Multi-Sessions Mechanism for Decentralized Cash on Delivery System

Nghia Duong-Trung[1], Xuan Son Ha[2], Tan Tai Phan[3], Phuong Nam Trieu[4],
Quoc Nghiep Nguyen[5], Duy Pham[6], Thai Tam Huynh[7], Hai Trieu Le[8]
Can Tho University of Technology Can Tho city, Vietnam[1,2,4,5,8]
FPT University, Can Tho city, Vietnam[1,2]
National Chiao Tung University, Taiwan[3]
Hanoi University Science Technology, Ha Noi, Vietnam[6]
Transaction Technologies Pte. Ltd., Singapore[7]

***Abstract*—To date, cash on delivery (COD) is one of the most popular payment methods in developing countries thanks to the blossom of customer-to-customer e-commerce. With the widespread of a very small business model and the Internet, online shopping has become part of people's daily activity. People browse for desirable products at the comfort of their homes and ask the online vendor that a shipper can deliver the merchandise at their doorstep. Then, COD allows customers to pay in cash when the product is delivered to their desired location. Since customers receive goods before making a payment, COD is, therefore, considered as a payment system. However, the crucial issue that previous research has not yet addressed is that their models only support single delivering session at a time. More precisely, if the current buyer is not available to receive the goods, the shipper has to wastefully wait for the complete payment and he/she cannot start shipping another merchandise. The tracking system seems to poorly handle this issue. In particular, we propose a multi-session mechanism, which consists of blockchain technology, smart contracts and hyperledger fabric platform to achieve distributed and transparent across delivering sessions in the decentralized markets. Our proposed mechanism ensure the efficiency of delivering process. The authors release our sources codes for further reproducibility and development. We conclude that the integration of multi-session mechanism and blockchain technology will cause significant efficiency across several disciplines.**

***Keywords*—Blockchain; cash on delivery; multi-sessions; decentralized system**

I. Introduction

With the adoption of modern technology and the Internet, selling products online has become a very active market in developing countries. There is an immense need to have a delivery solution of any physical items ranging from catering, beverages, clothing and home facilities. Meeting the needs of today's customer-to-customer e-commerce, many third parties have launched delivery services which utilize blockchain technology. It seems counter-intuitive that, in developing countries like Vietnam, credit card and online payment are not widely used in the market. People still prefer to pay in cash because they want to make sure that the products must be in perfect condition.

Cash on Delivery (COD) allows customers to pay in cash when the products are delivered to their home or a location they choose. This is sometimes called a payment system because customers receive goods before making a payment. COD has become increasingly popular in recent years and been considered one of the main payment methods in many countries [1], [2], [3]. Among research articles, most investigated payment methods is in general, rather than focusing on COD in particular. Transfer agents are often used as postal services, but usually, consumer and business shipments will be sent to COD by courier companies, commercial truck forwarders or organizations own delivery services. COD sales usually involve a delivered fee charged by the shipping agents and is usually paid by the buyer. In retail and wholesale transactions, shipments rely on COD-based payment method when the buyer does not have a credit account and the seller does not choose a payment method in advance. COD postal services [4] were first introduced in Switzerland in 1849, India and Australia in 1877, the United States in 1913, Canada in 1922 and the United Kingdom in 1926. Particularly in Vietnam, COD is accepted by almost online vendors and customers.

However, the crucial issue that previous research has not yet addressed is that their models only support single delivering session at a time. During a working day, a shipper can take as many orders from customers across the local area. Then, the shipper delivery the products sequentially. If the current buyer is not available to receive the goods, the shipper has to wastefully wait for the complete payment and he/she cannot start shipping another merchandise. The tracking system seems to ineffectively track this issue. Consequently, a mechanism is missing in the buyer and seller's dilemma [5]. Addition to the current seven core components, e.g. product delivery, product payment, delivery trust, payment trust, escrow account, legal document and reputation system, the authors introduce a new part that can be integrated into the dilemma. To the best of our knowledge, this novel idea is firstly investigated and implemented by the authors.

The rest of the paper is organized as follows. The authors present related research in the field of COD and decentralized system in Section II. Then in Section III, the authors summarize the most important technical background for comprehending the proposed mechanism. The core contribution of the paper is presented in Section IV. Next, several real-world scenarios and remarks are demonstrated in Section V. Finally, the authors make conclusion in Section VII.

II. Related Work

Hanan and Salah have mentioned some limitations of proof of delivery (POD) process which relies on a trusted third party to implement the process [6]. Therefore, a new POD process using Blockchain technology from Ethereum has been introduced with the number of transportations through several intermediaries by their research. Besides, a dual deposit mortgage mechanism is used for parties to comply with the contract. The development of current e-commerce and the important role of Blockchain technology has been dedicated in Ha et al. research [7]. Besides, the limitation of the traditional CoD model is mentioned in detail such as trusted in the third party, order management, and the payment process between the parties in the system. So that, Blockchain technology using Hyperleder and smart contract is built to solve the issues of COD.

