Cross-Blockchain Technology: Integration Framework and Security Assumptions

Article *in* IEEE Access · April 2022

DOI: 10.1109/ACCESS.2022.3167172

4 authors, including:

Babu Pillai
Griffith University
9 PUBLICATIONS **36** CITATIONS
SEE PROFILE

Kamanashis Biswas
Australian Catholic University
59 PUBLICATIONS **1,138** CITATIONS
SEE PROFILE

Zhe Hou
Griffith University
47 PUBLICATIONS **129** CITATIONS
SEE PROFILE

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier xx.xxxx/ACCESS.2022.DOI

Cross-blockchain technology: integration framework and security assumptions

BABU PILLAI [1], Kamanashis Biswas [2], Zhé Hóu [1], and Vallipuram Muthukkumarasamy [1]
[1]Griffith University, School of ICT, Gold Coast, QLD, Australia
[2]Australian Catholic University, Brisbane, QLD, Australia

Corresponding author: Babu Pillai (e-mail: babu.pillai@griffithuni.edu.au).

ABSTRACT Interoperability is identified as one of the major design constraints for blockchain technology. Cross-blockchain technology is fast evolving as the demand for value transfer among different blockchain systems is growing. A generic cross-blockchain design methodology for interoperability requires a set of suitable components to facilitate the integration process. One of the main challenges in the integration is to protect the integrity of the shared data. Various integration systems and their components may operate with different underlying security assumptions and this may lead to compromise of the overall security. In this paper we review the latest advancement in blockchain integration systems and provide a comprehensive analysis of integration characteristics for cross-blockchain technology and then define the essential components and modes of integration. Based on the outcome of our review, we propose a novel integration design decision framework that identifies key assumptions and critical characteristics of the cross-blockchain technology. The proposed framework facilitates the best-practice decision-making process. This reduces the potential for design errors and the associated security risks and performance degradation of the overall system.

INDEX TERMS blockchain interoperability, cross-blockchain technology, blockchain integration system, cross-blockchain protocol, cross-blockchain integration framework.

I. INTRODUCTION

Blockchain emerged as a disruptive technology that can transform the transaction process in the digital world. This technology has the potential to disrupt processes across many industries [1] by offering a tamper evident and verifiable proof to track a series of digital transactions. In an explosion of research activities and investment by entrepreneurs towards developing a blockchain-based ecosystem, various proof-of-concepts and pilot systems have emerged. Many of these projects offer a solution to a specific problem [2] and operating in their own respective silos with their specific networks [3].

Even though the core concept and capabilities between the blockchain platforms and networks remain similar, due to the inherent nature of the technology, they are not readily interoperable [4]. Therefore, these networks must be appropriately integrated to achieve interoperability. Integration is a process of connecting different applications so that data from one system can be accessed and accepted by the other system.

Generally, integration involves middleware mechanisms to transport the data and to make it accessible to the other network. In the case of blockchain-based systems, the most significant obstacles in interoperability are the consensus of each chain and how data moves from one chain to another.

A. CROSS-BLOCKCHAIN TECHNOLOGY

Cross-blockchain is a relatively new concept seeking to enable interoperability between blockchain networks. The fact that blockchain networks operate in silos and their inability to cross-communicate has made them unsuitable for many real-world applications. Cross-blockchain technology aims to solve these issues by connecting blockchain networks through integration processes that enable cross-communication [5].

The evolution of *integration mechanism* advances from data integration using common data standards and file sharing methods to functional integration that allows real-time data and functionality exchange. Within blockchain technology, several projects experimenting with cross-blockchain integration, proposing interoperability through various inte-

gration architectures (cf. Section VII). There are projects focused on solutions that connect homogeneous networks. For example, in [6] the *SubChains* and *InterChain*, and in [7] *alliance chain* and *private chain* makes up the interoperable network of homogeneous networks, whereas, other heterogeneous network need to be customised to connect to this network. Although Polkadot [8] and Cosmos [9] platforms bring a new level of interoperable blockchain ecosystem, they are not directly interoperable to each other. In practice, the Polkadot bridge and the Cosmos inter-blockchain communication protocol functions in a similar fashion and they both enable similar outcome of interoperability but only within their own ecosystems.

There are multiple works on cross-blockchain technology classifying blockchain interoperability in various ways (cf. Section VII). In early 2016, Buterin [10] proposed to divide cross-chain interoperability protocols into three types: notary schemes, sidechains/relays, and hash locking. Niclas et al [11] discuss about three patterns of manual asset exchange, notary schemes, and relays. Similarly, Belchior et al [12] classified blockchain interoperability approaches into three categories: cryptocurrency-directed approaches, blockchain engines, and blockchain connectors. Gang [13] categorised interoperability as chain-based, bridge-based and dApp-based interoperability. Apart from these, Qasse et al [14] classified them as sidechains solutions, blockchain routers, smart contracts, and industrial solutions. Relatively, these categories and classifications approach the interoperability problem differently.

A crypto-currency [12] directed approach aims to address interoperability based on the *functionality* (cf. Section III-B). On the other hand, *mechanisms* like notaries, relays and hash locking [10] aim towards integration networks. Unlike the above, the chain-based, bridge-based and dApp-based solutions [13] mainly depend on the *mode of integration* (cf. Section IV-C) In a nutshell, the approaches described in [12], [13], [15] are mostly function-oriented and they are named after the mode of integration, whereas, the approaches proposed in [10], [14] are largely using specific mechanisms and named after mechanisms.

In general, the current research and development of cross-blockchain technology is in its early stage. Different projects adopt different approaches based on the application requirements. In this paper, we identify the *integration system* (cf. Section IV-A) as the core functional unit of cross-blockchain technology. Depending on the nature of the network interconnecting, an integration system needs to meet precise requirements. There have been several integration models proposed in the literature ([10], [13], [14]), relatively not knowing about the properties of the actual integration process. In this paper, we identify and analyse the key issues related to security assumption of the integration process in addition to providing a cross-blockchain design decision framework.

Currently a growing number of cross-blockchain technology seek to enable interoperability (e.g., Interledger or Polkadot). However, cross-blockchain integration pose different technical requirements (e.g., integration mode or verification mechanisms) and non-technical requirements (e.g., concerning performance and security) on the integration system. Consequently, developers must carefully compare cross-blockchain technology in order to choose an artifact that best suits their use case.

B. RESEARCH GAP AND CONTRIBUTIONS

The issue of interoperability is likely to be addressed through designing an effective cross-blockchain integration system. The overall security of a cross-blockchain trade is subject to the security of the weakest link in the ecosystem. Therefore, integrating blockchain networks to exchange data remains the question of credibility to trust the integration system. Although the independent cross-blockchain protocols can determine the correctness and atomicity [16], [17] of the transfer, there remains two key challenges: (i) how can the network nodes trust the authenticity of cross-blockchain data and, (ii) what are the underlying security assumptions of various integration scenarios?

This research aims to address these challenges by defining cross-blockchain integration component characteristics and proposing a novel cross-blockchain integration design decision framework that can systematically identify the security assumptions of a given integration scenario.

The main contributions of this paper can be summarised as follows:

- Identify and define integration characteristics and challenges for cross-blockchain technology and then propose a generic integration system model.
- Develop a cross-blockchain integration design decision framework that effectively identifies the characteristics of various application scenarios.
- Formulate and analyse various application scenarios of cross-blockchain integration through the proposed framework.
- Evaluate and compare different security assumptions of various cross-blockchain integration scenarios.

The outcomes of this research will pave the way for new research directions related to integration of blockchain networks in a more secure manner.

C. GENERAL ASSUMPTIONS

In a cross-blockchain transaction, we assume that the transaction result can be a value transfer or the execution of a contract, and that:

- Underlying networks are secure with a concept of transaction finality within finite time, after which the transactions cannot be rolled back [18].
- We loosely use ‘value’ as the generic term to represent the cryptographic object that the blockchain carries.
- Security aspects such as double spending and 51% attack of cross-blockchain systems are addressed by the integration protocol.
- We do not consider the semantics of data and exchange rate between two different tokens.

II. TECHNICAL BACKGROUND

In information systems, interoperability is generally understood as the ability of two or more systems to communicate and exchange information [19]. The process related to exchange is the use of the exchanged data to carry out an operation, referred to as interoperation [20]. Interoperability can also be characterised as a relationship of the exchange and cooperative use of data. Interoperability occurs when two systems successfully use the exchanged data despite differences in language, interface, and execution platform [21]. Recently, interoperability has gained different definitions within the context of blockchains. An increasingly common usage refers to the *interaction* and *exchange* of data between networks of blockchains. This opens up various possibilities of cross-blockchain transactions, for example value transfer in the form of asset or payment versus payment and payment versus delivery schemes or information exchange [10].

A. BLOCKCHAIN INTEROPERABILITY

Generally, interoperability is developed through functional design principles and standards. Thus, it forms a base for different applications/systems to cross-communicate [22]. In information system, many approaches exist to achieve interoperability, such as the Integrated approach – where a globally agreed format of data structure exists; the Unified approach – a common format with semantic understanding exists; and the Federated approach – where connections will be dynamically established rather than predefined [22].

For blockchain, a token standardisation effort by Ethereum group result in development of a series of standards called Ethereum Request for Comments (ERCs)[1]. This standard focus on setting up guidelines and define fundamental functionalities for token contracts such as total supply, divisible, fungible, non fungible etc [23]. However, for interoperability they can only provide consistent functionality across different networks.

In [24] the authors defined it as semantic dependence between distinct ledgers of blockchains for the purpose of transferring or exchanging data/ value, with assurances of validity or verifiability. Authors in [4] expect that, if one blockchain's network rules accept a transaction from another network, then they are interoperable with each other. Authors in [25] and [12] cited a technical report from the National Institute of Standards and Technology defining interoperability as the ability of one system to change the state of another blockchain system. In [26] the authors anticipate blockchain interoperability to "connecting multiple blockchains to access information and act on that information by changing their own state or the state of another blockchain". However, in our early work [27] we expressed that cross-blockchain transactions may not make direct state changes to another blockchain network. Instead, a cross-blockchain transactions should trigger some set of functionalities on the other system that is expected to perform an operation within its own network.

In [28] interoperability is defined as "the capacity to openly share information across multiple networked blockchains" such that the users should be able to easily interact with one another even though they use different protocols to agree within their local blockchains. It is also defined as the ability of one blockchain to change the state of another blockchain enabled by cross-blockchain transactions [12]. Authors in [13] describe it as the ability to correctly conduct asset transfer without compromising the legacy design philosophy of either blockchain system. In [29] defined as a transaction of blockchain network N_1 causes or requires another transaction in blockchain network N_2. This paper use cross-chain interaction to represent cross-chain interoperability.

In conclusion, *interoperability leads to the possibility of exchanging or sharing messages or values in the form of data across different networks of blockchains*. However, the unique components and properties of blockchain such as consensus, independent state and its finality, and crypto assets introduce unique challenges for blockchains to interoperate.

B. DATA ACCEPTING AND ACCESSING CHALLENGES

Let us consider the value transfer problem between two different blockchain networks. The *value* carries, and *transfer* is two key components in this problem. The value is a kind of virtual asset entity existing within the network. A private key symbolizes the ownership of this asset in the network. A transfer is a process of altering the ownership of an asset within a network. A cross-blockchain transfer moves a user's asset into a different position and shifts its value into a different blockchain network. However, such operations are challenging as they require the networks to make state changes based on the data obtained from other networks. To achieve this, the networks involved must trust and accept data from one another. Because of the inherent nature of blockchain, there are technical and practical limitations for blockchain networks to *accepting* and *accessing* data from one another to make state changes.

1) Technical difficulties in accepting data

Blockchain network maintain state through their internal consensus mechanism. Participants must be in sync with the current state to learn about current transactions. This synchronisation requires distributed consensus capability to be an important component of the blockchain system. Consider the Internet or an intranet as an overall network and blockchains as sub-networks that hold data about various digital values. *Most state-of-the-art blockchain technologies are designed so that they operate as isolated or stand-alone systems.* That is, the network of nodes, who are the stakeholders, decides on the current state of the system based on an agreed protocol [1]. This protocol dictates the value and the consensus model. Most importantly, the value has been created by and exists only *within* the system and its nodes [30]. Therefore, enabling

[1] https://eips.ethereum.org/erc

interoperability that introduces a way to exchange value from one blockchain to another is challenging because it is hard to reach consensus across different networks. More specifically, one blockchain can not establish the correctness of another network's state without comparing and validating the ledger data of the other network.

