DOULP: Deep Omnichain Unified Liquidity Protocol

July 2023

Bisonai Labs

Abstract

The world of blockchain has transformed from a single-chain approach to an omnichain, where a set of blockchains can interact together. It is vital that any omnichain participant can freely transfer assets between any pair of blockchains within the omnichain in a secure and reliable manner. At the time of this writing, there are more than a dozen bridges that transfer approximately $5 billion worth of assets monthly on average. However, the current evolution of bridges poses significant risks to Decentralized Finance (DeFi), introduces fragmentation into the market, or creates limitations in one way or another.

In this paper, we propose DOULP: Deep Omnichain Unified Liquidity Protocol, which aims to create a more efficient and frictionless omnichain market. Unlike other protocols, it does not rely on wrapped or intermediate tokens that are susceptible to liquidity fragmentation and hacks. With DOULP, liquidity is equally accessible to all connected chains without any restrictions. Moreover, the protocol allows for seamless expansion to new chains without the need for additional liquidity. The protocol was purposefully designed with the users and liquidity providers in mind, prioritizing their needs and convenience, but not sacrificing the security of the protocol. To ensure secure and reliable message transfer, DOULP incorporates the OMP: Omnichain Messaging Protocol, which facilitates seamless omnichain communication.

1. Introduction

In the past several years the rise of new blockchains was staggering. Blockchain architects invented better ways to provide security, proposed solutions that led to faster and cheaper transactions, and whole ecosystem started to slowly embrace less tech-savvy users. However, apart from a few exceptions, the majority of blockchains tend to overlook the broader blockchain ecosystem. This behavior is primarily influenced by the deterministic nature of the blockchain and its native isolation.

Due to the increasing number of well established blockchains, many users did not want to limit the use of their assets to a single chain, and they sought for a way to transfer their assets from one blockchain to another. Together with the growth in number of blockchains, various cross-chain asset bridges started to appear that seemingly offered a solution to the cross-chain asset transfer demand.

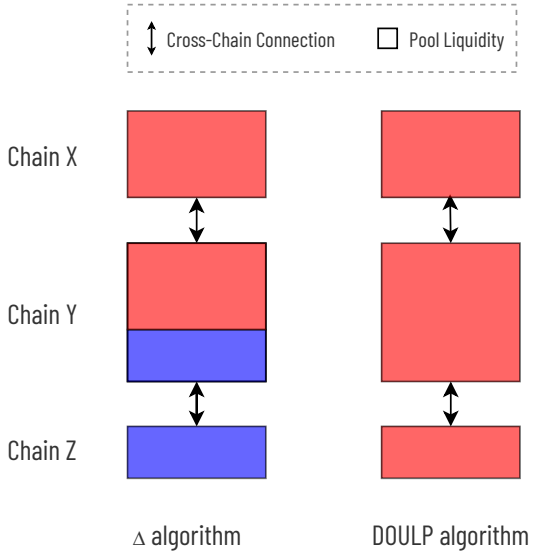

Figure 1. Comparison of pool liquidity between the Δ algorithm and proposed DOULP algorithm (see section 5): DOULP introduces a genuine unified liquidity property (section 3.2) that allows for the sharing of liquidity in the destination pool for all incoming asset transfers from all connected source pools.

At present day, there are more than a dozen of bridges, each trying to solve a different cross-chain bridge subproblem. However, as this field is still evolving, every cross-chain asset bridge comes with its own set of issues.

Before embarking on the design of a new cross-chain asset bridge, we sought to answer the question of what qualities an excellent cross-chain bridge should possess. Our findings yielded the following criteria: security, speed, affordability, scalability, and user-friendliness. We cross-examined available solutions (details in section 2) and dis-

tilled six key deficiencies in cross-chain asset bridges, including (1) the use of wrapped tokens, (2) intermediate tokens, (3) sparse cross-chain connections, (4) limited market and capital efficiency, (5) poor user-friendliness, and (6) insufficient security measures.

In this work we propose DOULP: Deep Omnichain Unified Liquidity Protocol that addresses the first five key deficiencies of cross-chain asset bridges. DOULP, being an omnichain asset bridge, does not utilize wrapped or intermediate tokens. It allows for a denser cross-chain connectivity without need for increasing the amount of liquidity. This is achieved using *genuine unified liquidity* that is shared with every connected source chain without any limitations. Additionally, it enables a greater liquidity utilization by removing the lower bound of pool liquidity, making them essentially deeper. DOULP does not pose restrictions on what market participants are allowed to do and when. It is perfectly possible to transfer all liquidity from one pool to the another across blockchains without user penalization. Liquidity providers (LP) benefit from this approach as well, because they can theoretically collect more fees in the process.

In the second part of this paper, we introduce OMP: Omnichain Messaging Protocol, which aims to address the final key deficiency: insufficient security measures. OMP utilizes a two-layer message validation system to enhance security measures for delivering cross-chain messages, thereby mitigating potential attacks that attempt to breach the security of the communication protocol.

The rest of the paper is composed as follows: Technical details of cross-chain asset bridge competitors are described in section 2. Section 3 introduces DOULP: Deep Omnichain Unified Liquidity Protocol and explains its advantages over previous approaches. Section 4 provides several detailed examples illustrating the process of asset transfer. The communication protocol, OMP: Omnichain Messaging Protocol, which powers DOULP-based bridges, is outlined in section 5. Finally, in section 6, we examine previous bridge attacks and evaluate the impact of the attack vectors used on DOULP-based bridges.

