipse Attacks - Explained - How do they?; Available from: https://www.gemini.com/cryptopedia/eclipse-attacks-defense-bitcoin,https://www.gemini.com/cryptopedia/eclipse-attacks-defense-bitcoin.

[55] Sapirshtein A, Sompolinsky Y, Zohar A. Optimal selfish mining strategies in bitcoin. In: International Conference on Financial Cryptography and Data Security. Springer; 2016. p. 515-32.

[56] Gervais A. Pow_mdp; 2017. Available from: https://github.com/arthurgervais/pow_mdp.

[57] Bitnodes. Bitnodes; 2022. Available from: https://bitnodes.io/.

[58] Blockchair. Litecoin / Node explorer;. Available from: https://blockchair.com/litecoin/nodes.

[59] BTC com. Pool Stats - BTC.com;. Available from: https://btc.com/stats/pool?pool_mode=year.