2) Practical challenges in accessing data

The specific design of an isolated network makes blockchain technology secure and reliable, as the network only needs to form a consensus on the data solely generated within its own environment. In addition, they use decentralisation to distribute the data to ensure the majority control the network. However, this can only provide a robust solution for users within their own network. Introducing interoperability among blockchains means that a blockchain system must depend on other systems' data to process cross-blockchain transactions. Given the decentralised nature of the technology, where a number of nodes participating in reaching finality, these nodes must retain the same result. For that, the nodes must have or be given the relevant information in order to process a transaction. If the nodes are set to fetch data from other blockchain systems, the dynamic nature of values interferes with the consensus. Therefore, each blockchain system generally has its own idiosyncratic integration issues that need to be addressed through a secure integration mechanism that facilitates integration and communication between blockchains running on the same platform or different platforms, and/ or off-chain systems such as APIs or Oracles.

C. PRELIMINARIES

Definition 1 (Cross-blockchain protocol): A cross-blockchain protocol aims to synchronise parts of ledgers on N_1 and N_2, both of which are inherently trusted to operate correctly.

Let us define P as the cross-blockchain transaction process in N_1 and Q as a process in N_2. A cross-blockchain protocol includes P and Q.

Definition 2 (Integration): Integration refers to the process of connecting networks so that data from one network can be accessed by the other networks.

Let us assume networks N_1 and N_2 aim to interoperate therefore they are integrated through some form of integration mode such as notary, bridges or connectors.

Definition 3 (Integration system): An integration system help the consensus participants of N_2 that P was included in N_1 who in turn enforce the inclusion of Q in N_2 or directly enforce the inclusion of Q in N_2.

Definition 4 (Integration mode): The integration mode is designed for nodes to perform physical connection of networks and help the integration system to transfer data between N_1 and N_2.

Definition 5 (Atomicity): The transfer operation should only execute one outcome, either the transfer succeeds and the asset is transferred to the recipient; or it is fails and the asset returns to the sender.

The above property is sometimes referred to as "all-or-nothing". In the case of failures during the protocol execution, every transfer participant must be able to regain possession of the originally owned assets.

Definition 6 (Asset and coin): We define asset v and coin c as digital representation of tradable objects available on a blockchain, where c represents native token of the system and v represents token created on the system through a defined protocol.

Definition 7 (Cross-blockchain trade): Cross-blockchain trade refers to the exchange of asset or transfer of its value between users in different blockchain networks.

Definition 8 (Transaction finality): Transaction finality refers to the guarantee that a transaction is permanently accepted by the network. In effect, it is computationally infeasible to revert or alter that transaction afterwards.

Definition 9 (Degree of Decentralisation): The degree of decentralisation (DD) is defined as the proportion of the total number of nodes against the number of validating nodes in the distributed network. Let n represent the number of nodes and x representing the validating node then DD is calculated by:

$$DD = \frac{number\ of\ validating\ nodes\ (x)}{total\ number\ of\ nodes\ (n)} \tag{1}$$

A report published by Algorand[2] defines this concept as the ratio of economic value securing the network by the economic value stored on the network.

Definition 10 (Standalone transaction): A standalone transaction is defined as a transaction that is included in a block and that block is accepted by the network.

Let us assume transactions are the smallest element of blockchain data and blocks are collections of transactions. In blockchain, transactions are built on top of its ancestry record where as a blocks are built on top of a previous block. Within the network, the validity of a transaction Tx is based on its previous transaction and for block b through the mining process of checking the validity of transaction and go through the consensus process. Therefore, once a transaction is validated by a miner and included in a block it can be accepted as a standalone transaction.

III. ASSET PROFILE AND CROSS-BLOCKCHAIN TRADE

Primarily, blockchain technology has two use-cases distinguished by the nature of operations: a technology for transacting digital objects and a platform for performing arbitrary computation. Within these use-cases, there exist various types of cross-blockchain requirements based on the applications. As a technology for digital transactions, blockchain applications represent digital value in the network. These values are digital representations of items that are being created and exist in a blockchain. To perform cross-blockchain trade, these digital objects must have an agreed understanding of

[2]https://arringtonxrpcapital.com/2021/07/19/illuminating-the-dark-age-of-blockchain-algorand/

value. This permits transacting parties to refer to the same definition of the virtual asset to be exchanged.

A. DIGITAL ASSET PROFILE

In the context of interoperability, the asset must be categorised for the participating systems to understand its syntax and semantics. Our previous work [31] maps blockchain cryptographic assets into different categories. This mapping is based on the purpose and type of the value the asset carries, which help to guide the design and implementation process of interoperability. In general, the application domain can be divided into three groups based on their value type: crypto-coin, asset token, and data.

1) Crypto-coin

Crypto-coins, also referred to as cryptocurrencies, are a new form of money implemented on the blockchain for the purpose of exchange independent of any central control such as a bank. Cryptocurrencies like BTC (Bitcoin) or ETH (Ethereum) are called native currencies because they are developed to exist within the system and are used to pay for the computational services offered by the system. These crypto-coins (BTC and ETH) are built into the system as part of the protocol. Therefore, they are not directly transferable to other systems; instead, they can only be exchanged.

2) Asset-token

Unlike crypto-coins, asset-tokens are not native to a blockchain network. They are created on top of a blockchain and can be used to represent a wide range of assets beyond currencies. Asset-tokens are commonly implemented in smart contracts that are associated with physical items such as cars, properties, or non-physical items such as company shares. A piratical use case for asset-tokens are Non-fungible tokens (NFT) that represent unique assets [32].

3) Data

Another advancement of blockchain technology is its ability to perform arbitrary computation and run business rules/logic in the form of smart contracts. For most of the use-cases in this domain, cross-blockchain interoperability is required since such functions need to interact across different networks [33]. For example, information can be transferred in the form of data, or smart contracts can be invoked across blockchains. To do this, the networks must deploy relevant smart contracts on both networks, and they must register the contract address of each other. After registering these contracts, each party can subscribe to events published by the other. Potential integration nodes listen for such events and will submit the offers regarding an event to the distribution contract.

B. CROSS-BLOCKCHAIN TRADE

In a nutshell, the application goal of blockchain systems can be classified into three categories as shown in Figure 1: a) *value exchange*, b) *value transfer*, and c) *data exchange*.

1) Crypto-coin exchange

Crypto-coin exchange aims to facilitate users trading their coins for other types of coins. For example, Alice sends one BTC to Bob on the Bitcoin network, and Bob sends an agreed amount of ETH to Alice on the Ethereum network. All that has happened here is a change of ownership of coins on both networks. BTC still remains in the bitcoin network, and ETH remains in the Ethereum network. This is because crypto-coins are bound to one network; therefore, they can only be swapped between users. As a result, for crypto-coins, the exchange always needs a counter-party who is willing to exchange tokens. This means the solution requires the existence of a liquidity provider to facilitate the exchange.

2) Asset-token exchange or transfer

Asset-token in the blockchain system is essentially a technology that produces virtual tokens representing values within a closed network. Blockchain applications using asset-token may require the tokens to be exchanged or transferred between networks. For asset-token exchange, the same solution as for crypto-coins can be used, whereas transfer is challenging because of the possibility of double-spending. At a technical level, current research explores the possibility of moving the asset by locking or destroying it on one network and unlocking or recreating it on the other network. But, at a semantic level, there is a need to solve the challenges pertaining to data acceptance (ef. Section II-B1) and accessibility (ef. Section II-B2).

3) Data exchange

As a platform for arbitrary computation, we assume the networks have programming capability. For example, in Ethereum, code in the form of arbitrary data insertion is performed by deploying it as a smart contract to the programming interface, which is then used as a back-end for applications to perform computation and data storage. Smart contract-based solutions target general interoperability, by deploying application-specific cross-blockchain functions to the networks [5]. This mechanism assumes that participants on one blockchain can communicate with smart contracts on another blockchain and vice versa. The cross-blockchain function call enables applications to access information that resides on one blockchain from another blockchain in a decentralised and trustworthy manner. Such functionality can create applications to exchange or retrieve data or to run function calls across different networks.

IV. INTEGRATION COMPONENTS OF CROSS-BLOCKCHAIN TECHNOLOGY

Cross-blockchain technology provides blockchain interoperability by implementing different integration schemes. In other words, this allows blockchain networks to communicate with each other in a defined way. However, such communication requires a specific configuration of integration components. This section describes an integration system model and its related security issues.

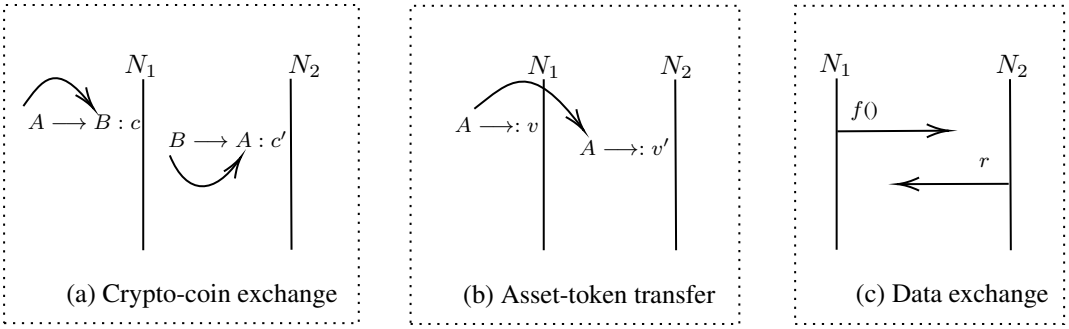

FIGURE 1. Cross-blockchain trade scenarios.

A. INTEGRATION SYSTEM MODEL

Figure 2 shows the key components of cross-blockchain technology identified in our research such as the networks (e.g., N_1 and N_2), cross-blockchain protocol, an integration mode, and an integration system.

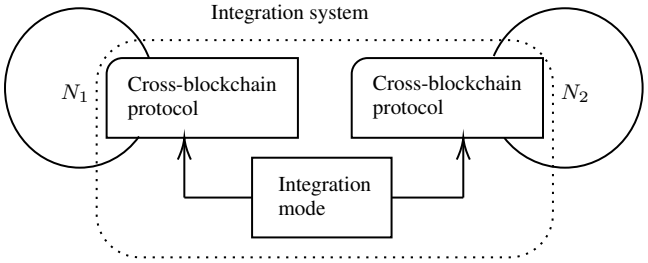

FIGURE 2. Components of cross-blockchain technology.

Let us assume that networks N_1 and N_2 are source and destination networks, respectively, running their own blockchain protocols (consensus algorithms and value type). To interoperate, these protocols should include or accept a cross-blockchain protocol that defines the acceptance and access criteria for cross-blockchain data. N_1 and N_2 can have different blockchain protocols, but they should use the same cross-blockchain protocol. The *integration mode* is part of the cross-blockchain system to integrate the data between N_1 and N_2.

B. INTEGRATION SYSTEM SECURITY ISSUES

We consider the networks integration components to be synchronous with an upper bound time delay. i.e. if a message is not received within the time-bound, then it is considered as a message fault. Technically, in realistic communication on an open network, the messages may have an arbitrary time delay. It is a challenge to determine whether a message is delayed or has been dropped. Therefore, we consider different types of attacks that might affect the cross-blockchain message.

Let us consider two networks, N_1 and N_2, integrated through integration system Ω, and let F be the corruption threshold that Ω can tolerate. We assume the application designer has chosen an appropriate integration design based on the application's security requirement. As these networks are independently governed, it is complicated to address the security of cross-blockchain transactions.

In this context, the fundamental problems are:

a: Security difference.

Security assumptions for every blockchain network are different. Some are backed by thousands of miners, while some are backed by few miners. Integrating data from a weaker trustable ledger to a stronger ledger can make the latter susceptible to third-party manipulation and various other discrepancies. Let us assume two networks of N_1 and N_2 interoperate through an integration system. A security downgrade [29] issue may occur if the integration system is less secure than N_1 and N_2.

b: The weakest link.

The security vulnerability of an integration system could be its weakest link in cross-blockchain networks. For example, if two networks with different security assumptions interoperate, what happens if one network's integration node gets compromised?

a) Integration system nodes could collude to compromise the cross-blockchain transaction, e.g., selfish mining or block withholding [34]. Therefore, if the networks transact large amounts, safety should be maximised so that almost all nodes would need to collude to break the system. That means, such a transaction must support a high safety threshold ($F > 51\%$) to increase the cost of attacks. High safety thresholds combined with maximum decentralisation create a network that is hard to compromise.