2. Background

This section explains the concept of cross-chain asset bridges, their categorization, and highlights the limitations of current bridge implementations based on the six key deficiencies identified in cross-chain asset bridges, as listed in section 1.

A cross-chain asset bridge is defined as a technology that connects different blockchain networks, allowing the transfer of digital assets between them. It ensures compatibility and interoperability, enabling assets to move seamlessly from one blockchain to another. By bridging the gap between disparate blockchains, cross-chain asset bridges enhance the overall liquidity and accessibility of digital assets across various blockchain ecosystems. Because of the multitude of diverse blockchains and the isolation of blockchain networks from the rest of the ecosystem, various bridges optimize for different sets of outcomes.

2.1. Trusted and Trustless Bridges

Regardless of the internal objectives of a bridge, there is one characteristic that can be used to compare all bridges: the level of trust associated with each of them. Blockchain bridges can be assessed on a scale ranging from the trusted to the trustless level.

Trusted bridges refer to blockchain bridges that rely on a centralized authority or intermediary to facilitate and validate transactions between different blockchain networks, introducing a level of trust in the bridge operation.

Trustless bridges are blockchain bridges that operate in a decentralized manner, utilizing smart contracts or cryptographic protocols to enable direct and secure asset transfers between different blockchain networks without the need for a centralized authority.

The design of the DOULP-based cross-chain asset bridge (section 3) that is proposed in this paper is considered a trustless bridge. This is achieved through decentralized collaboration between the Oracle and Relayer, as well as the on-chain proof and multisig verification provided by the Omnichain Messaging Protocol, which is discussed in section 5.

2.2. Cross-chain Asset Bridges

In this subsection, we delve into the largest groups of cross-chain asset bridges and explore different architectures, protocols, and mechanisms that enable seamless asset transfers.

Early type of asset bridges, some of which are some of the most popular ones [14, 24] until today, are based on *wrapped token*. Wrapped tokens are created through mint/burn bridge, allow for 100 % liquidity utilization and do not require LPs to function. Their properties are attractive, however, they come with a security risk and cause market fragmentation. Every wrapped token is at imminent risk of losing its value when bridge infrastructure is hacked, or when the source chain is simply disconnected. Wrapped tokens make market more fragmented, and therefore less efficient, because the same token transferred through wrapped token bridges results in different type of token on destination chain.

Other types of asset bridges [25] use an *intermediate token* that enable a cross-chain asset exchange between arbitrary types of tokens. Such bridges require two extra steps in the process of exchange through pools backed by the intermediate token. In addition to the risks associated with the value fluctuations of intermediate tokens, such bridge solutions are more complex and can result in unfulfilled trans-

fers on the destination chain due to significant slippage.

The large portion of the cross-chain asset bridge market has recently been captured by the native L1↔L2 bridges [2, 3, 9, 12, 13, 15, 23] that connect Ethereum with their scaling solutions. The highest priority for such bridges is a security, but their focus on increasing number of supported tokens, building a denser connection network and user-friendliness lacks behind.

There are other cross-chain bridge solutions available in the market, but they have not gained significant market share according to DeFiLlama bridge leaderboard [5], and do not provide substantial advancements compared to previously described cross-chain asset bridges. Therefore, we have chosen not to delve into them in this discussion.

The next section focuses on the Stargate bridge, which has spearheaded the emergence of a new type of blockchain bridges that exchange a native assets across chains.

2.3. Stargate

Stargate [18] has emerged as a colossal asset bridge with the highest asset volume, surpassing other bridges by a significant margin. It employs the omnichain communication protocol LayerZero [27] and the ΔBridge [26] for facilitating the exchange of *native tokens* across interconnected chains. Native tokens are issued by the same entity on both ends of the bridge connection. Although they may not be strictly fungible, they can be considered as such, and importantly, they do not encounter the same problems as wrapped tokens. Notable examples of native tokens include stablecoins such as USDC and USDT.

The rules utilized by the ΔBridge to transfer native tokens are determined by the Δ algorithm. The objective of this algorithm is to facilitate cross-chain asset transfers that fulfill three specific properties: *instant guaranteed finality*, *unified liquidity*, and *native asset transactions*. Ryan et al. argue that instant guaranteed finality is an essential property for any bridge, but we contend that it is merely a desirable feature that proves useful in rare circumstances.

To achieve instant guaranteed finality, the Δ algorithm necessitates the presence of soft-partitioned pools. Each soft-partitioned pool must have its own dedicated connection and cannot share liquidity with other pools that hold the same native token (see Figure 3). This design directly contradicts the purported achievement of unified liquidity claimed by the Δ algorithm. In reality, unified liquidity can only be assumed from the perspective of liquidity providers who can earn rewards from the pool they deposited into, regardless of which chain the assets originate from. However, users who intend to transfer native assets face limitations imposed by the size of soft-partitioned pools, even if the destination pool contains an ample amount of tokens.

Stargate facilitates the transfer of assets across a significant number of blockchains. However, due to the Δ algorithm, scaling becomes costly and capital inefficient. Whenever a new chain is connected to Stargate, it necessitates the provision of additional liquidity to meet the required bandwidth. Alternatively, funds from neighboring soft-partitioned pools need to be utilized, which effectively reduces the asset transfer limit for the connection to the pool from which the funds were used.

