A Cross-Chain Scheme for Goods Atomicity in E-commerce Transactions

Yuanzhe Liu[1,a], Yukun Zheng[1], Yahui Guo[1], Yinyan Dou[1], Zhiming Cai[2*]

[1] Faculty of Data Science, City University of Macau, Macau, China
[2] Faculty of Digital Science and Technology, Macau Millennium College, Macau, China

[a]D22091100672@cityu.edu.mo; *zmcai@mmc.edu.mo

Abstract. E-commerce has become a crucial part of global economic activities, and ensuring the atomicity of transactions in E-commerce is vital. So far, existing E-commerce solutions face the following atomicity problems: DigiCash with double-spending, SSL and SET with failing to ensure goods atomicity, and the Netbill protocol with dependency on intermediary services. Blockchain technology provides an effective solution to these problems. During an E-commerce transaction, customers and merchants can conduct transactions under two blockchains: the digital currency blockchain and the product supply chain blockchain, using Hash Time Locked Contracts (HTLCs) to facilitate cross-chain transaction and achieve transaction atomicity. This paper designs a blockchain-based scheme and implements a cross-chain experiment targeted at atomicity in E-commerce transaction.

Keywords: Blockchain · Atomicity · E-commerce · Hash Time Locks

1 INTRODUCTION

E-commerce has become an integral part of the global economy. However, existing E-commerce systems still face numerous challenges, particularly in ensuring atomicity throughout transaction phases. Based on the concept proposed by J.D. Tygar in 1996, the atomicity of E-commerce is divided into three levels: money atomicity, goods atomicity, and certified delivery atomicity[1]. Money atomicity effect the transfer of funds from one party to another without the possibility of the creation or destruction of money. Goods atomicity ensures that if a buyer has paid, they must receive the product and vice versa. Certified delivery atomicity involves mutual verification of the product content and quality by both buyer and seller. Thus, fulfilling these three types of atomicity is crucial for protecting the rights of both parties and ensuring the fairness of transactions.

So far, existing solutions to the atomicity problem in E-commerce transactions exhibit specific limitations. DigiCash, based on the blind signature protocol, primarily focuses on ensuring customer anonymity. However, there is a potential for double-spending, which violates the requirement for money atomicity. The SSL and SET protocols were initially designed to address the security of online credit card transactions,

A. Haldorai et al. (eds.), *Proceedings of the 2024 3rd International Conference on Artificial Intelligence, Internet and Digital Economy (ICAID 2024)*, Atlantis Highlights in Intelligent Systems 11,
https://doi.org/10.2991/978-94-6463-490-7_41

focusing on the security of data transmission and identity verification, rather than the integrity of the transaction process. Therefore, while they meet the requirements for money atomicity, they fail to ensure goods atomicity. The Netbill protocol can ensure the highest level of certified delivery atomicity, but it relies on the Netbill Server as a trusted intermediary[2]. This dependency introduces the risk of hacker attacks.

The integration of blockchain technology with E-commerce brings multiple benefits. In E-commerce transactions, parties can use blockchain to store transactional funds and goods information, with funds and goods managed on separate blockchains. Using homomorphic encryption technology, it is possible to effectively protect information about transaction products on the chain, transaction amounts, and the real identities of the transaction parties. Additionally, sidechain technology enhances the operational efficiency of blockchain, enabling it to meet the demands of large-scale E-commerce transactions. Hash Time Locked Contracts ensure that transactions are initiated by recognized parties and completed within a set timeframe, or else the transaction is automatically revoked[3]. These technologies together enhance the security and efficiency of transactions.

By building a cross-chain transaction model based on the Hyperledger Labs Weaver interoperability platform and the Fabric blockchain[4], [5], this paper attempts to address the goods atomicity issues present in E-commerce systems.

- This paper explores the cross-chain transaction issues between the currency chain and the product supply chain.
- Designing a blockchain cross-chain experiment based on Weaver, studying the resolution of goods atomicity issues in E-commerce.
- Demonstrating the feasibility, presenting experimental results.