In other words, the defined degree of decentralisation (DD) of a truly decentralised system should remain ≥ 1 as the number of users n grows.

b) If one of the networks is compromised, the entire cross-blockchain trade is at risk because of the possibility of double-spending. In this paper, we assume that the participating networks are secure. However, these are open security concerns that need to be considered when designing the integration system. Current research addresses this by using hash-time locks with the cross-blockchain protocol.

c: Lazy node.

We consider the possibility of some integration nodes to be rational and some to be byzantine. A rational node deviates from the protocol if it can gain better benefit whereas a byzantine node may intentionally misbehave to corrupt the network.

Generally, in blockchain, incentive mechanisms are used to encourage nodes to follow the protocol. However, a rational node may get influenced by other factors that provide better benefits to deviate from the protocol. For example, in *lazy behaviour*, a rational validator node may not care about the correct results, therefore simply agree or accept other nodes results that maximise his benefit by not executing the request. This could lead to an incorrect result being successfully submitted if it is proposed by a byzantine node.

Let us assume N_1 and N_2 are integrated with n integration nodes out of which n' rational nodes and n'' byzantine nodes. An integration incentive mechanism must enforce these n' nodes to respond with the correct result. In [35] this is enforced by requesting a response that force every validator node execute results.

Compared to the blockchain network, distributed consensus process where nodes may participate passively may result in weak security of the integration process; therefore, the integration nodes must actively participate in the integration process.

d: Integration system's DD.

This DD specifies the distribution of the integration systems' validators. In a panel discussion,[3] Vitalik talks about the balance between centralised and distributed can also be depend on its functioning layer. For example, a base layer (layer 1) benefit from maximum decentralisation, whereas in the application layer (layer 2), where users directly interact with each other, decentralisation is less important.

When it comes to cross-blockchain technology, instead of complete decentralization of the integration process, a mix of distributed nodes with and some form of cryptographic technologies like encryption or zero-knowledge proofs allow the integration system to provide stronger safety guarantees without compromising much on the performance.

The DD of an integration system is an open challenge. Let us assume N_1 and N_2 interoperate through an integration system Ω. Will the DD of Ω should be of N_1, N_2 or lower than that?

The wormhole protocol[4] us a multisignature scheme for validators. Considering the *signature length* and the *gas cost* the wormhole integration protocol optimises to set of 19 validators in their updated (v.2) protocol. These validator nodes are picked from highly reputed institution that has a reputation in the validation.

[3]https://youtu.be/vsrA83z7Coo?t=2250

[4]https://wormholenetwork.com/en/network/

e: Full vs partial verification.

We make the distinction between full and partial verification such that: full verification of a transaction is done by miners through the mining process, whereas partial verification of a transaction is simply checking whether that transaction is included in a block accepted by the network.

Technically, it is not feasible to replicate one network data to another network for verification; therefore, the current integration solution exchanges relevant data to prove the transaction proofs. This could arguably lead to a 'security downgrade' issue as per [29] because the exported standalone transaction data do not have visibility of its previous history. However, as per definition 10 a standalone transaction data carry transaction semantics and proofs. The security threshold of data acceptance problem (II-B1) can be leveraged by full verification or partial verification. For example, using *full verification* through a relay (the destination network to run a full node on source network to get the full history of destination network data for verification) or *partial verification* where the networks only verify the relevant cross-blockchain transaction data.

f: Cross-blockchain transaction liveness.

Liveness ensures that a message from a source N_1 to destination N_2 or from N_2 to N_1 must be accessible and available to the interested party. So, for any two networks, N_1 and N_2, with a cross-blockchain system Y in the middle, assuming that N_1, N_2, and Y make progress following their consensus rules, then there is a time in the future such that any request from N_1 or N_2 can be properly finalised within a set time frame. If the threshold, F, is minimal, i.e. $F = 1$, and that one node goes offline, there should be a fall-back solution. The cross-blockchain protocol for Y for a) recover any value that Y might have locked, or/and, b) fix the state of blockchain N_1.

The integration system solution must be robust against eventual faults. This feature implies the redundancy of components and the avoidance of a single point of failure. The requirement for the right integration architecture is important: Any application on any N_1 can communicate with N_2 via an integration system using a unified integration process. For that, the design needs to consider the application's requirements and retain the properties of the connected blockchain networks.

C. CROSS-BLOCKCHAIN INTEGRATION MODES

Several integration strategies have emerged that offer cross-blockchain operations [10]. Figure 3 shows the architecture of integration modes identified in this paper.

1) Direct integration mode

In this mode, nodes of N_1 network verify the information of N_2 on N_1 through a light client running on N_1. A light client uses on-chain nodes to maintain state information about other networks on its own network. Each network runs a set of independent relays and the light client [10], [36].

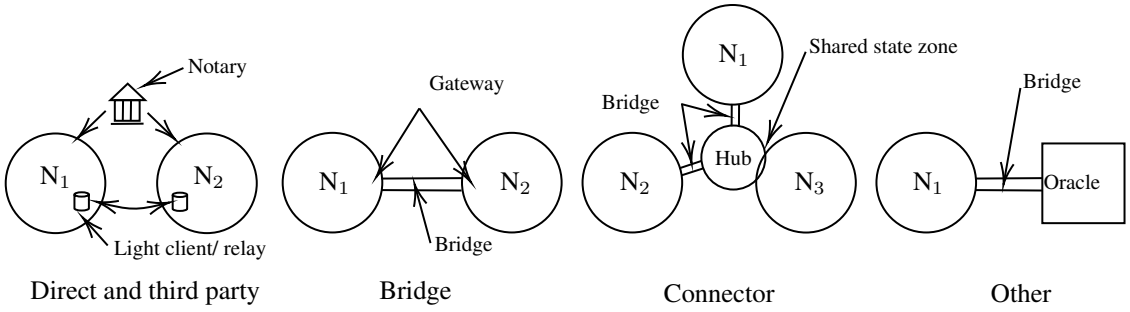

FIGURE 3. Different integration modes.

These nodes follow other networks' consensus status and store necessary data to verify their state information such as accounts, contracts and transactions details. Verifying the state of other chains by implementing light clients is not a scalable solution [13] and may create problems such as single point of failure and lack of decentralisation. Technically, this also requires each network to maintain a reasonable amount of resources for data storage and computation.

2) Third party integration mode

A third party mode aims to provide interoperability through a trusted entity. This is considered the most comprehensive method of connecting two blockchains. The scheme is dependent on a third party federation of validators/notaries to attest events on another network [10]. The federation helps the destination network to verify the event that took place on the source network. In other words, the notary scheme enables the mapping of data that exist in two different networks which helps the users to process the trade of assets.

Let us assume user $\mathcal{A}$ and $\mathcal{B}$ engage in a trade of asset v in N_1, given as $\mathcal{A} \rightarrow \mathcal{B} : v$. User $\mathcal{C}$ on N_2 is waiting for the successful execution of the trade Tx ($\mathcal{A} \rightarrow \mathcal{B} : v$) on N_1 to make a payment Tx' on N_2. In order to provide third party service, the validator node defined as $\mathbb{N}$ must be a participant in both N_1 and N_2 networks.

Once the Tx on N_1 executes successfully, $\mathbb{N}$ authenticates the Tx' details to $\mathcal{C}$ on N_2, given as $\mathbb{N} \rightarrow \mathcal{C} : \overline{x}$ ($\overline{x}$ denotes external data). In this situation, details about $\overline{x}$ are not known to N_2, therefore, as an intermediary, $\mathbb{N}$ will provide those details. $\mathbb{N}$ will have to establish necessary properties to satisfy the network consensus rules so that the network protocol accepts $\mathbb{N}$ as a trusted entity.

3) Bridge integration mode

Bridge mode can be viewed as built-in integrated links for a blockchain interoperability. The link will act as a connector for blockchain network participants to access external data. A bridge system uses gateway nodes to facilitate cross-blockchain communication. Gateway nodes interact with connect networks perform computation based on such interactions. Depending on the distributed ledger, bridges may be full nodes resilient to crashes [37] and participate in the consensus process of the connected networks. Gateways can be embedded in the user's wallet or in a third party server that connects the user's request to the corresponding blockchain network. Generally, a gateway is used to leverage information from the other system with which the user is not associated with [38].

Gateways are also helpful to connect blockchain nodes to the outside world to get data or information. For example, smart contracts deployed on a blockchain system usually depend on the data from their own network to execute the transaction. However, there are use-cases where a smart contract needs to access information outside of its network. In a distributed environment, this will be extremely difficult, and miners will have different results. Gateways can help the nodes to access the data from an external source and thus can expand blockchain systems capabilities [39], [40].

4) Connector integration mode

The architecture of a connector mode can be seen as different blockchain networks connected through an integration hub that creates a network-of-networks. The integration hub consists of a network that helps to connect the networks in a decentralised way. The hub will create the pathway for communication between network components and will have the rules that govern those interactions. The hub will be able to facilitate the connection of many networks and act as a routing device for participating networks. The connected networks are either preconfigured as sidechain with the hub or connected through a bridge in this integration mode.

5) Other modes

Apart from the defined modes, blockchain networks have other potential mechanisms to connect with off-chain data sources such as oracles and APIs. Connecting a blockchain network with an off-chain system requires an integration infrastructure system such as a bridge. Typically, these interfaces are custom-made to listen for specific events from the off-chain data source, fetch the data, and, if required, format it. A major concern about these mechanisms is how to ensure the authenticity and correctness of off-chain data sources.

D. CROSS-BLOCKCHAIN PROTOCOLS

Cross-blockchain protocols are an active area of research that tries to provide certain guarantees when moving value/data between blockchains networks [41]. Several protocols have been proposed over the past few years, such as hash time locks, cryptographic proofs - rollups [42], Proof-of-Burn [43]–[45] and signature-based protocols [46]. Technically, these proposals built on the security aspects of the cryptography involved in the protocols.

In the context of cross-blockchain value exchange/transfer, the total unit of value that exists on the networks should remain the same. In order to implement such a scheme, the value representation on one blockchain needs to be locked or removed to represent the same unit of value on the other network. That is, if n tokens' value is transferred from N_1 to N_2, the corresponding value of n tokens on N_1 must be decreased, and an agreed value on N_2 must be increased. Technically, this process are achieved by specific protocols such as *atomic swap*, *lock/unlock* or *burn/mint* processes described below.

1) Atomic swap protocol

The atomic swap (*atomic swap*) protocol enables bidirectional transfer of tokens through a series of transactions which will not transfer the tokens until both transactions are successful [47]. The *atomic swap* transactions are processed through a *hash time lock contract* (ef section IV-E1) where the tokens are locked with a secret code, γ whose hash value H(γ) is included in the transaction. In the hash-locking method, a participant exposes it's γ first, and then subsequent participants use the exposed γ to obtain the initial user's asset, thus ensure atomicity. The time-lock is a condition that prevents tokens from being redeemed or refunded, until or after a specific time interval has elapsed [48].

In simple terms, *atomic swap* enable two users to swap their tokens in a trustless manner. However, they are limited to token swaps and can not enable use cases such as token transfer. In reality, *atomic swap* exchanges tokens between users; therefore, the users always require a counter-party willing to exchange tokens. An *atomic swap* protocol can also be extended to involve multiple users and networks [47].

Current research on decentralized finance (DeFi) protocol[5] looking at token swap in a decentralised environment. Since decentralization, process are governed by sophisticated smart contracts with the combination of automated market makers (AMMs) [49] and liquidity pools [50].

2) Lock and unlock protocol

This protocol is designed for systematically locking token value on one network and unlocking an agreed value on another network. In other words, tokens are immobilised on one blockchain by locking them into a custodian smart contract, ω and the custodian simultaneously unlocks the agreed value on the other network. This system refereed in Figure 4 is typically used to temporarily transfer a token's value to another network while the token's ID is retained on its original network. This is technically called a *peg system* where the ratio of value to lock and unlock is predefined, i.e. for locking c tokens, an agreed number of c' is minted by the unlocking process.