Another important objective of the Δ algorithm is to maintain a token balance equilibrium among the connected soft-partitioned pools. This equilibrium is achieved by applying an additional fee that increases nonlinearly based on the amount transferred, the current balance of the soft-partitioned pool, and the ideal balance of the pool. We strongly advocate against penalizing users solely for utilizing the cross-chain asset bridges for their intended purpose.

Finally, the process of redeeming deposited funds from the Stargate pool is convoluted. It only permits direct redemption of a limited amount. For larger amounts, one or two cross-chain messages must be exchanged to complete the redemption process.

The limitations of Stargate described above reflect architectural flaws that result in low capital and market efficiency. These limitations are effectively addressed by DOULP.

3. Deep Omnichain Unified Liquidity Protocol

In this section, we introduce DOULP: Deep Omnichain Unified Liquidity Protocol which addresses shortcomings of ΔBridge (see Figure 1). DOULP enhances market efficiency within the omnichain ecosystem by facilitating less restrictive cross-chain transfers, and enables liquidity providers to benefit from increased capital efficiency.

The design of DOULP was guided by the first five key deficiencies (outlined in section 1) that we identified during our research of cross-chain asset bridges. DOULP does not directly aim to enhance the security, however, due to its simplicity, it helps mitigate a potential set of attack vectors. Security measures are addressed through the utilization of OMP: Omnichain Messaging Protocol (see section 5). As discussed in section 2, the utilization of wrapped and intermediate tokens presents notable security risks and inefficiencies. Consequently, we have made the decision to avoid their use in our protocol.

In the following subsections, we will describe how DOULP works and detail the benefits derived from its design, directly addressing the issues prevalent in cross-chain asset bridges, including sparse cross-chain connections, limited market and capital efficiencies, and poor user-friendliness.

3.1. Protocol Design

DOULP is a cross-chain asset bridge that enables transfer of native tokens. The main component of the proto-

col is a single-sided asset pool, which is connected to another single-sided pools through a unidirectional connection. Pool can be designated as either a source or a destination based on the direction of asset transfer. In order to allow transfer between two pools in both directions, two unidirectional connections must be created. Once the connection is established, it is possible to transfer up to the available liquidity amount on the destination chain, if desired.

Destination pools must have a sufficient balance, which can be added by either liquidity providers or users who have transferred native assets in the reverse direction. Upon depositing funds, liquidity providers receive redeemable LP tokens.

The process of transferring native tokens from the source chain to the destination chain is illustrated in Figure 2. It is divided into two steps: depositing tokens into the pool on the source chain (lines 2 - 3) and receiving an equal amount of tokens on the destination chain (lines 6 - 14).

When tokens are deposited into the source pool p_{src}, the pool balance b_{src} is increased by the deposited amount t (line 2), and a message containing information about the destination address and the deposited amount is sent to the destination pool p_{dst} (line 3).

On the destination chain, a message is received containing the transferred asset amount t and the destination address $addr_{dst}$, which is then redirected to the destination pool p_{dst} (line 6). If the amount of the transferred asset is less than or equal to the balance b_{dst} of the destination pool, t tokens from the destination pool p_{dst} are transferred to the destination address $addr_{dst}$, and the balance of the destination pool is adjusted accordingly (lines 7 - 9). If the destination pool balance b_{dst} is insufficient (lines 10 - 13), only the available amount of tokens b_{dst} is transferred to the destination address (line 11). The remaining amount is converted into LP tokens, which can be redeemed for an equivalent amount of tokens from either the destination pool p_{dst} or any connected source pool p_{src}. LP tokens are transferred to the destination address $addr_{dst}$ (line 12).

The protocol design has several notable implications, including *genuine unified liquidity*, *dense cross-chain connectivity*, and *probabilistic finality*, which position DOULP as a superior solution for cross-chain asset bridging. In the following subsections, we will discuss these design implications and their significance in the context of bridging assets across different chains.

3.2. Genuine Unified Liquidity

The protocol design results in the genuine unified liquidity property, allowing for the sharing of liquidity with any connected pools. This feature benefits both users and liquidity providers. Users are not restricted by the size of pool partitions, while liquidity providers receive rewards regardless of the source pool from which the asset transfer originated. Unified liquidity in DOULP-based bridges fundamentally leads to enhanced capital efficiency.

```
1  # source pool p_src
2  b_src += t
3  send (addr_dst, t) to p_dst
4
5  # destination pool p_dst
6  receive (addr_dst, t) from p_src
7  if t ≤ b_dst then
8    transferToken t to addr_dst
9    b_dst -= t
10 else
11   transferToken b_dst to addr_dst
12   mintLpToken t-b_dst to addr_dst
13   b_dst = 0
14 end if
```