2 ATOMICITY MODEL IN E-COMMERCE TRANSACTIONS

2.1 Scheme Overview

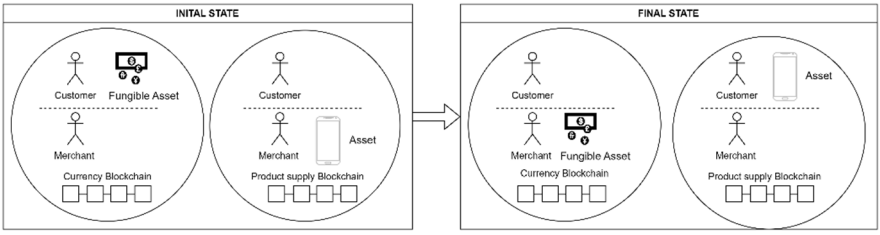

Fig. 1. Asset Exchange Model

This paper focuses on a transaction under E-commerce. Assuming there are two blockchains: a currency blockchain represented by common digital currencies such as Bitcoin and Ethereum[6], and a product supply chain maintained by various

stakeholders like manufacturers, distributors, retailers, customers, and logistic agencies, which stores transaction information of products[7]. It is assumed that customers and merchants use digital currency for financial transactions, and the products sold by the merchants can be traced on the product supply chain. The atom-exchange asset model is shown in Fig. 1.

2.2 Optimistic Transaction Flow

The optimistic transaction flow is shown in Fig. 2:

1. Merchant first checks the product availability on the product supply chain.
2. Customer checks the product information and price on the sales page.
3. Customer places an order, confirming the product information and price.
4. Customer executes a Hash Time Lock smart contract on the currency blockchain, locking the transaction amount.
5. Merchant executes a Hash Time Lock smart contract on the product supply chain, locking the transaction product.
6. Due to the activation of the smart contract, the supply chain coordinates with the warehouse to dispatch the product to customer.
7. After receiving the goods, the customer verifies if the product serial number matches the information on the product supply chain.
8. Upon confirming the match, the preimage is revealed on the product supply chain, transferring the ownership of the product.
9. The merchant unlocks the transaction funds on the currency chain based on the revealed preimage, thus completing the transaction.

After the above steps, the transaction was successfully completed.

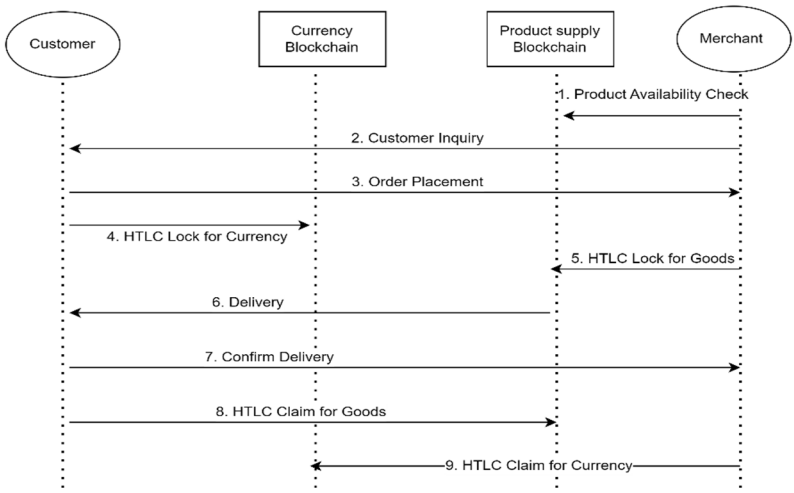

Fig. 2. Transaction Process

Next, the solution to various atomicity problems will be discussed from the perspective of E-commerce atomicity, as illustrated in Fig. 3. Firstly, the issue of money atomicity is ensured by the design of digital currency itself, which inherently includes a solution to the double-spending problem. Secondly, the atomicity of the goods is guaranteed by a hash time lock, which effectively meets the requirements for the atomicity of goods. Finally, the atomicity of confirmation and sending is ensured by the design of the supply chain itself. The supply chain needs to assign a unified number to each item of goods and support the customer's ability to trace the production history of the goods, thus ensuring the quality of the goods.