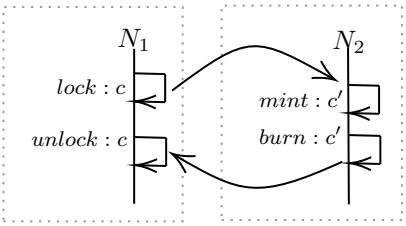

FIGURE 4. Lock and unlock process.

In a peg system, the peg-lock contract ω will keep the asset c and issue a temporary or agreed asset c'. This c' can be another value that the ω is holding. In general, this ω takes the token as collateral value and issues another value. Current research is exploring the possibility of implementing the integration of a peg-lock system through a federated approach instead of a centralised entity [13].

Currently, a good amount of Bitcoin[6] and Ether tokens has been locked and ported its value to other networks through pegged sidechains (ef. Section IV-E3). Generally, they are referred to as wrapped tokens to differentiate them from the same asset when they exist on their native network [51].

3) Burn and mint protocol

This protocol refereed in Figure 5 is useful when transferring tokens from one network to another in a permanent manner. That is in the burn process, the tokens are permanently destroyed from one network, and an agreed amount of tokens are minted on the other network. The process of burn consists of transferring the v to an address defined as burn-address β where the transferred asset, v becomes unspendable because the private key of the corresponding address is not known/ accessible. A burn-address given as β is an address to which one can send assets, but from which the value can never be recovered. Typically, burn addresses are verifiable but unspendable because those addresses do not have a corresponding private key and, therefore, the asset at those addresses are not spendable. The proof-of-burn protocol [43] proves that if the underlying cryptography is secure, then the probability of finding a private key for a given burn address is negligible.

The burn process has been identified as a cryptographic method to generate a proof to transfer, i.e., a token. It gives the guarantee that the token is permanently destroyed before being transferred onto another network [43]. We refer to our previous work for details on a burn-to-claim protocol [44], [45].

[5]https://blog.crypto.com/defi-swap-whitepaper/

[6]https://cointelegraph.com/news/one-percent-of-bitcoin-s-supply-has-been-locked-in-the-wbtc-protocol

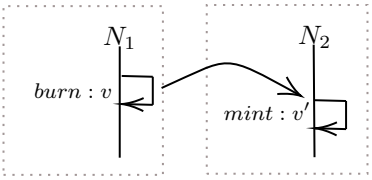

FIGURE 5. Burn and mint process.

4) Cross-blockchain message protocol

A cross-blockchain message protocol (message) aims to route data packets between a target blockchain networks or applications. This can be implemented by any distributed ledgers, provided they satisfy a small set of requirements. For example, the participating networks N_1 and N_2 had deployed the relevant smart contract, and the participants have the pre-knowledge of the address and its function. Let us assume network N_1's operation need some data from N_2. For that, a user $\mathcal{A}$ or a smart contract on N_1 send an instruction in the form of a message to invoke the smart contract function on N_2 and expects to receive a response message or perform an operation that updates the state of N_2. This scheme assumes the networks have similar data structures, messaging formats and shared security to interoperate with each other. Some industry projects define this scheme as Inter-blockchain communication (IBC)[7] and cross-chain interoperability (CCIP)[8] protocol.

E. CROSS-BLOCKCHAIN MECHANISMS

The technical solutions for integration are implementation through specific cross-blockchain mechanisms namely *Hashed Timelock Contracts*, *Relays* and *Sidechain*.

1) Hashed Timelock Contracts

The Hashed Timelock Contracts (HTLC) [52] technique allows fair exchange of tokens between users without going through a third party. Let us assume Alice ($\mathcal{A}$) wishes to swap her token c on N_1 with Bob's ($\mathcal{B}$) token c' on N_2. Both users agree on the amounts of value on each network to be exchanged and lock their tokens to a $HTLC$ smart contract (SC) using a secret key γ such that γ unlocks both the tokens. The protocol guarantees token swap if all parties conform to the protocol, else no conforming party lose their tokens. Therefore, this operation is called an $atomic\ swap$ [47].

As an example, $\mathcal{A}$ locks a transaction transferring c to $\mathcal{B}$ in a smart contract SC_{N_1} using a secret key γ that $\mathcal{B}$ does not know and publishes the hash of γ ($H(\gamma)$). Once $\mathcal{B}$ sees that $\mathcal{A}$ has locked his token c, $\mathcal{B}$ learns the $H(\gamma)$ and using that $H(\gamma)$, $\mathcal{B}$ locks a transaction transferring c' to $\mathcal{A}$ in SC_{N_2} such that $\mathcal{A}$ needs to reveal γ to SC_{N_2} to redeem c' token.

Given that both SC_{N_1} and SC_{N_2} coded to release token with same γ. If $\mathcal{A}$ who know γ first redeem c' locked by $\mathcal{B}$ by revealing γ, then $\mathcal{B}$ can use γ to redeem c locked by $\mathcal{A}$ thus both the parties get their desired token and the $atomic\ swap$ is marked completed.

In addition, $HTLC$ transactions are $time\ lock$ed in a way that *within the time period t redeem the tokens* or *reclaim their own token if they fail to provide γ within time period t* [47]. The $time\ lock$ time t is set up at the beginning based on the network latency and other factors that provide the users with a time span for sending transactions and the network to process and confirm the transactions. If $\mathcal{A}$ does not reveal γ in t time, the contract obligations are not fully met therefore the contract refunds the token to its original users. The desired property of $atomic\ swap$ is **atomicity** (cf Definition 5). In the context of this paper it assumes that if the first party (Bob) redeems the token, then the second party (Alice) redeems the token too.

2) Relays

is a mechanism that facilitates a blockchain network to verify other blockchain networks' data without relying on external third-party sources [36]. A relay system carries a smart contract and functions as a light client on a network, and records block header information from other networks. More specifically, relay is a mechanism where a light client node in N_1 actively listens to and keeps a subset of state ($\overline{Q}$) from N_2 such as block header, account balance etc. This $\overline{Q}$ of state information will be useful and enough for any node in N_1 to verify part of the transaction details belonging to N_2 using a standard verification process [10].

3) Sidechain

is a distinct network of blockchain attached to a parent blockchain network through a *peg* mechanism. The peg mechanism contract (ω) enables bidirectional transfer of token and other digital assets between a parent and $sidechain$ [53]. Technically, both the networks have a shared state, therefore, this mechanism enables the token to be locked on one network while it is being used on another network without compromising the security of the token.

Assume that N_1 is the main parent network and N_2 is a $sidechain$. The peg contract acts as the custodian of tokens from N_1 by locking them in this shared state of N_1 and N_2 to prevent double-spending, while they are being used in N_2. Technically, any node on network N_1 or N_2 can verify the peg lock status; therefore, tokens can not be misused in any of the networks.

While N_1 is referred as the parent network, and N_2 as the $sidechain$, those can be two *independent* or *dependent* networks. In the former, both networks can be sidechains of each other, with its own native token. In the latter, N_2 the $sidechain$ is dependent on N_1 the parent network and may not create its own token; instead, it can only derive token value from the parent chain N_1.

Conceptually, $sidechain$ networks are optimised for specific purposes to overcome limitations such as scalability and interoperability. Their advantage is that they can per-

[7] https://stargate.cosmos.network/
[8] https://blog.chain.link/introducing-the-cross-chain-interoperability-protocol-ccip/

form instant transactions at a higher speed and volume. The *sidechain* system also provides micropayments [54] from an interconnection chain, which realizes the off-chain exchange of value [55]. However, those solutions are focused on homogeneous networks and limited to a one-to-one relation [56]. Another trade-off of the *sidechain* implementation is that the vulnerability might increase in the main chain or other *sidechains* if there is a compromised *sidechain* in the network [56].

V. THE PROPOSED DESIGN DECISION FRAMEWORK

The integration process of blockchain networks involves many different aspects. In this section, we propose a cross-blockchain integration design decision framework (CBIDD) that details the step-by-step process of designing a cross-blockchain integration system. This CBIDD framework will be useful for blockchain developers/ project stakeholders in reviewing and identifying their interoperability requirements and determining the appropriate options and security assumptions.

A. INTEGRATION FRAMEWORK

The proposed CBIDD framework consists of five stages. Figure 6 illustrates the components and parameters of the proposed CBIDD framework and details the activities to be performed at each stage.

a: Stage 1: Select an application domain.

The first stage is to identify and select the application value type. The value type is determined based on the application's use case.

For a digital transaction application, if the value is in native coins, then the value type should be identified as crypto-coins (section III-A1), whereas if it is tokens created on top of the blockchain network, then, the value should be identified as crypto-assets (section III-A2). It is important to select a correct value type, because there are limitations on cross-blockchain trades of these values. Now, if the application is to perform arbitrary computation, the value type should be identified as data (section III-A3) since the application transfers instructions or information in the form of data.

b: Stage 2: Select an integration goal.

In this stage, the integration goal is determined based on the identified use case and its value type, the integration goal is determined. A common goal of interoperability is to enable cross-communication among different networks using a variety of technologies. For blockchain systems, the desired objective is to connect blockchain networks in order to facilitate cross-blockchain communication. In that context, interoperability aims to exchange information or instructions in the form of data. Further, with the exchanged data, a blockchain system can have functionality that falls into categories of *transfer* or *exchange* of value between blockchain networks.

Let us consider different selection situations: if the application domain is crypto-coins, the only possible trade option is exchange (section III-B1) between users within the same network because the value is native to the network therefore not transferable. If the selected domain is a crypto-asset, then the trade options are exchange or transfer (section III-B2). On the other hand, if the application domain is data, then it is exchange (section III-B3).

In a nutshell, stages 1 and 2 will help the designer or developer to identify the value type that the application is handling and also identify the possible trade options for that type of value. The next stage is to identify an appropriate integration approach.

c: Stage 3: Identify an integration approach.

Now, with the identified value type and goals of the application, at this stage, the process of integration has to be determined at this stage. It is important to choose the appropriate integration approach because this approach determines the security of cross-blockchain trade.

Interoperability requires the integration system to access networks beyond the boundaries of a single network. However, addressing this issue involves challenges related to preserving the properties of decentralisation. Consider the decentralised nature of the technology, where a number of nodes participating in the process to reach finality, these nodes must retain the same result. For that, the nodes must have or be given the same information to process the transaction. If the nodes are set to fetch data from other blockchain systems, the dynamic nature of values interferes with the consensus. Therefore, the integration process must be carefully designed in accordance with the system security. This leads to integration falling into either a *centralised* or *distributed* approach. The suitability of the approach depends on the type of application and the level of interoperability required.

- Centralised scheme - a single entity controls the integration process.
- Distributed scheme - a set of entities control the integration process.

In a centralised scheme, a single entity operates the integration process. This entity is responsible for the operations on both networks. In contrast, the distributed scheme is an improved version, where a set of entities control the integration, instead of a single entity. Therefore, compared with the centralised approach, distributed scheme achieves a DD, but it adds complexity by including the consensus of participants being needed. Although the centralised approach is easy to implement due to its simple design, the central entity needs to be trusted. While a centralised scheme provides some kind of efficiency, it is subject to a single point of failure and other centralisation issues. The developer team must analyse different types of integration schemes and ensure the right approach is selected.