Figure 2. On-chain part of DOULP algorithm

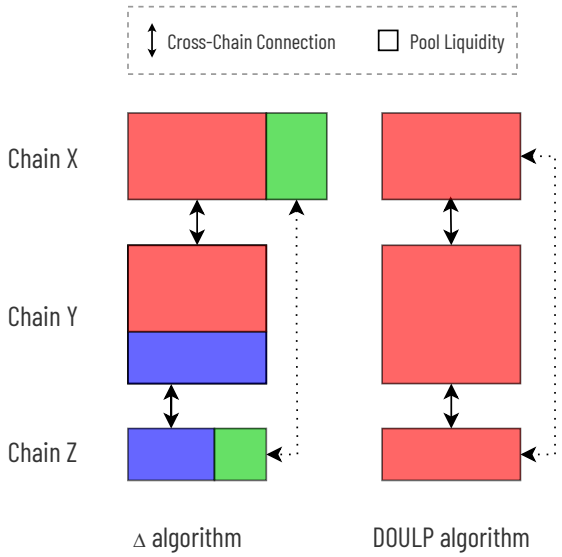

Figure 3. Comparison between the Δ algorithm and the proposed DOULP algorithm when a new connection is created between existing pools on chains that have not been connected yet. The dotted line in the figure represents a new connection that is being created between the pool on chain X and chain Z. The initial state of the pools before the creation of a new connection can be seen in Figure 1. The Δ algorithm either requires adding new liquidity (as exemplified in the figure on chain X with green liquidity) or taking liquidity from other existing pools (as displayed with green liquidity on chain Z). The DOULP algorithm does not require adding or taking over other pools' liquidity, and the connection between existing pools can be created by simply creating a new two unidirectional connections.

3.3. Dense Cross-chain Connectivity

Cross-chain asset bridges that provide dense cross-chain connectivity are regarded as more user-friendly since they eliminate the need for users to conduct research each time they transfer their assets to another chain. Scaling a bridge solution to new chains is a challenging task that directly impacts the extent to which dense cross-chain connections can be established.

DOULP's distinctive design, leveraging genuine unified liquidity (section 3.2), enables significant potential for scaling. When establishing a new connection between existing pools, we can simply create two opposite unidirectional connections between the pools, without the need to add any additional liquidity to either pool (see Figure 3). If we want to create a pool on a new chain and connect it to an existing pool on another chain, additional liquidity only needs to be provided to the new pool. This demonstrates that the increased capital efficiency resulting from genuine unified liquidity enables the realization of dense cross-chain connectivity.

3.4. Optimistic Finality

Cross-chain asset transfers utilizing the DOULP algorithm operate based on an optimistic finality approach. The on-chain source pool does not verify the availability of sufficient liquidity on the destination chain, which can result in the transfer partially being made in LP tokens instead. Nonetheless, thanks to genuine unified liquidity, the probability of encountering insufficient liquidity is relatively low.

The possibility of insufficient liquidity on the destination chain can be further mitigated by predicting the amount of liquidity on the destination chain at the time of token release on the destination chain.

3.5. Discussion

The details of the protocol design, as described in section 3.1 and demonstrated in Figure 2, are sufficient for on-chain implementation. In this section, we will delve into rare edge cases of protocol behavior and offer recommendations on how to mitigate them even further.

DOULP operates based on the concept of optimistic finality (section 3.4), assuming the presence of sufficient liquidity on the destination chain. The likelihood of encountering insufficient liquidity on the destination chains is low due to the genuine unified liquidity property of the DOULP algorithm. However, it is beneficial for the entire ecosystem to have sufficient liquidity that enables uninterrupted cross-chain asset transfers. We recommend setting a constant fee, proportional to the amount transferred, which we refer to as the *incentive fee*. This fee is charged for every cross-chain asset transfer and is deducted on the destination chain, then added to the incentive pool. When users transfer assets in the opposite direction or LPs deposit more tokens to the pool, they receive an incentive fee proportional to the transferred or deposited amount, respectively. We discourage the use of dynamic fees, as seen in some other cross-chain asset bridges (e.g., Stargate [18]). Such configurations penalize users for transferring amounts of assets beyond the permissioned limit or for moving assets in an undesirable direction.

Because DOULP is an open and unrestricted asset transfer protocol, it is technically possible to drain a destination pool without incurring any penalties, except for transaction and incentive fees. When this scenario occurs and the bridge is drained or close to being drained, there is a higher likelihood that some of the users' asset transfers will result in partial or full LP token payout on the destination chain. It is reasonable to assume that such situations will not be well-received, as users aim to transfer native assets from the source chain to the destination chain and receive the full expected amount of tokens on the destination side, without any LP tokens. The solution to this problem is straightforward. The bridge operator has access to information about all bridges, pools, balances, network congestion, pool drain speed, and other relevant data. This information can be utilized to train a predictive model that estimates the probability of a transfer not being executed 100 % in native assets. Users can utilize this information when interacting with the bridge frontend to make an informed decision on whether to proceed with the transfer or not. Additionally, the asset transfer probability API can be made accessible to third-party services that wish to integrate with the existing bridge.

We want to stress that under no circumstances does the user lose money when the pool is drained or when the transfer is not performed using native assets only. Importantly, such cases are rare and become even more infrequent with increased pool liquidity.

4. DOULP Examples

DOULP is very simple but powerful protocol for cross-chain asset transfers. In this section we will go step by step through three examples (Figure 4) of transferring assets between three different chains (X, Y and Z). To maintain simplicity in this example, we have chosen to assume zero protocol and gas fees. As a result, the value of the transferred amount remains unchanged.

The initial state of pools on all chains is the same. Each chain has liquidity providers who deposited 100 tokens, resulting in the pool issuing 100 LP tokens, and the balance of each pool being 100 tokens.