Camp et al. [8] had provided a digital purchasing method with a digital token on the network. They offer to issue invoices signed by sellers and digital goods which have been encrypted and sent from a seller to a customer. The participants such as sellers and buyers will be anonymous, making transactions through commitments by signing confirmation. There are no legal or property constraints. Le et al. [9] has mentioned the important roles of blockchain, especially, the decentralized users model, to builds the transportation process and offers mechanisms to promote and ensure the interests of participating parties. The benefits of the seller are enhanced and penalized the shippers who deliberately cheated. Therefore, the real data has applied to the system so that the delivery of multiple senders suitable for their Blockchain system is transparent. The process is built by all cash payments.

Altawy et al. [10] have compared the differences between buying handicrafts using cash payment and buying goods via the Internet. Online shipping needs more trust and the information of the parties who join the system to perform several actions such as making payment, delivery, and making sure the right items. Besides, the types of e-commerce used in trading which help the process faster. Anonymously purchase of a buyer is a big concern so that the Lelantos system has built based on Blockchain to detect and cancel the anonymous purchases which affect on the trading process. Discussing trading on a digital platform with the trust of the participants, Asgaonkar and Krishnamachari [11] have issued a deposit protocol for trading by the participants. They applied the trust of the Blockchain system to make payment by participants without a trusted third party. This protocol asked dual-deposit amount on the contract with payment of both sides and the price of the product is always fixed. However, the product for the parties conducting the transaction has not been verified.

Halaweh presented the rapid growth of the COD model as an important method in making payment and transport in e-commerce [12]. The author gave statistics on the study of the COD process to customers and conducted the prediction and testing of factors affecting the COD process by using a questionnaire methodology. Moreover, it also predicts the factors that affect the COD process such as safety and security of the system for the products and privacy of the participants. Barkhordari et al. have proposed a concern of the bank using the Internet to negotiate and solve customers' needs [13]. That is the trust of customers and the security of the system. Their article has deployed a survey regarding influencing factors to payment transactions. That surveys emphasize two factors, e.g. trust and security. Similar to the above survey, the payment and transportation in COD need the trust of the participants and the security of the system.

OpenBazaar platform [14] provides a procedure for making deposits when agreed by the buyer, seller, and trader which is known as multi-signature escrow. A third party will participate in the process of trading an item called a moderator. The moderator will resolve disputes when a problem occurs. Bitcoin currency will be used for payment in transactions. The process has not yet delivered the person, the role of the deliverer is not specified. Besides, the need for a third party to resolve the dispute will consume more assets and time of the parties involved. This has been solved by using the smart contract as a third party as well as solving the problems of the parties involved in the contract terms.

COD model using two smart contracts is introduced by Le et al. [15]. An outcome is a positive deployment process of decentralized applications, which enforces contracts with the exact terms. The price of the order is deposited by the participants. However, the management of orders becomes more difficult when the data and the number of orders scale up. According to the process, the second contract will be implemented immediately after the first contract is executed. The implementation is based on the memorization of the address of each contract. As a result, it is a major limitation in the deployment process if the process is applied to multiple orders since the system could not perform several contracts at the same time.

III. Materials and Technical Background

A. Cash On Delivery

COD is a service of collecting money collected in the group of services of buying goods by post (Cash On Delivery or Collect On Delivery). It is the association between postal service and money transfer service with several stages: First, the shipper proceeds to send the goods to the recipient via a courier company. Next, the courier company will send the goods to the recipient by delivery service. The branch or post office of the delivery company delivers the goods to the recipient and the receiver makes payment. After that, the branch or delivery office issued a COD check (similar to a money order) sent to the shipper. From this point on, COD will be similar to a money transfer service.

B. Blockchain Technology

Blockchain is a list of continuously written logs, called blocks, linked by encryption. Each block contains the previous block's cryptographic hash function, timestamp, and transaction data. Each block has a block header and a body containing data and hash values of the previous block. The hash value is the result of a hash function. The hash function transforms data of any length into a fixed-length string or numeric value, such as 256 bits (32 bytes) with SHA256. Blockchain is a technology that allows secure data transmission based on an extremely complex encryption system, similar to accounting books of a company where cash is closely monitored. In this case, the blockchain is an accounting ledger [16] that

works in the digital field. A special feature of blockchain is that transactions are done at a high level of trust without disclosing information. All types of business and management can participate in the network and use the properties of the Blockchain system to ensure transparency of stakeholders.

C. Ethereum

Ethereum [17], [16], [18] is a distributed, public, and open-source computing platform based on blockchain technology. It features smart contracts (scenarios), facilitating online contract agreements. This platform includes a complete Turing virtual machine and Ethereum Virtual Machine [19], [20], which can execute scripts using an Ethereum computer network. Ethereum also provides a cryptocurrency called Ether, which can be transferred between accounts and used to pay peaches to help perform calculations. Gas is an internal transaction pricing mechanism, used to minimize spam and allocate resources on the network. When creating, each transaction is charged by a certain amount of gas, its purpose is to limit the amount of work needed to execute the transaction and pay for this execution at the same time.