A good way to define the DD of an integration approach is first to measure the ratio of the DD of the networks

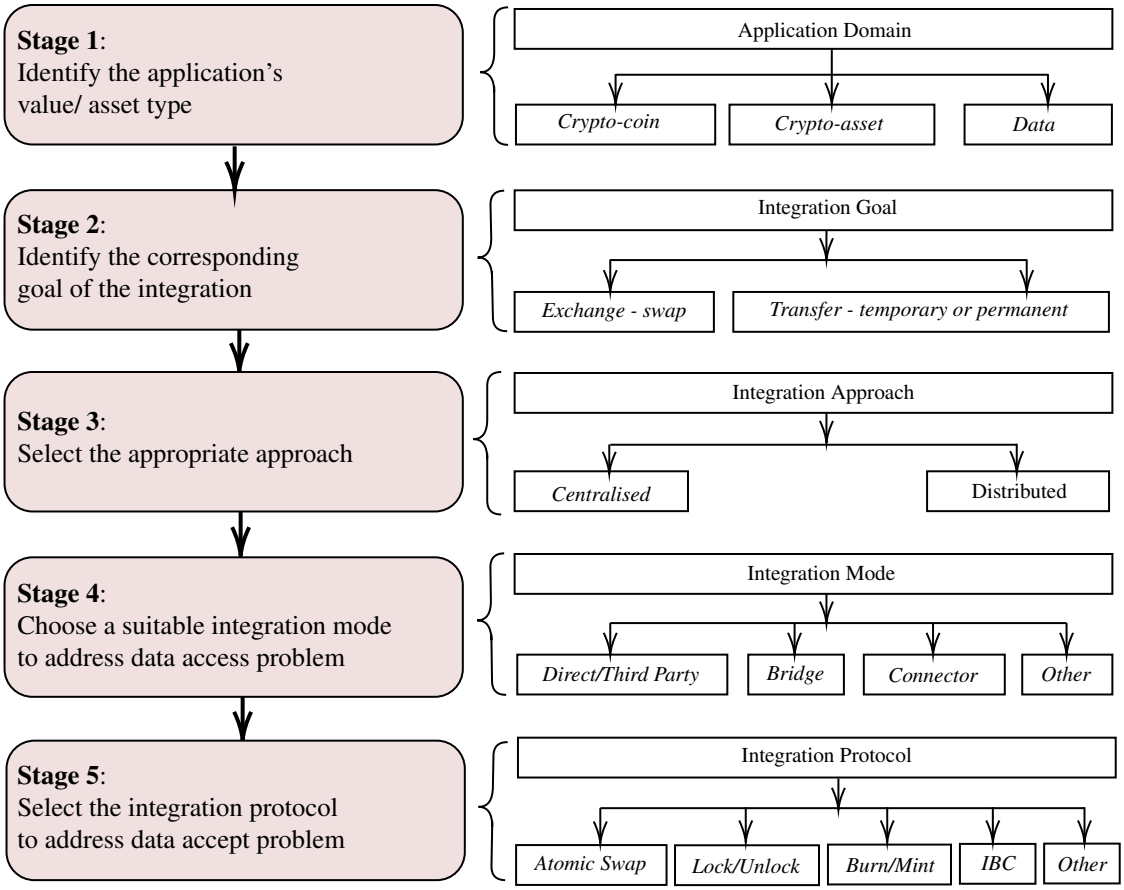

FIGURE 6. An overview of our cross-blockchain design decision framework.

that are connecting and use that ratio of nodes to build the cross-blockchain integration protocol. The actual number of integration nodes (for example, gateway node $\mathbb{G}$) may vary, but the validation of cross-blockchain data fetched by the integration nodes must be distributed to satisfy the system security.

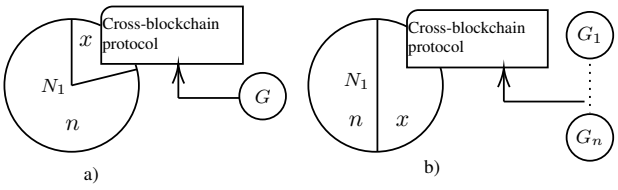

FIGURE 7. Example of integration distribution scheme

Figure 7 presents two different integration scenarios: a) a small number of validators with a single gateway node, b) a large number of validators with a number of gateway nodes. Regarding integration approaches, two things are to be decided: the number of integration nodes and the percentage of validator nodes. These are decided based on the security and performance requirement of the integrating system. For example, let us assume a network N_1 runs cross-blockchain process P using a single gateway node and 1% of the total validators and N_2 runs Q using 10 gateways with 50% of validators. In this case N_1 has a lower DD than N_2.

d: Stage 4: Select an integration mode.

The mode of integration acts as overriding the environment that facilitates direct communication between blockchain networks. The connected networks may have different underlying technologies, but the applications are designed for a specific purpose with a defined value type. Therefore, the integration process requires the design of physical connectivity between networks. The following four *modes of integration* are identified in this paper: *Third party*, *Bridge*, *Connector* and *Other*.

- Direct - network integrate through direct observing and following other network's consensus.
- Third party - aims to integrate networks through a trusted entity known as validator/notary (section IV-C2). In a federated system, the validation is done by a group of validators.
- Bridge - aims for a built-in integration through gateway nodes that relay the data between networks (section IV-C3).
- Connector - aims for connecting several networks through a hub (section IV-C4).

- Others - aim to connect off-line data resources such as oracles or APIs (section IV-C5).

For any cross-blockchain use cases, an integration mechanism builds a communication link between networks, and therefore, security assumptions of these *modes of integration* are essential to analyze the potential security risks and mitigate the vulnerabilities. For example, in a bridge mode, the developers and security architects need to analyze the risks of the bridge technology. The gateway in a bridge could be an easier target for single point of failure attacks. Table 1 summarises the security assumptions of various integration modes (cf. Section VI-D).

e: Stage 5: Select an integration protocol.

Integration modes help with physical connectivity between networks, but to address the data acceptance problem (cf. Section II-B1), the networks need to address the data validation and verification problem. Integration systems aim to address this problem by implementing different protocols to accomplish the composability of cross-blockchain transactions between integrating networks. Consequently, there exist a range of cross-blockchain protocols such as *atomic swap*, *lock/unlock*, *burn/mint* to be used based on the use-cases [10], [12], [13], [44].

- *atomic swap* - token swap between users in different network that guaranteed either both transfers happen or neither of them happens (Section IV-D1).
- *lock/unlock* - temporary transfer of token value from one network to another with a provision of retaining back to the original network (Section IV-D2).
- *burn/mint* - permanent transfer of value from one network to another (Section IV-D3).
- Others - custom protocols built for message exchange (Section IV-D4).

Cross-blockchain protocols are selected based on the type of application. For example, a crypto-coin exchange can only use *atomic swap* or *lock/unlock* protocol since the value is not transferable from the source network. With the *atomic swap* protocol, the token can be swapped between users or with *lock/unlock* protocol, the value in locked in the source network and, with that proof, unlocks an agreed value in the destination network. Whereas a crypto-asset trade, can use *atomic swap*, *lock/unlock* or *burn/mint* protocol depending on the use-case. If it is exchange or swap, the *lock/unlock* or *atomic swap* protocol has to be used, whereas the *burn/mint* protocol needs to be used in case of transfer. In *burn/mint*, the protocol permanently destroys the value and, with that proof, mints the agreed amount of value on the other network.

Given this integration design decision framework, let us examine a practical example of how this CBIDD framework can be applied in a real-world example.

B. A PRACTICAL EXAMPLE OF CROSS-BLOCKCHAIN INTEGRATION

A blockchain-based solution has been proposed for a wide range of applications. In [57], the authors propose a public blockchain-based solution for the digitisation of land ownership in Bangladesh, which embodies the characteristics of public chain data synchronisation and transparency, easy access, and immutable record management. Even though this project covers the technical requirements for a land title management system, there was no discussion about interoperability. To leverage the maximum benefits of the system, it must provide interoperability among different stakeholders.

Let us consider this application use case and see how the proposed CBIDD framework helps the developer/designer to make decisions on selecting the appropriate mechanisms based on the type and nature of the application. This system mainly deals with the timestamp logger of current land ownership and ongoing land handovers at a very abstract level.

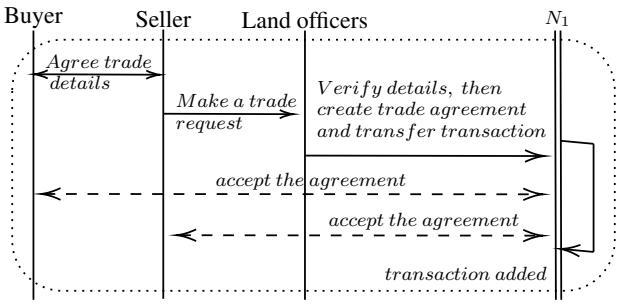

FIGURE 8. Land ownership transfer process

a: Current system.

The participants in this system are seller $\mathcal{A}$ and buyer $\mathcal{B}$ - they are normal users who have an account in the system. To create an account, a user has to go to the land office or notary service provider and verify their identification. Once verification is completed, the land officer registers the new user with the system. The Land officers are government employees with special permission to write to the ledger. Trusted public or private organisations such as universities, courts, post offices, banks make up the mining nodes in this system.

The user registration and land purchasing process are encoded in the form of a smart contract deployed on a public blockchain. A simplified version of a land trade is shown in Figure 8. The seller $\mathcal{A}$ and buyer $\mathcal{B}$ agree on the trade, then they communicate with the land office, who will verify details. If $\mathcal{A}$ has the legal ownership, and $\mathcal{B}$ has the financial balance, then the land officer logs the trade agreement transaction for both parties to act on. Once the agreement is accepted, and the financial payment has settled, the transaction executes the transfer of ownership from $\mathcal{A}$ to $\mathcal{B}$ and updates the details on the blockchain.

Let us assume the developers of this project decided to incorporate interoperability features into this system.

- Stage 1 - *identify the value type* - As this system registers land title details in a digital form, the value type is identified as asset-tokens. Land IDs are minted by the land officers and registered against their owner's name.

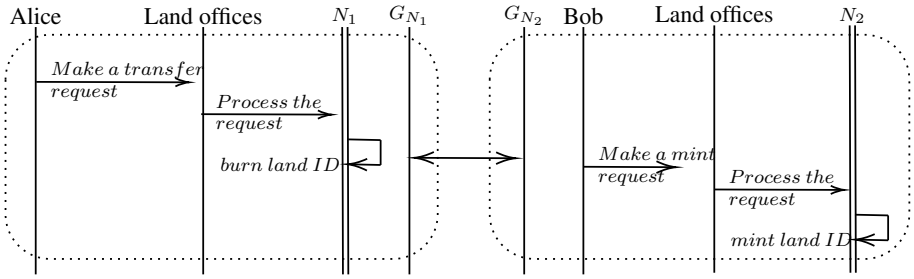

FIGURE 9. Overview of cross-blockchain land ID transfer process

- Stage 2 - *define the goal* - let us assume the goal is permanent transfer of land ID from one network to another.
- Stage 3 - *select the integration approach* - Since there are trusted entities such as land officers involved in the approval process, we assume a similar approach for integration is suitable such as a centralised approach.
- Stage 4 - *choose the integration mode* - In this use case, let us aim for connecting two networks together and for that a bridge mode is more suitable. If there were a number of networks to be connected together then the connector mode would be a good option.
- Stage 5 - *select an integration protocol* - Since it is a permanent transfer, *burn/mint* will be a suitable protocol.

b: Integration process.

Figure 9 provides an overview of the integration process between networks N_1 and N_2. Users and land officers are the same as in the original design. Alice $\mathcal{A}$ on N_1 seeks to transfer ownership of her land to Bob $\mathcal{B}$ located on a different N_2. Let us assume that both the networks adopt the same type of value (land ID) and cross-blockchain protocol (burn/mint), and within the mining nodes, some nodes act as gateway nodes to make a bridge between the networks.

$\mathcal{A}$ (owner of the land) makes a land transfer request to the land office. The land officer checks the details, approves and processes the transaction. This transaction will burn the land ID to a burn address and record the burn details to N_1. The gateway nodes pass the proof-of-burn details from N_1 to N_2. Using the burn transaction as proof, $\mathcal{B}$ makes a mint request to the land office on N_2. Land officers on N_2 check the proof and issue a new land ID to $\mathcal{A}$ on N_2.

VI. CASE STUDY AND SECURITY ANALYSIS

Since blockchain is an emerging technology, cross-blockchain integration needs to be validated in real-life application scenarios. This section presents the scenario of cross-blockchain trade among various networks working in the integration environment.

Technically, a secured blockchain C can validate a transaction if its state Q has the relevant data on-chain to trust the transactions. Integration solutions require networks to authenticate data from other networks, resulting in the data acceptance problem (subsection II-B1). Therefore, the integration process should introduce security to permit one network to trust the data from the other networks. We propose three forms of security model: Third party security, distributed security, and shared security summarised below.

- Third party security - An external entity guaranteeing the authenticity of the transaction.
- Distributed security - Based on the trustworthiness of integration nodes in the distributed network.
- Shared security - Independent networks leverage the security through a shared state or set of validators.

A. CRYPTO-COIN APPLICATION

Application scenarios. Let us assume $\mathcal{A}$ and $\mathcal{B}$ are users on two different networks N_1 and N_2. $\mathcal{A}$ has crypto-coin x on network N_1 and $\mathcal{B}$ has crypto-coin y on network N_2. In this example, shown in figure 10 the value of crypto-coins are not transferable because they are native to their networks. Therefore, they can only be swapped between users within a network.