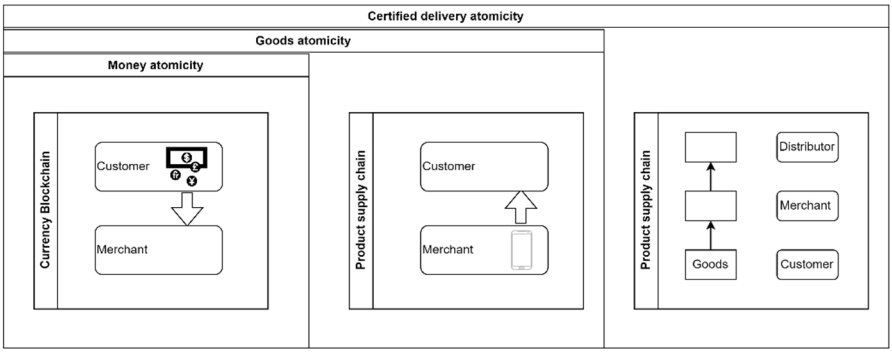

Fig. 3. Atomicity in E-commerce Transactions Under a Blockchain Solution

2.3 Goods Atomicity Model (HTLC)

This chapter will focus on solving the goods atomicity problem. Based on the definition of goods atomicity mentioned in Chapter 1, the experiment will show how to ensure that the exchange of money and goods is atomic. A simulation experiment was designed based on the Hyperledger Labs Weaver interoperability platform and the Hyperledger Fabric network.

In this study, the currency chain and the product supply chain are simplified into the chaincode "simpleasset". This chaincode provides management functions for both types of assets (goods and currency), including their creation, updates, and queries.

Based on the model described in Chapter 2, experiments were conducted using the Weaver framework's command-line tool, fabric-cli[1]. the tool implements the functionalities required for asset exchange based on HTLC. A complete cross-chain transaction experiment primarily includes the following processes: generating a key, locking the transaction amount, verifying the locked transaction amount, locking the goods, verifying the locked goods, claiming the goods, and claiming the transaction amount.

For ease of writing, this paper introduces the following symbols to describe the roles in the E-commerce transaction process: C represents the customer; M represents the merchant; cc represents the currency chain, and rc represents the supply chain. It is

[1] Fabric CLI: A CLI for interacting with the Fabric test-net and relays.

assumed that the customer and the merchant reach an agreement where customer spends 1500 tokens to purchase merchant's product, phone04.

1. Setup: Generate Hashes Secret. Customer generates preimages s and hashes them to produce H:

$$H=SHA256(s) \tag{1}$$

2. Customer Lock for Currency: The customer uses a hash lock set to H on the currency chain, specifies the recipient, locks 1500 fungible tokens, and sets a transaction timeout limit of 3600 seconds as part of the transaction. Once the lock is successful, a contract ID is generated for subsequent verification and claims processes.

$$CID_{cc}=Lock\ (cc, C, M, H, 3600, token1:1500) \tag{2}$$

3. Merchant Verify Lock for Currency: The merchant verifies that the tokens have been successfully locked by querying the contract ID on the currency chain. Only after confirming that the customer has locked the agreed 1500 tokens, can the merchant proceed with the next steps of locking the goods and shipping them.

$$Status=IsLock(cc, C, M, CID_{cc}, token1:1500) \tag{3}$$

4. Merchant Lock for Goods: The merchant similarly locks the goods (phone04) on the product chain. This process also requires specifying the recipient (the customer), a hash value H, product information, and a time T/2. Once the lock is successful, a contract ID is generated. The lock duration on the product chain is set to be shorter than that on the currency chain to prevent situations where the customer's time lock might expire and the currency is unlocked prematurely, while the goods remain locked.

$$CID_{rc}=Lock\ (rc, M, C, H, 1800, phone:phone04) \tag{4}$$

5. Customer Verify Lock for Goods: The customer verifies whether the goods have been successfully locked by querying the contract ID. This confirms that the merchant has shipped the goods.

$$Status=IsLock(rc, M, C, CID_{rc}, phone:phone04) \tag{5}$$

6. Customer Claim for Goods: The customer uses the previously generated preimage to claim the locked goods. This step verifies that the customer is the legitimate recipient of the transaction, exposes the preimage on the supply chain, and allows customer to retrieve the goods.

$$Status=Claim(rc,M,C,s, phone:phone04) \tag{6}$$

7. Merchant Claim for Goods: After the customer successfully claims the goods, the merchant uses the same preimage to claim the locked transaction amount. This step verifies that the merchant (Bob) is a legitimate participant in the transaction and allows him to access the tokens locked in the transaction.

$$Status=Claim(cc,C,M,s, token1:1500) \quad (7)$$

2.4 Threat Model: Premature Unlocking of Locked Assets

A possible way to attack the atomicity of goods, where customers or merchants do not want to abide by a contract and unlock locked assets prematurely. the smart contract will prevent such actions until the time lock expires, at which point the assets will automatically revert to the party that initially locked them.