D. Smart Contracts

A cryptocurrency is a decentralized platform that a distributed ledger is used to interact with virtual money. A contract is an instance of a computer program that executes on the Blockchain. Users transfer money by publishing transactions and interacting with contracts in the cryptocurrency network where information is propagated, data is stored among miners or network's nodes. An underlying cryptocurrency system supports the utilization of smart contracts. A smart contract contains program code, a stored file and an account balance. Any user can submit a transaction to an append-able-only log. When the contracted is created, its program code cannot be changed. An append-able-only log, called a blockchain, which imposes a partial or total arrangement on submitted transactions is the main interface provided by the cryptocurrency. The integration of smart contract in COD has been discussed in [15].

E. Decentralized Applications – dApps

DApps [21], [22], [23] are as similar as normal applications except that they are completely decentralized. It is also controlled by nodes running Ethereum networks. These dApps do not depend on any central server or third party for operating, and therefore, without the central point of failure. Thanks to the blockchain technology, the database is encrypted and stored in a decentralized fashion. By using a modern mean of communication protocols, participants can store and retrieve data without the risk of censor and intervention [24]. DApps are expected to resist attack and censorship while being able to operate in a fully autonomous model.

IV. Proposed Multi-Sessions COD Process

A. Abstract Model for a COD system

The authors start this session by presenting a general description of multi sections in COD transport process. The abstract model for COD system is illustrated in Fig. 1. First, the product information is uploaded to the sale contract by the seller where the buyer can verify through the app, and send the purchase request. The sale contract will trigger the purchase contract for activating the term that the buyer has to transfer the amount of mortgage money as same as the valuation of the order to the purchase contract. Thus, a delivery request is sent to the system by the shipper after seeing an available order. The sale contract will trigger to delivery contract for activating the terms that the shipper and seller have to transfer mortgage money and delivery fee, respectively. Finally, the money will be transferred to the seller from the purchase contract. The mortgage money and delivery fee will be transferred to the shipper by the delivery contract which used to store the delivery fee and order money from the seller and shipper respectively.

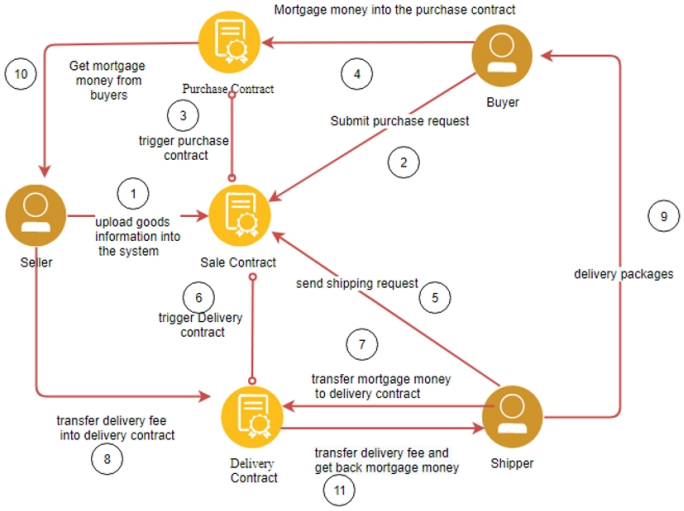

Fig. 1. General description of our proposed multi sections of COD transport process.

B. Detailed COD Scenarios

1) Shipper successfully delivers goods and the buyer successfully receives the goods: The sale contract triggers to purchase contract, seller contract and delivery contract for activating the money transferability function. The order money will be transferred to the seller by purchase contract when the buyer confirms successful delivery to the delivery contract. On the other hand, the delivery contract returns the mortgage money to the shipper which is already deposited before the receiving the order, and the delivery fee is also transferred to the shipper by seller contract immediately. This scenario is illustrated in Fig. 2.

2) Shipper unsuccessfully delivers goods: The sale contract triggers to purchase contract, seller contract and delivery contract for activating the money transferability function. The order mortgage money and delivery money will be transferred to the seller by seller contract and delivery contract due to the shipper failed delivery, then the purchase contract returns the mortgage money to the buyer. This scenario is illustrated in Fig. 3.

3) Buyer refuses to receive goods: The sale contract triggers to purchase contract, seller contract and delivery contract