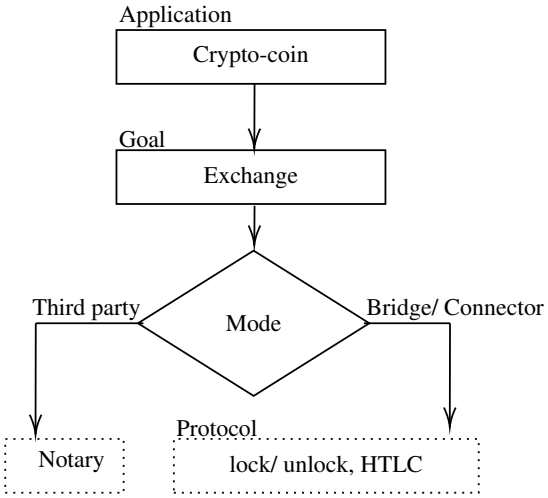

FIGURE 10. Crypto-coin application selection through framework

a: Third party mode.

The first scenario is that the user $\mathcal{A}$ wants the value the user $\mathcal{B}$ has in N_2 and $\mathcal{B}$ wants the value that $\mathcal{A}$ holds in N_1. $\mathcal{A}$ and

$\mathcal{B}$ do not trust each other; therefore, they process this trade through a third party security mode. We define $\mathcal{C}$ as the third party who has accounts on both networks and is willing to perform the exchange service for a fee. The process shown in Figure 11 is both $\mathcal{A}$ and $\mathcal{B}$ transfer their values to $\mathcal{C}$ first $\mathcal{A} \rightarrow \mathcal{C} : x$, and then $\mathcal{C} \rightarrow \mathcal{A} : y$, finally, $\mathcal{C}$ gives them relevant coins on their networks $\mathcal{B} \rightarrow \mathcal{C} : y$ and $\mathcal{C} \rightarrow \mathcal{B} : x$.

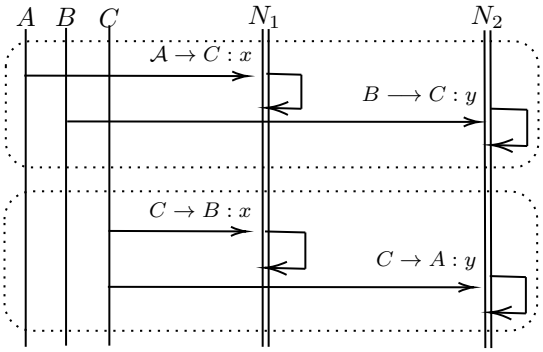

FIGURE 11. Third party exchange

b: Bridge mode.

Now let us assume an exchange scenario where the user $\mathcal{A}$ on N_1 only wanted to temporarily transfer the value v he owns in N_1 to N_2. This can be done through a *lock/unlock* system where $\mathcal{A}$ locks v in a peg contract and unlock v' in N_2. This is a use case for a crypto-coin (native coin) to be used as a lateral deposit and the issue of an agreed amount of value on another network for a period of time. Once the time or purpose is fulfiled, the temporarily issued v' must be destroyed to unlocks the original token locked in N_1. The role of the bridge node $\mathbb{G}$ is to pass $lock$ and $unlock$ information between N_2 and N_1 shown in Figure 12.

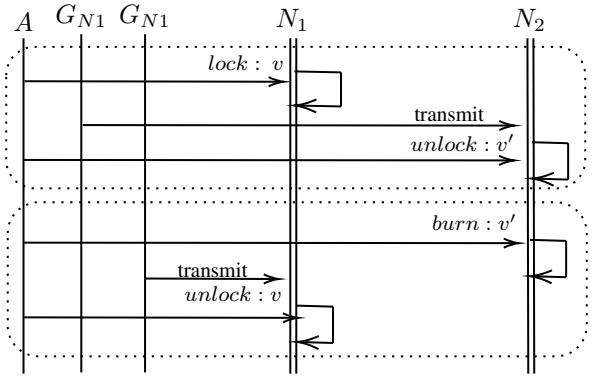

FIGURE 12. Bridge mode - message flow with gateway

We assume this network includes the bridge $\mathbb{G}$. $\mathbb{G}_{N_1}$ and $\mathbb{G}_{N_2}$ are bridge nodes deployed on the respective networks. On N_1, the user $\mathcal{A}$ locks the v in a transaction T_{x1} and the bridge node $\mathbb{G}_{N_1}$ broadcasts this information to N_2. $\mathbb{G}_{N_2}$ then updates this information in its network N_2. This will address the problem of *data access* (subsection II-B2) and with the data available on N_2 solves the problem of *data acceptance*. Now through the unlock protocol, $\mathcal{A}$ can mint v' on N_2 within the agreed period of time. After the time, $\mathcal{A}$ burns v' and the bridge will add this information to N_1, which allows $\mathcal{A}$ to unlock the original token.

c: Connector mode.

The design goal is to connect several networks. Let us assume $\mathbb{U}$ as the connector and N_1, N_2 and N_3 are the connected networks.

In case of exchange, $\mathcal{A}$ transfers his value x to $\mathcal{B}$ in network N_1 and $\mathcal{B}$ transfers y in network N_2. Here, $\mathbb{U}$ acts as the intermediary to oversee the exchange and to settle any issue. Additionally, $\mathbb{U}$ can act as the liquidity provider[9] and accept value from $\mathcal{A}$ and $\mathcal{B}$ and then issue relevant coins on each network.

In case of temporary exchange, the user $\mathcal{A}$ locks the v in the shared state of N_1 and $\mathbb{U}$. Then, to unlock, $\mathcal{A}$ mint v' within the shared state of N_2 and $\mathbb{U}$. In this case, the *lock/unlock* information is exchanged between N_1 and N_2 through the shared security of $\mathbb{U}$, N_1 and N_2. Let us assume that N_3 can not share its state with $\mathbb{U}$, such networks can be connected through bridges, which will pass the *lock/unlock* information between N_3 and $\mathbb{U}$.

B. ASSET-TOKEN APPLICATION

In this scenario, it is assumed that asset-tokens are created on top of a blockchain protocol and the application ledger will track their creation, distribution and balance. As shown in Figure 13, asset-tokens can have applications of exchange or transfer. An asset-token exchange process will be the same as the crypto-coin exchange. Therefore, here we explore the transfer process only. These are situations of permanent transfer using *burn/mint* protocol.

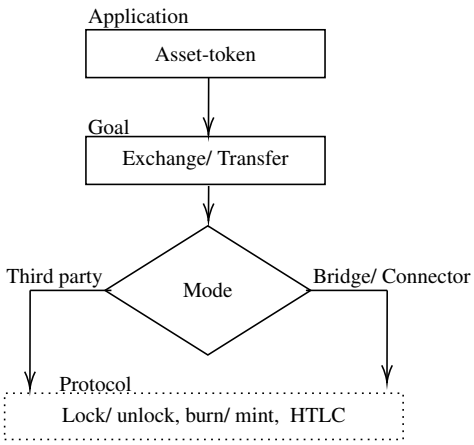

FIGURE 13. Asset-token application selection through framework

Application scenario. Let us assume an echo system of networks N_1 and N_2, users $\mathcal{A}$ and $\mathcal{B}$ are participants and they

[9] for example Uniswap https://uniswap.org/

engage in a transfer agreement of tokens. $\mathcal{A}$ has asset-token q in network N_1 and $\mathcal{B}$ has asset-token r in network N_2.

a: Third party mode.

In third party mode, $\mathcal{A}$ and $\mathcal{B}$ use $\mathcal{C}$ to authenticate the transfer of assets across the networks. First, user $\mathcal{A}$ burns the asset q he owns on N_1 through a transaction. Then $\mathcal{C}$ authenticates this proof-of-burn to N_2. With that proof, $\mathcal{B}$ mints r on N_2.

b: Bridge mode.

In the case of bridge mode, the bridge acts as the integration node to exchange the *burn/mint* information between networks.

c: Connector mode.

The connector's hub network $\mathbb{U}$ will take part in the operations of cross-blockchain transactions. The user $\mathcal{A}$ burns q in the shared state of N_1 and $\mathbb{U}$. Then, $\mathcal{A}$ mints r within shared state of N_2 and $\mathbb{U}$. The *burn/mint* information exists within the shared security of $\mathbb{U}$, N_1 and N_2. In this case, N_3 can not share its state with $\mathbb{U}$, but N_3 can be connected through a bridge, which will pass the *lock/unlock* information between N_3 and $\mathbb{U}$.

C. DATA EXCHANGE APPLICATION

With data exchange, the integration is based on the assumption that the participating networks N_1 and N_2 had deployed relevant smart contracts, and the participants have pre-knowledge of the addresses and their functions.

Application scenario. Let us assume N_1's operation needs some data from N_2. The process is, a user $\mathcal{A}$ or a smart contract from N_1 invokes a smart contract function on N_2 and expects to receive a response or perform an operation that updates the state of N_2.

Since most of the blockchain systems are passive and unable to create a data validation proof on another blockchain network, arbitrary data exchange is more challenging than other application domains. However, the use-cases enabled by arbitrary data exchange can be specific to the application, such as verifying token balance or running a function. Therefore, there are application-specific solutions where the application itself should be able to identify the correct source of data.

The projects such as Cosmos and Polkadot are built with inbuilt cross-chain messaging mechanism. In Cosmos a messaging protocol called IBC is used to pass message between networks of Cosmos blockchain. Polkadot has a different model they use substrate and a XCMP protocol to send message between different substrates.

D. SECURITY ANALYSIS OF INTEGRATION MODES

Cross-blockchain technology invariably inherits the security requirements of blockchain in general. While the decentralisation characteristic of blockchains makes independent networks resistant to attacks, the integration infrastructure is the weakest point to attack. Integration solutions lead to a specific set of node(s), which integrate through interfaces and communicate with the networks. While these choices introduce interoperability, they violate the principle of decentralisation, resulting in trade-offs in security. Vulnerability and security issues of various components of the blockchain have been comprehensively analysed in the given literature [34], [58], [59]. In this research, we consider the security provided by the integration system to makes the cross-blockchain data trustworthy, assuming that the participating network components and protocol are safe. We summarise the security assumptions of various integration scenario in Table 1.

1) Direct mode

Let us assume that direct integration modes use a full verification [29]. This method adapts cross-blockchain verification by directly following the consensus of the connected blockchain. Therefore, the cross-blockchain interaction security is the same as the on-chain security.

2) Third party mode

Third party security is provided by an external entity that guarantees the authenticity of the transaction. Third party mode aims to address trust issues that arise from centralisation through a group of semi-trusted notaries. Thus, this mode assumes a weaker trust model and can often withstand the adversarial behaviour of a fraction of the notaries.

A trade using a third party mode is completed as follows: Two parties, $\mathcal{A}$, and $\mathcal{B}$, agree on a trade transfer of values through authority service $\mathcal{C}$. When $\mathcal{C}$ received both values, it finishes the exchange by transferring the appropriate value to the other party. In this approach, $\mathcal{C}$ holds temporary ownership of the value to be traded. In this trade, relying on $\mathcal{C}$ removes counter-party risk for $\mathcal{A}$ and $\mathcal{B}$, but it requires both $\mathcal{A}$ and $\mathcal{B}$ to trust that the intermediary $\mathcal{C}$ will not default or compromise their value. This can be easily incorporated into the permission system, but it is a matter of trusting $\mathcal{C}$ to make the trade for a permissionless system. There have been various efforts to address the centralised trust issue through voting, staking etc. In [60] cross-blockchain transaction are proposed though notary and aim to address the security using advance hash locking system.

3) Bridge mode

A Bridge interface aims to automate inter-connectivity through a set access point to fetch and emit messages, but these are limited to passive operation because the incoming and outgoing messages may not require cross-platform checking or consensus. Consequently, it is possible that the message included are not a valid transaction.