If the current time is less than the expiry time, the smart contract will refuse the request to unlock the asset. The smart contract is executed by multiple machines within the blockchain network, utilizing a consensus mechanism to ensure that any malicious actions by individual machines do not affect the outcome once consensus is reached. The related smart contract pseudocode is as follows:

```
Function unlockAssetCommon
Inputs:
ctx - Transaction context interface
    expiryTimeSecs - Expiry time in seconds
  locker - Identifier for the entity who locked the asset
  contractId - Unique identifier for the contract
Begin
   ...
   Step 1: Get the current time in seconds
     currentTimeSecs = uint64(time.Now().Unix())

   Step 2: Compare current time with expiry time
     If currentTimeSecs < expiryTimeSecs
       Return "cannot unlock asset associated with the contractId [contractId] as the expiry time is not yet elapsed"
   ...
   Return success
End
```

3 PERFORMANCE EVALUATION

This chapter discusses the performance of cross-chain transactions in blockchain, utilizing two types of hardware configurations and different numbers of Fabric network nodes. The hardware configurations are divided into low-end (2 cores, 8GB RAM, 32GB storage) and high-end (4 cores, 16GB RAM, 32GB storage). The experiment considered both single-node and dual-node setups to analyze the impact of node count on performance. Performance data were collected at multiple stages using the Linux 'time' command, including network initialization, ledger initialization, the entire transaction process, key generation, locking and verifying transaction amounts, locking and verifying goods, and claiming goods and transaction amounts.

3.1 Impact of Hardware Configuration

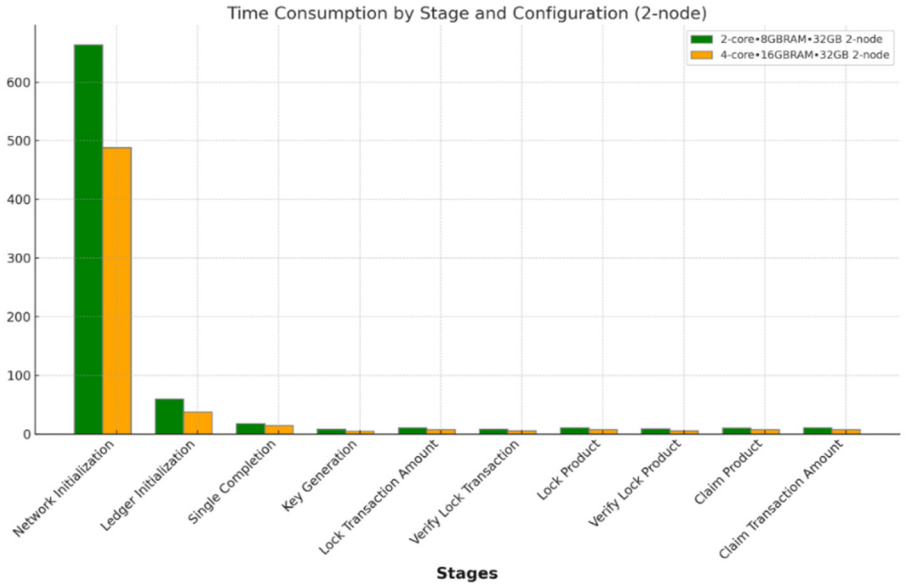

Fig. 4. Time Consumption at Various Stages Under Different Configurations

Fig. 4 displays the time consumption at each stage under different configurations for dual-node (2-node) setups. By comparing the green and orange bars, we can observe that across all stages, the execution time with the 4-core configuration is generally shorter than with the 2-core configuration, demonstrating that higher core counts and memory contribute to improved system performance. Especially during the network initialization phase, the 4-core configuration saves more time compared to the 2-core configuration, indicating that stronger hardware configurations can significantly enhance performance when handling more complex dual-node setups.

3.2 Impact of Node Count

Fig. 5 displays the time consumed at each stage under the same hardware configuration for single-node and dual-node setups. We found that, except during the network initialization stage, the differences in time between single-node and dual-node configurations are not significant in other stages. These observations indicate that increasing the number of nodes has a limited impact on system performance for tasks that do not involve extensive network communication. However, during the network-intensive initialization phase, an increase in nodes significantly affects performance. This helps us to understand in greater detail the specific impacts of different network configurations on overall system performance.