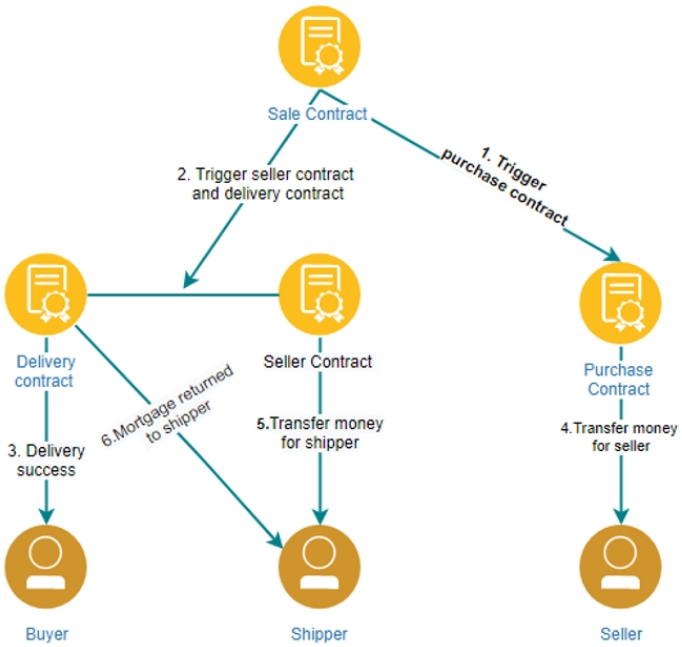

Fig. 2. Case 1: Shipper successfully delivers goods and buyer successfully receives the goods.

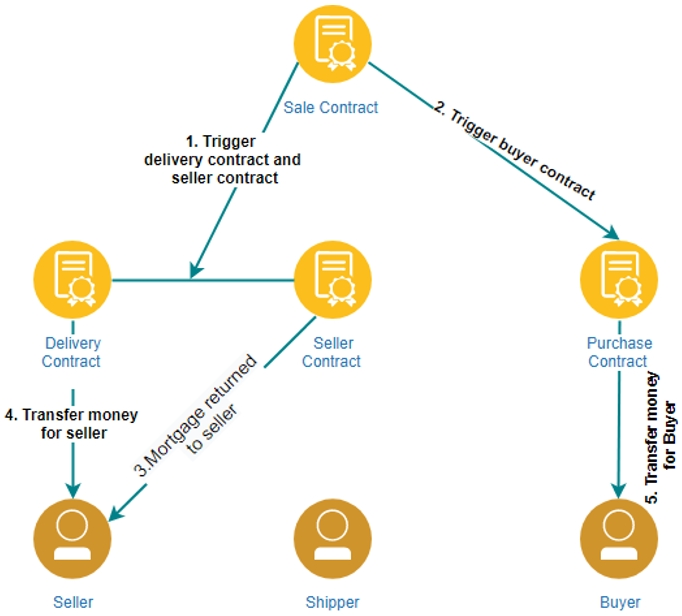

Fig. 3. Case 2: Shipper unsuccessfully delivers goods.

for activating the money transferability function when the buyer does not receive the order. The purchase contract utilizes the mortgage money from the buyer to make a payment for the seller, this means that the buyer will be lost their deposit because of "Booming order". On the other hand, the shipper will receive the mortgage money and delivery fee from the delivery contract and seller contract, respectively. This scenario is illustrated in Fig. 4.

4) Seller provides incorrect goods: The sale contract triggers to purchase contract, seller contract for activating the

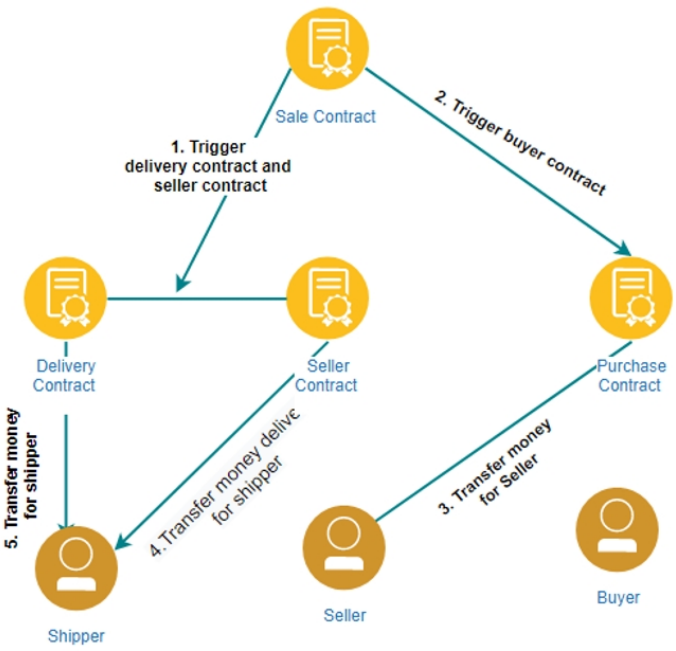

Fig. 4. Case 3: Buyer refuses to receive goods.

money transferability function when the seller is wrong order. The shipper checks order from sale contract and notifies the order is wrong, then mortgage money which stored in purchase contract will be returned to the buyer because of failure transaction. Therefore, the seller returns the mortgage money to the seller. This scenario is illustrated in Fig. 5.