Also, interactions are through bridge interfaces, which may be a single node, thus becoming centralised. Therefore, the bridge system can expand a blockchain's capabilities,

TABLE 1. Summary of security assumptions

Value	Goal	Approach	Mode	Protocol	Security assumptions
Crypto-coin	Exchange	Centralised	Direct	*atomic swap*	On-chain trust.
		Centralised	Third party	*atomic swap*	Single trusted entity.
		Distributed	Third party	*atomic swap*	Group of semi-trusted entities.
		Distributed	Bridge	*atomic swap*	Trustworthiness of gateway node.
		Distributed	Connector	*atomic swap*	Shared state or validates.
Crypto-coin	Temporary transfer	Centralised	Third party	$lock/unlock$	Single trusted entity.
		Distributed	Bridge	$lock/unlock$	Trustworthiness of gateway node.
		Distributed	Connector	$lock/unlock$	Shared security.
Asset-token	Exchange	Centralised	Direct	*atomic swap*	On-chain trust.
		Centralised	Third party	*atomic swap*	Single trusted entity.
		Distributed	Bridge	*atomic swap*	Trustworthiness of gateway node.
		Distributed	Connector	*atomic swap*	Shared state or validates.
Asset-token	Permanent transfer	Centralised	Third party	$burn/mint$	Single trusted entity.
		Distributed	Bridge	$burn/mint$	Trustworthiness of gateway node.
		Distributed	Connector	$burn/mint$	Shared state or validates.
Data	Exchange	Centralised	Third party	*message*	Single trusted entity.
		Distributed	Bridge	*message*	Trustworthiness of gateway node.
		Distributed	Connector	*message*	Shared state or validates.

but there is a need to address the trust issue. In Allbridge[10] project the protocol use a *verifiable action approval* process for guardians to verify and signing message. A signed wormhole message includes sufficient metadata information to interpret the message. A valid message must be signed by super majority guardians and this which act as a cross-blockchain proof to be posted on the other networks.

Depending on the number of bridges in the integration system, security of the cross-blockchain transaction is safeguarded by degree of decentralisation (DD). The bridges may use a membership schema or a staking model. In any case, if the bridge provides wrong information, there are provisions to punish them either by slashing their deposited stakes or by removing them from the network.

Let us assume connecting multiple networks using a bridge. You need to run a number of gateway nodes for all the networks you want to connect. It will be expensive and very intensive in terms of hardware and maintenance; therefore, connector mode may be a better option for multiple network integration.

Allbridge[11] project working on blockchain bridge that can work with EVM[12] compatible or non-EVM[13] compatible networks. The project supports a number of token exchange[14] and transfer type sucg as native token to wrapped tokens and wrapped tokens to native.

4) Connector mode

Compared to the other modes, the connector's scope is to connect multiple networks and transfer messages between networks. The hub of the connector consists of nodes that maintain a separate network with the relevant capability of blockchain and other services (as per the platform). In other words, the connector's hub $\mathbb{U}$ itself is a blockchain with its own security that is backed by a censorship-resistant network. We assume that the hub is secure and able to provide a trustworthy environment for the participating networks.

The hub may leverage security of the connected network in different ways. For example, let us assume that N_1 and N_2 are two independent networks and a cross-blockchain Tx $N_1 \rightarrow N_2 : v$ from N_1 needs to go to N_2.

Option 1: shared security.

This happens in two stages with two transactions. First, between N_1 and $\mathbb{U}$ where N_1 and $\mathbb{U}$ share some part of state $Q_{N_1,\mathbb{U}}$ and, through their shared state Tx_1 brings the message to $\mathbb{U}$. The second stage happens between the shared state of $\mathbb{U}$ and N_2 and, through that shared state $Q_{\mathbb{U},N_2}$, Tx_2 updates the message to N_2.

Option 2: gateway. Similarly, this is also a two-stage process through one or more gateways. First, N_1 connects to $\mathbb{U}$ through gateway $\mathbb{G}_{N_1}$ and processes Tx_1 which brings the message to $\mathbb{U}$. We assume that these networks will have their own mechanism/criteria to select $\mathbb{G}$ nodes for their own network. Once Tx_1 is confirmed in $\mathbb{U}$ through $\mathbb{G}$, the protocol routes or processes Tx_2 to N_2, bringing the message to N_2.

5) Other - oracle mode

Blockchain based oracle [61] are entities that provide access to external data for a blockchain system. The problem here is to ensure the authenticity and integrity of the data because we have to trust the oracle that it behaves honestly [62]. To achieve better resiliency and to increase trust, oracles can adopt a decentralised architecture [63]. Technically, it is difficult to get data into a decentralised blockchain network in a non-centralized way. This is perhaps the most important problem in the blockchain. Without a secure and reliable

[10]https://docs.allbridge.io
[11]https://docs.allbridge.io
[12]Like Ethereum, Polygon, BSC
[13]like Solana, Terra
[14]https://docs.allbridge.io/allbridge-overview/networks-and-tokens

way to get data into smart contracts running on blockchain, the security and reliability of the blockchain and, thus the advantages of the entire system, are lost.

E. DISCUSSION

Interoperability is a broad problem in the domain of information systems. The scope of the work described here is focused around interoperability for blockchain-based technology. Current research on interoperability in blockchain technology addresses the challenges of interpreting and exchanging values through various integration systems. However, the security assumptions when integrating through different integration systems have not been identified and discussed. In [64] propose interoperability for the users to switch between different blockchain services. There are various metrics identified in this paper for the users to consider. However, the security assumptions of switching, which will be done through some form of integration system, is as crucial as the security of the system.

Let us assume an interoperable ecosystem where various blockchain networks operate in cooperation with each other to suit the needs of varying use-cases. A defined integration system and its process will address cross-blockchain settlement for each use case. The settlement mechanism and process will vary based on the selection made for that integration system. The key to the settlement is that the integration system will serve as the connection layer for all those connected networks. The settlement layer provides security and objective finality for transactions that happen on the connected networks. It is important to note that the security guarantee of the integration process on the settlement layer is dependent on the choice of the integration system. Therefore, choosing the right integration system for an interoperable ecosystem is important.

Even though interoperability is made possible through integration systems, this solution leads to a specific set of node(s), integrated through interfaces for cross-blockchain settlement, resulting in trade-offs in security. In reality, not every integration process needs to prioritize absolute decentralisation; rather, it can prioritize the application's usability with varying degrees of decentralisation that trade-off security.

The application must choose the level of decentralisation that provides the best protection for the key capabilities offered by the network. Thus, the solutions are leading to a mix of centralised and distributed integration methods for the next generation of blockchain networks.

We conclude that, although interoperability offers a wide range of functionalities, there exist security assumptions, and there is a trade-off at the cost of security. Therefore, it is important to understand the security assumptions when integrating through different integration systems. Hence, the cross-blockchain integration framework we propose is useful for deciding which integration architecture to choose and understanding the trade-off of each solution.

VII. RELATED WORKS

In [6] the authors propose a framework for inter-connections networks of blockchain through an InterChain. The InterChain has its own validates, and SubChains networks are connected to InterChain through gateway nodes. In [40] the authors discuss blockchain interoperability through the design philosophy of the Internet architecture. At the mechanical level, the internet routes message packets through its router network. Similarly, for blockchain suggest using gateways to rout messages between networks. In [65] the authors proposed a solution using a proof-of-burn protocol that utilises bridge and relay mechanisms to verify and inclusion of cross-blockchain transactions in the networks. In [66] the authors present an application-based solution for cross-blockchain communication using DApps to process the cross-blockchain request and incentivised verifier nodes to maintain the integrity of shared data.

In [7] the authors presented a solution for asset sharing between inter-firm alliance chains and private chains. Users on both the sender and recipient chain interact with the alliance chain, prove their identity and obtain a certificate. Once the users initiate a cross-blockchain transfer request, the alliance chain confirms the users' ownership of the asset and the transfer of assets through the cross-blockchain interaction process the transfer of assets. Authors in [15] proposed a solution for permissioned networks using the publish/subscribe pattern where the source network emits transaction events, and the destination network subscribe to these events to get the information. The source and destination must enrol as a publish and subscribe through a broker network. The broker network keeps a record of the data being transferred between blockchain networks.

Authors in [24], proposed an interoperability architecture for permissioned networks. The networks cross-communicate through trusted relays using a communication protocol that is structured to provide details about the network and ledger. Each interoperating node deploys a special system contract that enforces the network rules about data exposure and acceptance. A Ripple [67] network deals with cross-blockchain transactions for finical institutions. Ripple project explored the possibility of the financial settlement between banks with cryptocurrencies through interconnected networks. The Cosmos [9] project use an Inter-Blockchain Communication protocol to communicate with networks that are connected via a hub and zone model. In Polkadot [8], all of the parachains and external blockchains connect to the relay chain (the main chain) via a bridge.

An early research report [10] by Buterin introduce chain interoperability schemes and techniques of notary, side chain, relay and hash-locking. In effect, a notary scheme simply relays on trusted intermediaries(s) to provide information about one network to another. Whereas with relay techniques, instead of relying on trusted intermediaries, the network keeps some part of block data to validate the state from the other network. The sidechain based interoperability is generalised to peg system, a mechanism for transferring assets

between the main chain and the sidechain [68]. Built on top of these solutions, other research emerged with distinct classification.

A review paper by Belchior et al. [12] classified blockchain interoperability into categories of cryptocurrency-directed, blockchain engines, and blockchain connectors. The cryptocurrencies directed approach aims to deal with the transaction of cryptocurrency tokens created and exists within the network. The blockchain engines consist of a middleware network through which multiple networks interoperate. And with blockchain connectors, cross-blockchain transactions are routed to the network by trusted escrow such as relays.

Gang [13] categories them as chain-based, bridge-based and dApp-based interoperability. The chain-based interoperability focus on applications of cryptocurrencies to perform token swaps between networks users. Bridge-based solution targets the implementation of a "bridge" as a connection component between networks of blockchain. dApp-based solutions looking for application to interoperate between networks. Qasse et al. [14] categories as sidechains solutions, blockchain router, smart contracts, and industrial solutions. In [15], the authors classified blockchain interoperability into three types: cryptocurrency directed (value transfer in the form of digital assets), blockchain engines (integration systems for data flow), and blockchain connectors (using an intermediate blockchain to act as a trusted relay).

[11] three distinct patterns of corss-chain technology a manual asset exchange, notary schemes, and relays.

VIII. CONCLUSION

Currently, there are a number of constraints in cross-blockchain integration for blockchain networks. Those constraints depend on the application and its settings. This has led to a complex formation of different solutions, which have been reported in the literature in an isolated manner. In this paper, we have reviewed those application solutions and (re)identified appropriate solutions for each scenario. The proposed CBIDD framework makes it easier for enterprises to design and integrate different blockchain applications and to evaluate their security assumptions more accurately.

a: Future work.

In this research, we focused on layer 1 solutions to trade value between networks. However, there are possibly application level, layer 2 solutions that solve for scalability and interoperability. We intend to make a comparison between layer 1 and layer 2 solutions in future work.

b: Broader impact.

Findings published by PWC[15] estimate that blockchain technology has the potential to boost global GDP by US$1.76 trillion over the next decade. The Gardner[16] report says, by 2023, blockchain will support $2 trillion worth of goods and services annually. There is increasing interest in new technology that can integrate trust into processes without depending on intermediaries. However, the key capability to move digital assets from one blockchain to another without intermediaries is still an open question[17]. Given the urgent need of interoperability, our findings and proposed CBIDD framework will add significant value to the blockchain community by providing the necessary tools and resources. Those identified underlying security assumptions, with their integration solutions, will help organisations to design appropriate integration systems fit for purpose.

[15]https://image.uk.info.pwc.com/lib/fe3111707564047570 1c74/m/2/434c46d2-a889-4fed-a030-c52964c71a64.pdf

[16]https://www.gartner.com/en/information-technology/insights/blockchain

[17]https://www.finextra.com/blogposting/18972/blockchain-and-interoperability-key-to-mass-adoption

REFERENCES

[1] Zibin Zheng, Shaoan Xie, Hongning Dai, Xiangping Chen, and Huaimin Wang. An overview of blockchain technology: Architecture, consensus, and future trends. In 2017 IEEE international congress on big data (BigData congress), pages 557–564. IEEE, 2017.

[2] Emmanuelle Anceaume, Antonella Del Pozzo, Romaric Ludinard, Maria Potop-Butucaru, and Sara Tucci-Piergiovanni. Blockchain abstract data type. In The 31st ACM Symposium on Parallelism in Algorithms and Architectures, pages 349–358, 2019.

[3] Jack Thomas. Blockchain interoperability remains a critical missing puzzle piece. https://journal.binarydistrict.com/blockchain-interoperability-remains-a-critical-missing-puzzle-piece/, accessed date: 10 October 2019, 2019.

[4] Pascal Lafourcade and Marius Lombard-Platet. About blockchain interoperability. Information Processing Letters, page 105976, 2020.

[5] Stefan Schulte, Marten Sigwart, Philipp Frauenthaler, and Michael Borkowski. Towards blockchain interoperability. In International Conference on Business Process Management, pages 3–10. Springer, 2019.