In the first cross-chain transfer, the user intends to send 40 tokens from chain X to chain Y. The user deposits 40 tokens into the pool on chain X, resulting in an increase in the pool's balance from 100 to 140 tokens. On chain Y, 40 tokens are released from the pool to the user, and the balance is updated to 60 tokens.

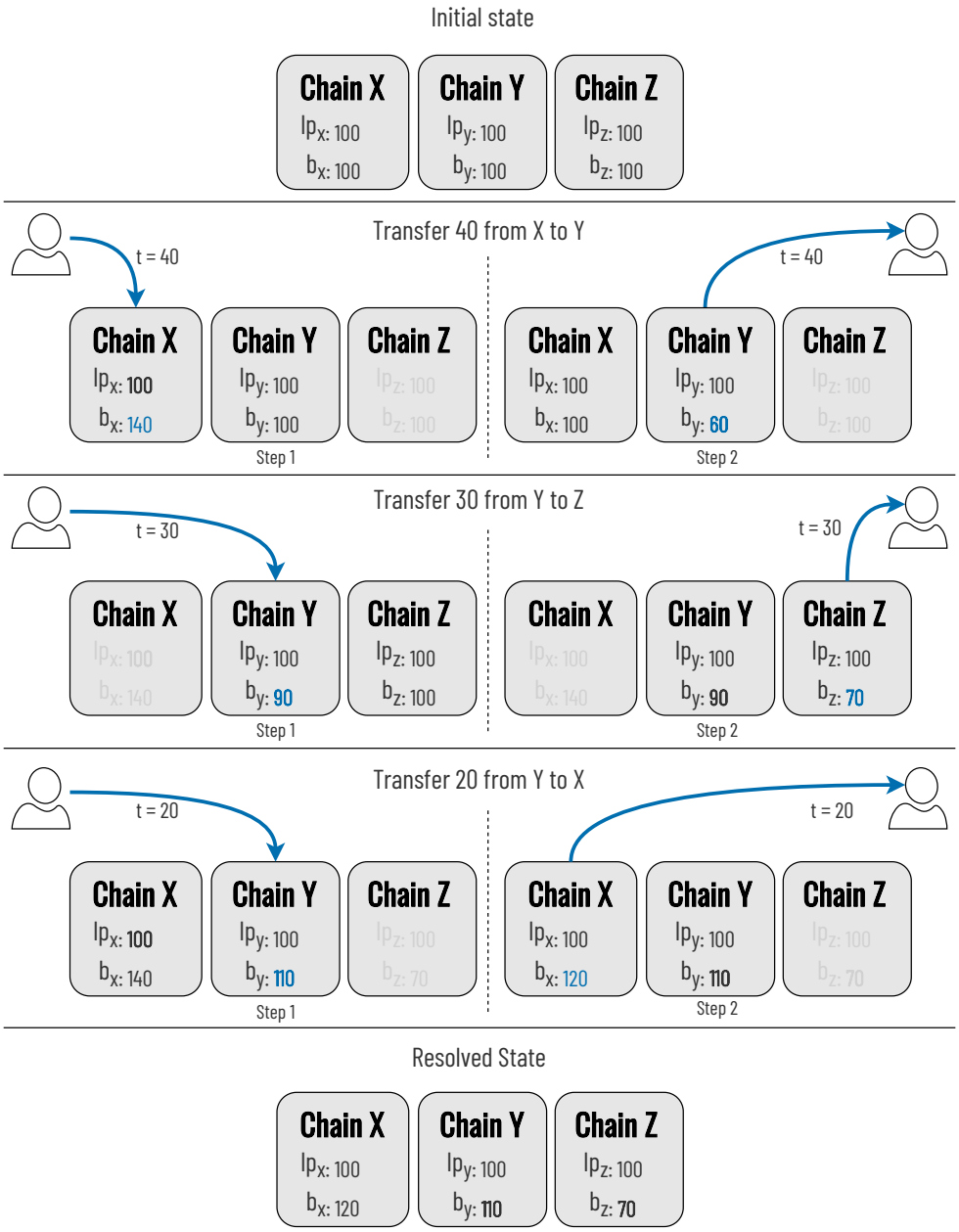

Figure 4. Examples of three consecutive transfers between pools on different chains. A detailed description of every step for each asset transfer is described in section 4.

In the second example, the user deposits 30 tokens into the pool on chain Y, causing the balance value to increase from 60 to 90 tokens. On chain Z, the user receives 30 tokens from the pool, resulting in a decrease in the pool's balance from 100 to 70 tokens.

The last example illustrates a user who wishes to transfer 20 tokens from chain Y to chain X. The user begins by depositing 20 tokens into the pool on chain Y, resulting in an increase in the pool's balance from 90 to 110 tokens. On chain X, the pool's balance is decreased from 140 to 120 tokens, and 20 tokens are transferred to the user.

None of the transfers affected the values of LP tokens; however, the pool balances were updated in both the source and destination pools. The final balances of the pools on chain X, Y, and Z are 120, 110, and 70, respectively.

5. Omnichain Messaging Protocol

Every cross-chain application requires a secure communication layer between blockchains to ensure safe and secure message transfers. The security of cross-chain asset bridges is particularly crucial, as their pools hold large amounts of tokens that attract hackers. This is evidenced in section 6, where we highlight the prominence of cross-chain asset bridge attacks within the DeFi ecosystem.