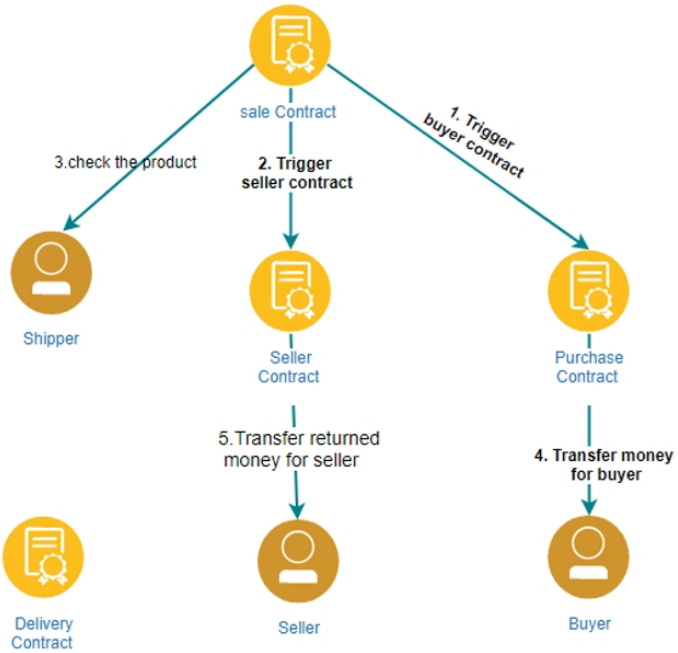

Fig. 5. Case 4: Seller incorrectly provides goods.

C. Algorithms

The algorithm (1) is the money transferability algorithm. The temporary address is generated at line 1, 2 and 3. Line 4 is to trigger the seller contract for transferring the money to the shipper. The money is transferred to the seller by triggering purchase contract at line 5. Line 6 is to trigger the delivery contract to get back the mortgage money.

Algorithm 1 Money_transferability_algorithm

Input: Order code
Output: Trigger the contracts to execute money transferability and get back mortgage money

1: Address seller_deposit_temp
2: Address buyer_deposit_temp
3: Address shipper_deposit_temp
4: Trigger DepositSeller(seller_deposit_temp) contract and seller function to transfer money
5. Trigger Depositbuyer(buyer_deposit_temp) contract and buyer function to transfer money
6. Trigger DepositShiper(shipper_deposit_temp) contract and shipper function to get back mortgage money

The algorithm (2) is the refund algorithm. Line 1, 2, and 3 create temporary addresses. Triggering the seller contract to get a mortgage is executed at line 4. Trigger purchase contract to get back mortgage is done at line 5. Line 6 is to trigger delivery contract to transfer the mortgage of the shipper's products to the seller.

Algorithm 2 Refund_algorithm

Input: Order code
Output: Trigger the contracts to execute money transferability and get back mortgage money

1: Address seller_deposit_temp
2: Address buyer_deposit_temp
3: Address shipper_deposit_temp
4: Trigger DepositSeller(seller_deposit_temp) contract and function to get back mortgage money
5. Trigger Depositbuyer(buyer_deposit_temp) contract and function to get back mortgage money
6. Trigger DepositShiper(shipper_deposit_temp) contract and shipper function to transfer money

The algorithm (3) is called the seller failure algorithm Line 1, 2, and 3 create the temporary addresses. Shipper checks the order and returns the money if the order is not correct is done at line 4. Delivery mortgage money of the seller is a return to the seller is executed at line 5.

1) Case 1: The failure is caused by the shipper. The mortgage money as same as the valuation of the order is triggered to refund payment method as set at line 1 in the algorithm (4).

2) Case 2: The failure is caused by the buyer. When the situation happens, the shipper will receive the package and delivery fee. It is done at line 1 in the algorithm (5) to trigger the transfer money method.

Algorithm 3 Seller_failure_algorithm

Input: Order code
Output: Trigger the contracts to execute money transferability and get back mortgage money

1: Address seller_deposit_temp
2: Address buyer_deposit_temp
3: Trigger DepositSeller(seller_deposit_temp) contract and function to get back mortgage money
4: Trigger Depositbuyer(buyer_deposit_temp) contract and function to get back mortgage money

Algorithm 4 Case 1: Shipper_is_failed

Input: Order code
Output: Trigger refund money method

1: Trigger refund money method

Algorithm 5 Case 2: Buyer_is_failed

Input: Order code
Output: Trigger refund money method

1: Trigger transfer money function

3) Case 3: The shipment is done successfully. The buyer transfers money to the seller. The seller transfers money to the shipper. The shipper takes the money. Line 1 in the algorithm (6) triggers the transfer money method.

Algorithm 6 Case 3: The shipment is done successfully

Input: Order code
Output: Trigger transfer money method

1. Trigger transfer money function

4) Case 4: The order is wrong because of the seller. Shipper checks the order at line 1 and 2 in the Algorithm (7). When the order is wrong, the ReUnfundSellerFail function is activated at line 3. The shipment stops unsuccessfully.