[6] Donghui Ding, Tiantian Duan, Linpeng Jia, Kang Li, Zhongcheng Li, and Yi Sun. Interchain: A framework to support blockchain interoperability. Second Asia-Pacific Work. Netw, 2018.

[7] Jianbiao Zhang, Yanhui Liu, and Zhaoqian Zhang. Research on cross-chain technology architecture system based on blockchain. In International Conference in Communications, Signal Processing, and Systems, pages 2609–2617. Springer, 2019.

[8] Gavin Wood. Polkadot: Vision for a heterogeneous multi-chain framework. White Paper, 2016.

[9] Jae Kwon and Ethan Buchman. Cosmos: A network of distributed ledgers. URL https://cosmos. network/whitepaper, 2016.

[10] Vitalik Buterin. Chain interoperability. R3 Research Paper, 2016.

[11] Niclas Kannengießer, Michelle Pfister, Malte Greulich, Sebastian Lins, and Ali Sunyaev. Bridges between islands: Cross-chain technology for distributed ledger technology. In Proceedings of the 53rd Hawaii International Conference on System Sciences, 2020.

[12] Rafael Belchior, André Vasconcelos, Sérgio Guerreiro, and Miguel Correia. A survey on blockchain interoperability: Past, present, and future trends. arXiv preprint arXiv:2005.14282, 2020.

[13] Gang Wang. Sok: Exploring blockchains interoperability. IACR Cryptol. ePrint Arch., 2021:537, 2021.

[14] Ilham A Qasse, Manar Abu Talib, and Qassim Nasir. Inter blockchain communication: A survey. In Proceedings of the ArabWIC 6th Annual International Conference Research Track, pages 1–6, 2019.

[15] Sara Ghaemi, Sara Rouhani, Rafael Belchior, Rui S Cruz, Hamzeh Khazaei, and Petr Musilek. A pub-sub architecture to promote blockchain interoperability. arXiv preprint arXiv:2101.12331, 2021.

[16] Leslie Lamport. Proving the correctness of multiprocess programs. IEEE transactions on software engineering, (2):125–143, 1977.

[17] David P Reed. Implementing atomic actions on decentralized data. ACM Transactions on Computer Systems (TOCS), 1(1):3–23, 1983.

[18] Marianna Belotti, Stefano Moretti, Maria Potop-Butucaru, and Stefano Secci. Game theoretical analysis of Atomic Cross-Chain Swaps. PhD thesis, Caisse des dépôts-Institut pour la recherche et Banque des territoires . . . , 2019.

[19] FB Vernadat. Interoperable enterprise systems: architectures and methods. IFAC Proceedings Volumes, 39(3):13–20, 2006.
[20] Lawrence E Whitman, Danny Santanu, and Hervé Panetto. An enterprise model of interoperability. IFAC Proceedings Volumes, 39(3):609–614, 2006.
[21] Peter Wegner. Interoperability. ACM Computing Surveys (CSUR), 28(1):285–287, 1996.
[22] David Chen, Guy Doumeingts, and François Vernadat. Architectures for enterprise integration and interoperability: Past, present and future. Computers in industry, 59(7):647–659, 2008.
[23] Maxwell William. Erc-20 tokens, explained. https://ethereum.org/en/developers/docs/standards/tokens/, accessed December 2021, 2018.
[24] Ermyas Abebe, Dushyant Behl, Chander Govindarajan, Yining Hu, Dileban Karunamoorthy, Petr Novotny, Vinayaka Pandit, Venkatraman Ramakrishna, and Christian Vecchiola. Enabling enterprise blockchain interoperability with trusted data transfer (industry track). In Proceedings of the 20th International Middleware Conference Industrial Track, pages 29–35, 2019.
[25] Thomas Hardjono, Alexander Lipton, and Alex Pentland. Towards a design philosophy for interoperable blockchain systems. arXiv preprint arXiv:1805.05934, 2018.
[26] Gang Wang and Mark Nixon. Intertrust: Towards an efficient blockchain interoperability architecture with trusted services. Cryptology ePrint Archive, 2021.
[27] Babu Pillai, Kamanashis Biswas, and Vallipuram Muthukkumarasamy. Cross-chain interoperability among blockchain-based systems using transactions. The Knowledge Engineering Review, 35, 2020.
[28] Gewu Bu, Riane Haouara, Thanh-Son-Lam Nguyen, and Maria Potop-Butucaru. Cross hyperledger fabric transactions. In Proceedings of the 3rd Workshop on Cryptocurrencies and Blockchains for Distributed Systems, pages 35–40, 2020.
[29] Hong Su. Cross-chain interaction model in a fully verified way. arXiv preprint arXiv:2106.05463, 2021.
[30] Paolo Tasca and Claudio J Tessone. Taxonomy of blockchain technologies. principles of identification and classification. arXiv preprint arXiv:1708.04872, 2017.
[31] Babu Pillai, Kamanashis Biswas, and Vallipuram Muthukkumarasamy. Blockchain interoperable digital objects. In International Conference on Blockchain, pages 80–94. Springer, 2019.
[32] Non-fungible tokens. https://ethereum.org/en/nft/. Accessed: 13/01/2022.
[33] Peter Robinson and Raghavendra Ramesh. General purpose atomic crosschain transactions. In 2021 3rd Conference on Blockchain Research Applications for Innovative Networks and Services (BRAINS), pages 61–68, 2021.
[34] Ittay Eyal and Emin Gün Sirer. Majority is not enough: Bitcoin mining is vulnerable. In International conference on financial cryptography and data security, pages 436–454. Springer, 2014.
[35] Michael Sober, Giulia Scaffino, Christof Spanring, and Stefan Schulte. A voting-based blockchain interoperability oracle. arXiv preprint arXiv:2111.10091, 2021.
[36] Philipp Frauenthaler, Marten Sigwart, Christof Spanring, and Stefan Schulte. Leveraging blockchain relays for cross-chain token transfers. Gas, 300:600–000, 2020.
[37] R Belchior, A Vasconcelos, M Correia, and T Hardjono. Hermes: Fault-tolerant middleware for blockchain interoperability, techrxiv 14120291/1 (mar 2021). arxiv: 1, doi: 10.36227/techrxiv. 14120291. V1. URL/articles/preprint/HERMES_Fault-Tolerant_Middleware_for_Blockchain_Interoperability/14120291/1.
[38] Thomas Hardjono. Blockchain gateways, bridges and delegated hash-locks. arXiv preprint arXiv:2102.03933, 2021.
[39] Thomas Hardjono, Martin Hargreaves, and Ned Smith. An interoperability architecture for blockchain gateways, 2020.
[40] Thomas Hardjono, Alexander Lipton, and Alex Pentland. Toward an interoperability architecture for blockchain autonomous systems. IEEE Transactions on Engineering Management, 67(4):1298–1309, 2019.
[41] Oleksii Konashevych. Cross-blockchain protocol for public registries. International Journal of Web Information Systems, 2020.
[42] Cosimo Sguanci, Roberto Spatafora, and Andrea Mario Vergani. Layer 2 blockchain scaling: a survey. arXiv preprint arXiv:2107.10881, 2021.
[43] Kostis Karantias, Aggelos Kiayias, and Dionysis Zindros. Proof-of-burn. In International Conference on Financial Cryptography and Data Security, 2019.
[44] Babu Pillai, Kamanashis Biswas, Zhé Hóu, and Vallipuram Muthukkumarasamy. The burn-to-claim cross-blockchain asset transfer protocol. In 2020 25th International Conference on Engineering of Complex Computer Systems (ICECCS), pages 119–124. IEEE, 2020.
[45] Babu Pillai, Kamanashis Biswas, Zhé Hóu, and Vallipuram Muthukkumarasamy. Burn-to-claim: An asset transfer protocol for blockchain interoperability. Computer Networks, 200:108495, 2021.
[46] Thomas Eizinger, Philipp Hoenisch, and Lucas Soriano del Pino. Open problems in cross-chain protocols. arXiv preprint arXiv:2101.12412, 2021.
[47] Maurice Herlihy. Atomic cross-chain swaps. In Proceedings of the 2018 ACM symposium on principles of distributed computing, pages 245–254, 2018.
[48] Runchao Han, Haoyu Lin, and Jiangshan Yu. On the optionality and fairness of atomic swaps. In Proceedings of the 1st ACM Conference on Advances in Financial Technologies, pages 62–75, 2019.
[49] Market makers. https://www.gemini.com/cryptopedia/amm-what-are-automated-market-makers/. Accessed: 13/01/2022.
[50] Uniswap protocol. https://uniswap.org/. Accessed: 13/01/2022.
[51] Giulio Caldarelli. Wrapping trust for interoperability. a study of wrapped tokens. arXiv preprint arXiv:2109.06847, 2021.
[52] Tier Nolan. Alt chains and atomic transfers. https://bitcointalk.org/index.php?topic=193281.0, accessed date: 10 June 2019.
[53] Adam Back, Matt Corallo, Luke Dashjr, Mark Friedenbach, Gregory Maxwell, Andrew Miller, Andrew Poelstra, Jorge Timón, and Pieter Wuille. Enabling blockchain innovations with pegged sidechains. http://www.opensciencereview.com/papers/123/enablingblockchain-innovations-with-pegged-sidechains, 72, 2014.
[54] Joseph Poon and Thaddeus Dryja. The bitcoin lightning network: Scalable off-chain instant payments, 2016.
[55] Christian Decker and Roger Wattenhofer. A fast and scalable payment network with bitcoin duplex micropayment channels. In Symposium on Self-Stabilizing Systems, pages 3–18. Springer, 2015.
[56] Paul Sztorc. Drivechain - the simple two way peg. http://www.truthcoin.info/blog/drivechain/ accessed date: 10 June 2019.
[57] Kazi Masudul Alam, JM Ashfiqur Rahman, Anisha Tasnim, and Aysha Akther. A blockchain-based land title management system for bangladesh. Journal of King Saud University-Computer and Information Sciences, 2020.
[58] Ghassan Karame, Elli Androulaki, and Srdjan Capkun. Two bitcoins at the price of one? double-spending attacks on fast payments in bitcoin. IACR Cryptology ePrint Archive, 2012(248), 2012.
[59] Yonatan Sompolinsky and Aviv Zohar. Secure high-rate transaction processing in bitcoin. In International Conference on Financial Cryptography and Data Security, pages 507–527. Springer, 2015.
[60] Bingrong Dai, Shengming Jiang, Menglu Zhu, Ming Lu, Dunwei Li, and Chao Li. Research and implementation of cross-chain transaction model based on improved hash-locking. In International Conference on Blockchain and Trustworthy Systems, pages 218–230. Springer, 2020.
[61] Abdeljalil Beniiche. A study of blockchain oracles. arXiv preprint arXiv:2004.07140, 2020.
[62] Jonathan Heiss, Jacob Eberhardt, and Stefan Tai. From oracles to trustworthy data on-chaining systems. In 2019 IEEE International Conference on Blockchain (Blockchain), pages 496–503. IEEE, 2019.
[63] Steve Ellis, Ari Juels, and Sergey Nazarov. Chainlink: A decentralized oracle network. Retrieved March, 11:2018, 2017.
[64] Philipp Frauenthaler, Michael Borkowski, and Stefan Schulte. A framework for blockchain interoperability and runtime selection. arXiv preprint arXiv:1905.07014, 2019.
[65] Rongjian Lan, Ganesha Upadhyaya, Stephen Tse, and Mahdi Zamani. Horizon: A gas-efficient, trustless bridge for cross-chain transactions. arXiv preprint arXiv:2101.06000, 2021.
[66] Mohammad Madine, Khaled Salah, Raja Jayaraman, Yousof Al-Hammadi, Junaid Arshad, and Ibrar Yaqoob. appxchain: Application-level interoperability for blockchain networks. IEEE Access, 9:87777–87791, 2021.
[67] Frederik Armknecht, Ghassan O Karame, Avikarsha Mandal, Franck Youssef, and Erik Zenner. Ripple: Overview and outlook. In International Conference on Trust and Trustworthy Computing, pages 163–180. Springer, 2015.
[68] Amritraj Singh, Kelly Click, Reza M Parizi, Qi Zhang, Ali Dehghantanha, and Kim-Kwang Raymond Choo. Sidechain technologies in blockchain networks: An examination and state-of-the-art review. Journal of Network and Computer Applications, 149:102471, 2020.