In this section, we introduce the OMP: Omnichain Messaging Protocol (Figure 5), which enables highly secure transmission of messages between blockchains and directly addresses the last key deficiency of cross-chain asset bridges: insufficient security measures.

5.1. Components

OMP is composed of on-chain and off-chain components (OMP Endpoint, OMP Oracle, and OMP Relayer) that collaborate in a decentralized and trustless manner to securely deliver messages from applications on the source chain to applications on the destination chain.

The **OMP Endpoint** is a set of on-chain smart contracts deployed on both the source and destination chains, enabling the sending and receiving of messages, respectively. The Endpoint accepts message transfer requests, validates them, creates packets, and emits them to initiate cross-chain message transfers. On the receiving end of the communication channel, the endpoint receives the packets, validates them, and executes application functions that are encoded within them.

The **OMP Oracle** is an off-chain component consisting of a set of independent trustworthy parties that collaborate together to generate proof from the emitted packet, and which must be confirmed by more than 2/3 of the oracles. Once the proof is confirmed by the quorum, the oracles allow external parties to access the original packet enriched with the *proof* and *multisig* of the proof hash.

The **OMP Relayer** is an off-chain component that can be controlled by any entity. Its purpose is to relay data provided by the Oracle to the Endpoint on the destination chain. Prior to submitting the data to the destination endpoint, the Relayer appends a *proof component* to the packet. This proof component is utilized during the inbound validation process on the destination chain.

5.2. Protocol Design

The design of OMP draws inspiration from LayerZero [27] and Wormhole's Guardian Network [21]. It uses fewer transactions than LayerZero and combines the security approaches of both messaging protocols.

LayerZero's cryptographic security is based on the concept of ultra-light node, which verifies the correctness of receipt proofs within a single block without requiring the context of previous blocks. It is an efficient and relatively inexpensive way to prove that a certain event has occurred. However, as described in section 6, four out of eleven bridge attacks were caused by cryptographic vulnerabilities at the code level, which could potentially happen to ultra-light node as well. To mitigate the possibility of such attacks, we have decided to include an additional layer of security, inspired by Wormhole's Guardian Network. Proofs generated from the header of a block or transaction will be additionally signed by OMP Oracles, creating a multisig. This multisig will be verified by both the OMP Relayer and the OMP Endpoint, along with the proof.

OMP requires a certain level of modularity to accommodate the variety of different chains and the specific processes involved in proof generation and verification on each chain. This modularity is essential for both the on-chain (OMP Endpoint) and off-chain (OMP Oracle) validation libraries.

Figure 5 illustrates the steps of sending and delivering a single message from application on chain X to application on chain Y. For each described step, there is a circled number in the figure to assist in understanding the protocol's process.

Step 1: The cross-chain message communication scenario begins with a user submitting a transaction to an on-chain application that is integrated with the OMP Endpoint.

Step 2: The transaction initiated in step 1 generates an internal transaction within the OMP Endpoint, which constructs a *header* containing the destination application and destination chain information.

Step 3: The information from the header is validated against the registered set of destination chains and destination applications.

Step 4: Once the header validation is completed, the full packet (*header*, *payload*) is composed and emitted as an event from the OMP Endpoint.

Step 5: A set of trustworthy permissioned OMP Oracles listens to the event emitted in step 4 and waits for the block or transaction that includes the event to be confirmed. Once the event is immutably stored on the chain, the OMP Oracles generate a proof (e.g., receipt proof for EVM chains) and share it with each other. To validate the proof, at least 2/3 of the OMP Oracles must generate the same proof and sign it with their private keys. Both the *proof* and *multisig* are attached to the original message (*header*, *payload*) received through the event. The OMP Relayer can access the newly generated messages in the next step.

Steps 6-7: The OMP Relayer retrieves the message (*header*, *payload*, *proof*, *multisig*) from the OMP Oracles and requests a *proof component* (e.g., receipt root for EVM chains) from the source chain X. The OMP Relayer utilizes the proof component to verify the proof generated by the OMP Oracles. Once the correctness of the proof is con-