Algorithm 7 Case 4: Seller_is_failed

Input: Order code
Output: Trigger the reunfundSellerFail method

1: Trigger package [order code] name = name
2: If name !=_name
3: Trigger ReUnfundSellerFail function
4: EndIf

V. Experiments

On a blockchain Ethereum model, all of the interaction with the blockchain such as contract reaction, command translation, execution of function has to pay a fee which

called gas. Gas costs depend on the complexity and logic of that function. It is calculated based on how much computer resources will be required to perform the function. So that, code optimization is important in Ethereum to be able to save costs. The measurement in four experimental cases of COD has also performed. Case 1, the process takes place normally, the receiving and shipping take place successfully. Case 2, we will refer to the transaction error due to the problem on the shipper. Case 3 is a transaction error due to a buyer problem. Finally, the seller delivers the wrong product, e.g. Case 4. The details implementation of these cases is presented in this section. A complete codes solution is publicized on the authors' GitHub repository[1] (CC BY 4.0) to engage further reproducibility and improvement.

A. Case 1: Transport Successfully

TABLE I. CASE 1: STEP 1

From	0xca35b7d915458ef540ade6068dfe2f44e8fa733c
To	Seller.setPackage(string,uint256,string) 0x692a70d2e424a56d2c6c27aa97d1a86395877b3a
Transaction cost	106490
Execution cost	83618

1) Step 1: Seller creates a package. See Table I:

TABLE II. CASE 1: STEP 2

From	0x14723a09acff6d2a60dcdf7aa4aff308fddc160c
To	DepositBuyer.(constructor)
Transaction cost	270620
Execution cost	166432

2) Step 2: Buyer deposits an amount of money. See Table II:

TABLE III. CASE 1: STEP 3

From	0x14723a09acff6d2a60dcdf7aa4aff308fddc160c
To	Seller.buyItem(uint256,string,address) 0x692a70d2e424a56d2c6c27aa97d1a86395877b3a
Transaction cost	89622
Execution cost	66110

3) Step 3: Buyer buys goods. See Table III:

TABLE IV. CASE 1: STEP 4

From	0xca35b7d915458ef540ade6068dfe2f44e8fa733c
To	DepositSeller.(constructor)
Transaction cost	239581
Execution cost	143213

4) Step 4: Seller places a mortgage. See Table IV:

5) Step 5: Shipper places a mortgage and agrees to deliver goods. See Tables (V, VI, VII, and VIII):

B. Case 2: Transport Failure Caused by Buyer.

In this case, a flag is set to indicate that the buyer is the one who causes the transport cancellation. See Table IX.

[1]https://github.com/TrieuNam/Smart-Contract-Cash-on-delivery-4.0

TABLE V. CASE 1: STEP 5A

From	0x4b0897b0513fdc7c541b6d9d7e929c4e5364d2db
To	DepositShipper.(constructor)
Transaction cost	239169
Execution cost	142813

TABLE VI. CASE 1: STEP 5B

From	0x4b0897b0513fdc7c541b6d9d7e929c4e5364d2db
To	Seller.setShipperDepositAddress(uint256,address) 0x692a70d2e424a56d2c6c27aa97d1a86395877b3a
Transaction cost	43443
Execution cost	20635

TABLE VII. CASE 1: STEP 5C

From	0x4b0897b0513fdc7c541b6d9d7e929c4e5364d2db
To	Seller.setSellerDepositAddress(uint256,address) 0x692a70d2e424a56d2c6c27aa97d1a86395877b3a
Transaction cost	43465
Execution cost	20657

TABLE VIII. CASE 1: STEP 5D

From	0x4b0897b0513fdc7c541b6d9d7e929c4e5364d2db
To	Seller.setFlagBuyerAndShiper(uint256) 0x692a70d2e424a56d2c6c27aa97d1a86395877b3a
Transaction cost	56198
Execution cost	34798

TABLE IX. CASE 2

From	0x4b0897b0513fdc7c541b6d9d7e929c4e5364d2db
To	Seller.setFlagBuyerFail(uint256) 0x692a70d2e424a56d2c6c27aa97d1a86395877b3a
Transaction cost	56156
Execution cost	34820

C. Case 3: Seller Provides Incorrect Goods.

In this case, a flag is set to indicate that the seller is the one who causes the transport cancellation. See Table X.

TABLE X. CASE 3

From	0xca35b7d915458ef540ade6068dfe2f44e8fa733c
To	Seller.setFagSellerFail(uint256,string) 0x692a70d2e424a56d2c6c27aa97d1a86395877b3a
Transaction cost	34487
Execution cost	27319

D. Case 4: Shipper Fails to Deliver Goods. See Table XI.

TABLE XI. CASE 4

From	0x4b0897b0513fdc7c541b6d9d7e929c4e5364d2db
To	Seller.setFlagShipperFail(uint256) 0x692a70d2e424a56d2c6c27aa97d1a86395877b3a
Transaction cost	54439
Execution cost	33039