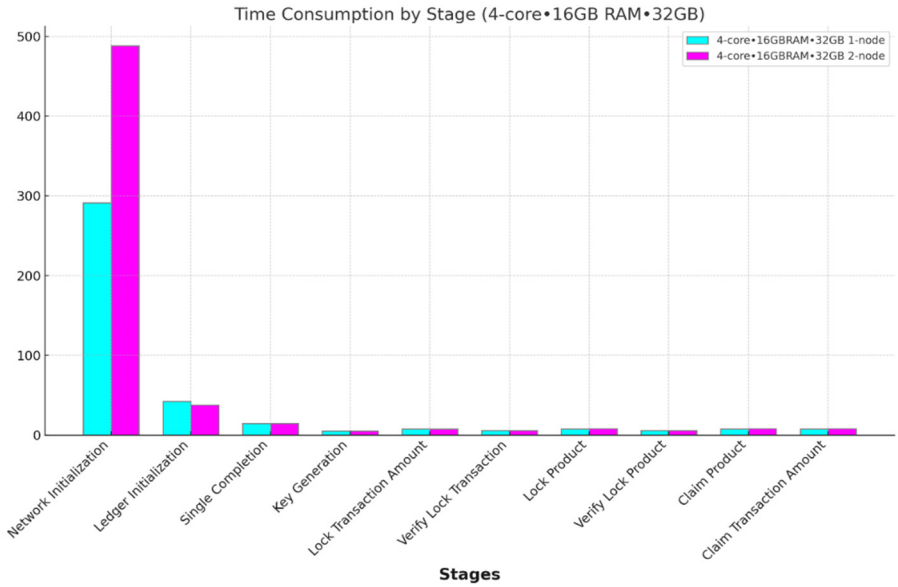

Fig. 5. Time Consumption at Various Stages Under Different Node Counts

4 CONCLUSION

This paper, through reviewing relevant literature, found that traditional E-commerce protocols rarely consider the atomicity of goods, while Hash Time Locked Contracts in blockchain cross-chain solutions effectively meet the requirement for goods atomicity. Therefore, an E-commerce transaction model incorporating blockchain cross-chain technology has been established. This model integrates the currency chain and the product supply chain to facilitate cross-chain asset exchanges, effectively ensuring the atomicity of transactions and preventing dishonest actions by any party.

Furthermore, this study conducts a detailed performance analysis and quantifies the time required at each stage of the transaction process. This includes from the initialization of the transaction to the final exchange of currency and goods, providing empirical data on the impact of hardware configurations and the number of network nodes on transaction efficiency. The results show that stronger hardware and a multi-node configuration can significantly enhance network performance, especially during network initialization and asset locking stages.

Future work could further explore the potential and challenges of blockchain technology in broader E-commerce scenarios.

REFERENCES

1. J. D. Tygar, "Atomicity in electronic commerce," in *Proceedings of the fifteenth annual ACM symposium on Principles of distributed computing - PODC '96*, Philadelphia, Pennsylvania, United States: ACM Press, 1996, pp. 8–26. doi: 10.1145/248052.248054.
2. B. Cox, J. D. Tygar, and M. Sirbu, "NetBill Security and Transaction Protocol.," in *USENIX Workshop on Electronic Commerce*, 1995. Accessed: Apr. 11, 2024. [Online]. Available: https://www.usenix.org/publications/library/proceedings/ec95/full_papers/cox.ps
3. "Overview of Block Chain Cross Chain Technology | IEEE Conference Publication | IEEE Xplore." Accessed: Apr. 13, 2024. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9410147
4. E. Androulaki *et al.*, "Hyperledger fabric: a distributed operating system for permissioned blockchains," in *Proceedings of the Thirteenth EuroSys Conference*, Porto Portugal: ACM, Apr. 2018, pp. 1–15. doi: 10.1145/3190508.3190538.
5. K. Narayanam, V. Ramakrishna, D. Vinayagamurthy, and S. Nishad, "Atomic cross-chain exchanges of shared assets," in *Proceedings of the 4th ACM Conference on Advances in Financial Technologies*, in AFT '22. New York, NY, USA: Association for Computing Machinery, Jul. 2023, pp. 148–160. doi: 10.1145/3558535.3559786.
6. G. Wood, "Ethereum: A secure decentralised generalised transaction ledger," *Ethereum project yellow paper*, vol. 151, no. 2014, pp. 1–32, 2014.
7. M. K. Lim, Y. Li, C. Wang, and M.-L. Tseng, "A literature review of blockchain technology applications in supply chains: A comprehensive analysis of themes, methodologies and industries," *Computers & Industrial Engineering*, vol. 154, p. 107133, Apr. 2021, doi: 10.1016/j.cie.2021.107133.