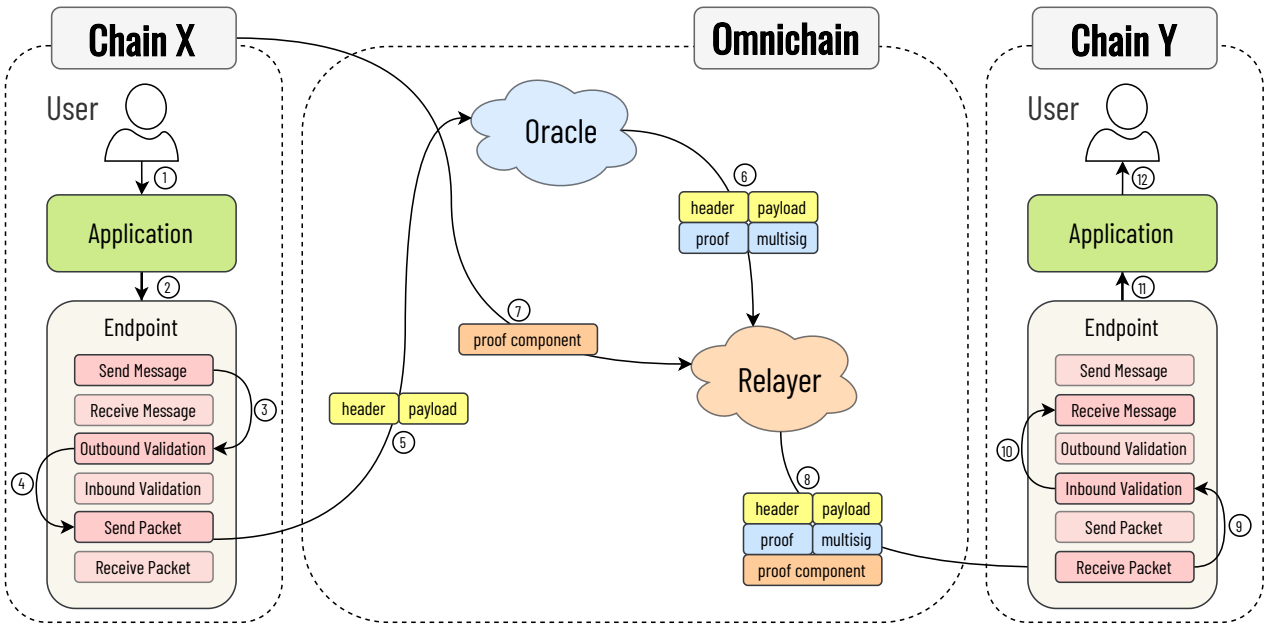

Figure 5. Communication flow between an application on chain X and an application on chain Y through OMP: Omnichain Messaging Protocol. For more details, see section 5.

firmed, the proof component is attached to the message from the OMP Oracles.

Step 8: The OMP Relayer is responsible for off-chain proof verification (step 7) and for submitting the message (*header*, *payload*, *proof*, *multisig*, *proof component*) to the destination OMP Endpoint on chain Y. The OMP Relayer can be any entity that aims to finalize the transaction by submitting it to the destination chain or a trusted permissioned entity that mitigates the risk of submitting a malicious message generated by a corrupted set of OMP Oracles.

Step 9: The transaction submitted by the OMP Relayer in step 8 triggers a series of internal transactions, starting with the inbound message validation. In this step, we verify that a minimum of 2/3 of active oracles have submitted the same proof and validate the correctness of the *proof* using the *proof component*.

Step 10-12: Once the proof is confirmed to be valid, the OMP Endpoint initiates an external call, as defined in the message *payload*, to the destination application specified in the *header*. This finalizes the cross-chain message transfer from chain X to chain Y.

6. Security

Cross-chain asset bridges are frequent targets of hackers due to the large amounts of tokens stored in its pools. According to Rekt [16], four out of top five largest attacks (table 1) in history of DeFi targeted cross-chain asset bridges.

The utilization of wrapped tokens in bridges amplifies the impact of attacks by enabling the creation of an unlimited quantity of tokens on the destination side, which could subsequently be transferred back to the source chain to acquire native tokens.

Besides the vulnerabilities inherent in wrapped token bridges, there have been numerous successful attacks on various cross-chain asset bridges. Ronin Network [17], Poly Network [11], and Harmony Bridge [6] were all compromised by attackers who illicitly obtained access to the private keys used for multisig protection in their respective cross-chain protocols. Cryptographic vulnerabilities caused by code-level bugs or improper use of cryptography libraries affected BNB Bridge [4], Wormhole [22], and Anyswap [1]. The first Poly Network [10] attack exploited a peculiar method of generating a function selector, which enabled making arbitrary external calls on behalf of the contract. Nomad Bridge [8] exposed itself to an attack due to an unfortunate incomplete update of the bridge, which allowed the bypassing of validation for arbitrary messages on the destination chain. Interestingly, THORChain [19, 20], which has been hacked twice already, does not fit into any of the previously mentioned attack vectors. In both cases, the attacker exploited a bug in the code. Multichain [7], previously known as Anyswap, had already been hacked [1] before the writing of this paper. Another attack occurred during the writing process, and a complete post-mortem report has not yet been released.

It is important to stress that many of the above-mentioned bridges have not undergone official audits or their audit reports are not available. Auditing DeFi proto-

Rank	Name	Cause	$M
1	Ronin Network [17]	phishing, 5/9 multisig	624
2	Poly Network [10]	contract vulnerability	611
3	BNB Bridge [4]	cryptographic vulnerability	586
5	Wormhole [22]	cryptographic vulnerability	326
8	Nomad Bridge [8]	incomplete bridge update	190
14	Multichain [7]	under investigation	126.3
18	Harmony Bridge [6]	2/5 multisig	100
75	THORChain [19]	lack of proper multi-event handling	10
78	Anyswap [1]	cryptographic vulnerability	8
94	THORChain [20]	off-chain code logic	5
97	Poly Network [11]	3/4 multisig	4.4

Table 1. Four of top five Rekt's leaderboard [16] entries are cross-chain asset bridge attacks.

cols is an essential step in the development phase and should never be considered an afterthought.