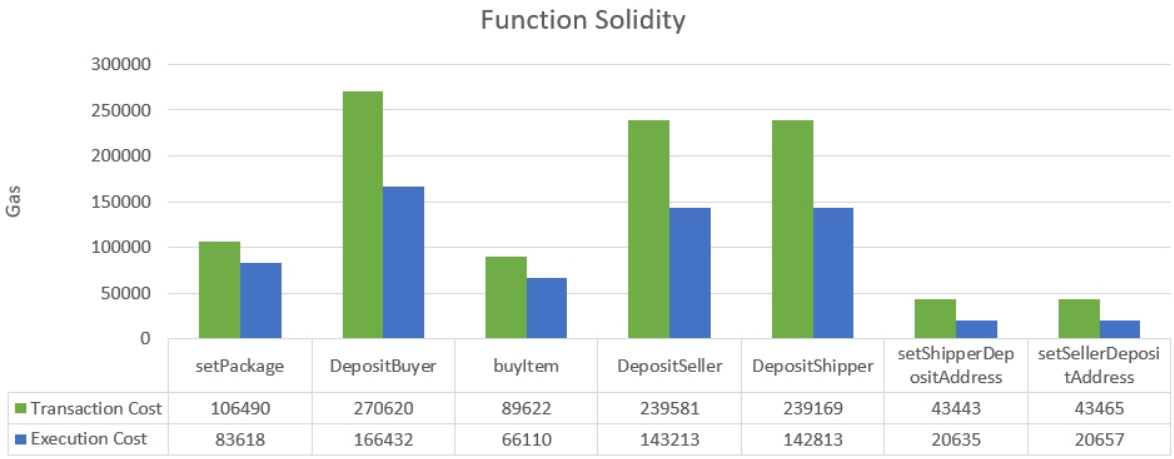

Fig. 6. Function solidity

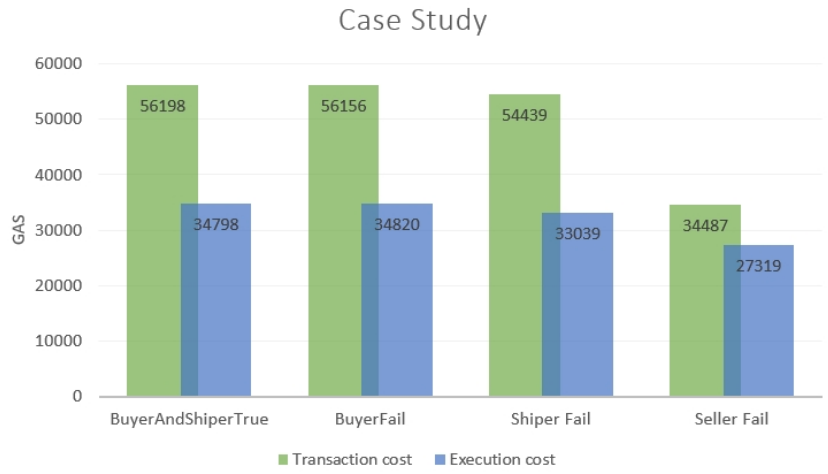

Fig. 7. Gas consumption in every case.

VI. Remarks

In the function solidity diagram, the gas consumption (Fig. 6), we see that the amount of gas in the modality does not pass the transaction and the execution are 300000 and 200000, respectively. The amount of gas increases due to the access to smart contracts as well as the complexity of the methods caused. In the Case Study diagram, e.g. Fig. 7, the amount of Gas in successful trading scenario shows that stable transactions and execution do not exceed 60000 and 40000 respectively. It is important to note that the amount of gas for contract transactions of the participants is very small. In case of transaction of the failure scenarios, the amount of the Gas will be lower than that of in the successful trade. The Gas will be decided by the smart contract when the execution stops, ensuring the amount of loss during the transaction process is insignificant.

VII. Conclusion

As we have demonstrated, the integration of multi-session mechanism in any cash on delivery systems is very effective. Our proposed idea is given to not only enhance the effectiveness of the shipper but also improve the overall performance of decentralized systems. The mechanism works transparently across participants. Several real-world scenarios have been discussed the feasibility of the proposed multi-sessions in boosting the performance and robustness of the COD systems. The crucial delivering issue that previous research has not yet addressed is sufficiently solved. Our proposed mechanism ensure the overall efficiency of delivering process. We are pleased to announce that a new core component of the buyer and seller's dilemma. The authors release our sources codes for further reproducibility and development. We believe that the integration of multi-session mechanism, blockchain technology and smart contracts will cause significant efficiency across several disciplines.

References

[1] M. Halaweh, "Cash on delivery (cod) as an alternative payment method for e-commerce transactions: Analysis and implications," *International Journal of Sociotechnology and Knowledge Development (IJSKD)*, vol. 10, no. 4, pp. 1–12, 2018.

[2] ——, "Intention to adopt the cash on delivery (cod) payment model for e-commerce transactions: An empirical study," in *IFIP International*

Conference on Computer Information Systems and Industrial Management. Springer, 2017, pp. 628–637.

[3] U. Tandon and R. Kiran, "Study on drivers of online shopping and significance of cash-on-delivery mode of payment on behavioural intention," *International Journal of Electronic Business*, vol. 14, no. 3, pp. 212–237, 2018.

[4] J. D. Alie and P. E. Vliek, "International cash-on-delivery system and method," Jul. 24 2007, uS Patent 7,249,069.

[5] A. Asgaonkar and B. Krishnamachari, "Solving the buyer and seller's dilemma: A dual-deposit escrow smart contract for provably cheat-proof delivery and payment for a digital good without a trusted mediator," *arXiv preprint arXiv:1806.08379*, 2018.

[6] H. R. Hasan and K. Salah, "Blockchain-based solution for proof of delivery of physical assets," in *International Conference on Blockchain*. Springer, 2018, pp. 139–152.

[7] H. X. Son, M. H. Nguyen, N. N. Phien, H. T. Le, Q. N. Nguyen, V. D. Dinh, P. T. Tru, and P. Nguyen, "Towards a mechanism for protecting seller's interest of cash on delivery by using smart contract in hyperledger," *International Journal of Advanced Computer Science and Applications*, vol. 10, no. 4, 2019. [Online]. Available: http://dx.doi.org/10.14569/IJACSA.2019.0100405

[8] L. J. Camp, J. D. Tygar, and M. R. Harkavy, "Anonymous certified delivery," Jun. 13 2000, uS Patent 6,076,078.

[9] N. T. T. Le, Q. N. Nguyen, N. N. Phien, N. Duong-Trung, T. T. Huynh, T. P. Nguyen, and H. X. Son, "Assuring non-fraudulent transactions in cash on delivery by introducing double smart contracts," *International Journal of Advanced Computer Science and Applications*, vol. 10, no. 5, 2019. [Online]. Available: http://dx.doi.org/10.14569/IJACSA.2019.0100584

[10] R. AlTawy, M. ElSheikh, A. M. Youssef, and G. Gong, "Lelantos: A blockchain-based anonymous physical delivery system," in *2017 15th Annual Conference on Privacy, Security and Trust (PST)*. IEEE, 2017, pp. 15–1509.

[11] A. Asgaonkar and B. Krishnamachari, "Solving the buyer and seller's dilemma: A dual-deposit escrow smart contract for provably cheat-proof delivery and payment for a digital good without a trusted mediator," *2019 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, pp. 262–267, 2018.

[12] M. Halaweh, "Intention to Adopt the Cash on Delivery (COD) Payment Model for E-commerce Transactions: An Empirical Study," in *16th IFIP International Conference on Computer Information Systems and Industrial Management (CISIM)*, ser. Computer Information Systems and Industrial Management, K. Saeed, W. Homenda, and R. Chaki, Eds., vol. LNCS-10244. Bialystok, Poland: Springer International Publishing, Jun. 2017, pp. 628–637, part 7: Various Aspects of Computer Security. [Online]. Available: https://hal.inria.fr/hal-01656252

[13] M. Barkhordari, Z. Nourollah, H. Mashayekhi, Y. Mashayekhi, and M. S. Ahangar, "Factors influencing adoption of e-payment systems: an empirical study on iranian customers," *Information systems and e-business management*, vol. 15, no. 1, pp. 89–116, 2017.

[14] "Sites like ebay or etsy but decentralized - our features." [Online]. Available: https://openbazaar.org/features/

[15] N. T. T. Le, Q. N. Nguyen, N. N. Phien, N. Duong-Trung, T. T. Huynh, T. P. Nguyen, and H. X. Son, "Assuring non-fraudulent transactions in cash on delivery by introducing double smart contracts," *International Journal of Advanced Computer Science and Applications*, vol. 10, no. 5, 2019. [Online]. Available: http://dx.doi.org/10.14569/IJACSA.2019.0100584

[16] G. Wood *et al.*, "Ethereum: A secure decentralised generalised transaction ledger," *Ethereum project yellow paper*, vol. 151, pp. 1–32, 2014.

[17] "Ethereum." [Online]. Available: https://www.ethereum.org/

[18] C. Dannen, *Introducing Ethereum and Solidity*. Springer, 2017.

[19] Y. Hirai, "Defining the ethereum virtual machine for interactive theorem provers," in *International Conference on Financial Cryptography and Data Security*. Springer, 2017, pp. 520–535.

[20] M. Wohrer and U. Zdun, "Smart contracts: security patterns in the ethereum ecosystem and solidity," in *2018 International Workshop on Blockchain Oriented Software Engineering (IWBOSE)*. IEEE, 2018, pp. 2–8.

[21] "What is a dapp? decentralized application on the blockchain." [Online]. Available: https://blockchainhub.net/decentralized-applications-dapps/

[22] S. Raval, *Decentralized applications: harnessing Bitcoin's blockchain technology*. " O'Reilly Media, Inc.", 2016.

[23] W. Mougayar, *The business blockchain: promise, practice, and application of the next Internet technology*. John Wiley & Sons, 2016.

[24] A. Wright and P. De Filippi, "Decentralized blockchain technology and the rise of lex cryptographia," *Available at SSRN 2580664*, 2015.