7. Conclusion

In this paper, we present two innovative cross-chain protocols; DOULP: Deep Omnichain Unified Liquidity Protocol and OMP: Omnichain Messaging Protocol. DOULP is a cross-chain asset protocol designed to provide deep unified liquidity across chains, while OMP ensures secure and trustless cross-chain messaging. The design of DOULP and OMP was motivated by the need to address six key deficiencies that we identified in the current state of cross-chain asset bridges: including the use of wrapped tokens, intermediate tokens, sparse cross-chain connections, limited market and capital efficiency, poor user-friendliness, and insufficient security measures.

The first two deficiencies are resolved by entirely excluding the integration of wrapped and intermediate tokens in the protocol design. The next deficiency, sparse cross-chain connectivity, is overcome by the genuine unified liquidity property of DOULP, which enables the reuse of existing pools when establishing new cross-chain connections. The limitation in market and capital efficiency is resolved through a combination of the simplicity of DOULP's algorithm, which allows for unrestricted transfer of assets without penalties, and the genuine unified liquidity that enables the connection between any pools without limiting the resources of the pool. User friendliness is improved by removing limitations on asset transfer direction and amount, as well as enabling the redemption of LP tokens in a single transaction. Insufficient security measures are addressed by the OMP cross-chain messaging protocol, which employs a two-layer security protection approach both off-chain and on-chain.

Bridges constructed using the DOULP and OMP protocols embody all the essential properties necessary for a safe and efficient cross-chain asset bridge. We consider these protocols to be impactful market catalysts and firmly believe in their tremendous potential to enhance the entire cross-chain asset ecosystem.

Disclaimer

This paper is for general information purposes only. It does not constitute investment advice or a recommendation or solicitation to buy or sell any investment and should not be used in the evaluation of the merits of making any investment decision. It should not be relied upon for accounting, legal or tax advice or investment recommendations. The opinions reflected herein are subject to change without being updated.

References

[1] Anyswap - Rekt. https://rekt.news/anyswap-rekt. Accessed: 2023-07-09. 8, 9

[2] Arbitrum Bridge. https://bridge.arbitrum.io. Accessed: 2023-07-05. 3

[3] Avalanche Bridge. https://bridge.avax.network. 3

[4] BNB Bridge - Rekt. https://rekt.news/bnb-bridge-rekt. Accessed: 2023-07-09. 8, 9

[5] DeFiLlama - Bridges. https://defillama.com/bridges. Accessed: 2023-07-09. 3

[6] Harmony Bridge - Rekt. https://rekt.news/harmony-rekt. Accessed: 2023-07-09. 8, 9

[7] Multichain - Rekt. https://rekt.news/multichain-rekt2. Accessed: 2023-07-09. 8, 9

[8] Nomad Bridge - Rekt. https://rekt.news/nomad-rekt. Accessed: 2023-07-09. 8, 9

[9] Optimism Gateway Bridge. https://app.optimism.io/bridge. 3

[10] Poly Network - Rekt. https://rekt.news/polynetwork-rekt. Accessed: 2023-07-09. 8, 9

[11] Poly Network - Rekt. https://rekt.news/poly-network-rekt. Accessed: 2023-07-09. 8, 9

[12] Polygon POS Bridge. https://wallet.polygon.technology/polygon/bridge. Accessed: 2023-07-05. 3

[13] Polygon zkEVM Bridge. https://wallet.polygon.technology/zkEVM-Bridge/bridge. Accessed: 2023-07-05. 3

[14] Portal Bridge. https://www.portalbridge.com. Accessed: 2023-07-05. 2

[15] Rainbow Bridge. https://rainbowbridge.app. 3

[16] Rekt Leaderboard. https://rekt.news/leaderboard. Accessed: 2023-07-04. 8, 9

[17] Ronin Network - Rekt. https://rekt.news/ronin-rekt. Accessed: 2023-07-09. 8, 9

[18] Stargate Bridge. https://stargate.finance. 3, 5

[19] THORChain - Rekt. https://rekt.news/thorchain-rekt2. Accessed: 2023-07-09. 8, 9

[20] THORChain - Rekt. https://rekt.news/thorchain-rekt. Accessed: 2023-07-09. 8, 9

[21] Wormhole - Guardian Network. https://book.wormhole.com/wormhole/5_guardianNetwork.html. Accessed: 2023-07-09. 7

[22] Wormhole - Rekt. https://rekt.news/wormhole-rekt. Accessed: 2023-07-09. 8, 9

[23] zkSync Era Bridge. https://bridge.zksync.io. Accessed: 2023-07-05. 3

[24] Anyswap: Fully Decentralized Cross Chain Swap Protocol. https://github.com/anyswap/Anyswap-Audit/blob/master/whitepaper/Anyswap-whitepaper.pdf, 2020. 2

[25] THORChain: A Decentralized Liquidity Protocol. https://github.com/thorchain/Resources/blob/master/Whitepapers/THORChain-Whitepaper-May2020.pdf, 2020. 2

[26] Ryan Zarick, Bryan Pellegrino, and Caleb Banister. Δ: Solving the Bridging Trilemma. https://www.dropbox.com/s/gf3606jedromp61/Delta-Solving.The.Bridging-Trilemma.pdf, 2021. 3

[27] Ryan Zarick, Bryan Pellegrino, and Caleb Banister. LayerZero: Trustless Omnichain Interoperability Protocol. https://layerzero.network/pdf/LayerZero_Whitepaper_Release.pdf, 2021. 3, 7