Towards a Mechanism for Protecting Seller's Interest of Cash on Delivery by using Smart Contract in Hyperledger

Ha Xuan Son[1], Minh Hoang Nguyen[2], Nguyen Ngoc Phien[3], Hai Trieu Le[4], Quoc Nghiep Nguyen[5], Van Dai Dinh[6], Phu Thinh Tru[7], and The Phuc Nguyen[8]

[1]FPT University, Can Tho City, Viet Nam
[2]Hanoi University of Science and Technology, Ha Noi, Viet Nam
[3]Center for Applied Information Technology, Ton Duc Thang University, Ho Chi Minh City, Vietnam
[3]Faculty of Information Technology, Ton Duc Thang University, Ho Chi Minh City, Vietnam
[1, 4, 5, 6, 7]Cantho University of Technology, Can Tho City, Viet Nam
[8]University of Trento, Trento, Italy

***Abstract*—In emerging economies, with the explosion of e-commerce, payment methods have increasingly enhanced security. However, Cash-on-Delivery (COD) payment method still prevails in cash-based economies. Although COD allows consumers to be more proactive in making payments, it still appears to be vulnerable by the appearance of a third party (shipping companies). In this paper, we proposed a payment system based on "smart contract" implemented on top of blockchain technology to minimize risks for parties. The platform consists of a set of rules that each party must follow including specific delivery time and place, cost of delivery, mortgage money; thereby, forcing parties to be responsible for their tasks in order to complete the contract. We also provided a detailed implementation to illustrate the efficiency of our model.**

***Keywords*—Blockchain; fintech; smart contract; customer; seller; shipper; cash on delivery; hyperledger**

I. Introduction

With the development of Internet, online retail is now growing significantly. As the payment trend is shifting from Cash-on-Delivery (COD) to online payments, COD continues to remain the most preferred type of payment in cash-based economies as customers can control more over online transactions to minimize the risk of fraud. By definition, COD is a type of transaction that payments are completed at the time of delivering. Otherwise, the product would be returned to the seller. According to Nielsen's Global Connected Commerce Survey, about 83% of Indian customers prefer COD instead of online payment. UAE has up to 60% of e-commerce transactions carried out by COD [1]. However, COD poses a major drawback as shopping process might be affected by shipping carriers – a third party. These risks include misplacing parcels, lost/damaged goods or even being hijacked by the delivery company. In addition, as payment is made between the customer and shipper which is not the seller, in case the shipping company has any financial problems, they may take advantage of the clients' collected money instead of returning them to the seller (e.g. bankrupt of City Link [2] in 2014, Hanjin Shipping [3] in 2017, GNN Express – a Vietnamese delivery company – owned customers up to US$ 230000 [4]). This situation is becoming more and more popular recently causing damage to the seller and loss of consumer confidence.

Being able to provide high transparency in transactions and e–commerce, blockchain technology recently is becoming the `de facto` technology for financial business worldwide. Blockchain is a digital ledger providing secure options for making and recording transactions, agreements and smart contracts – anything that needs to be recorded and verified as having taken place. With its potential, blockchain enables e-commerce to be faster, safer and more reliable to both sellers and customers. Among applications of blockchain for e–commerce, to resolve the issues of COD, in this paper, we focused on designing a peer–to–peer trading platform between clients and suppliers without the presence of a trusted third–party.

Our system is implemented using the concept of "smart contract", a core application of blockchain. Smart contract is built on top of a blockchain defining the rules between different parties who agree within the contract. Its mechanism eliminates the needs for trusted third parties as each party have to deposit some digital assets to the contract and the assets automatically get redistributed among the parties according to the contract if and only if all conditions are satisfied. Etherium [5] uses smart contract to exchange digital assets without a trusted third party. Toward our scenario, the sellers and shipping company will sign a smart contract to ensure interests/rights of the parties: the sellers always receive their payments/goods while the carriers receive his shipping fee.

Our system is built on Hyperledger fabric which is an open source project initiated by the Linux Foundation (on December 2015) [6], [7]. Hyperledger is developed on many sub-projects within it with the general purpose of supporting utilizing blockchain technologies for enterprises. The contribution of this paper is two-fold: (i) providing a framework to ensure the customer's payment; and (ii) reducing the risk of third-party in the COD process.

This paper is organized as follows. In the second section, we briefly review related work. The following section

illustrates the concept of blockchain technology and applied technique in detail. We described our proposed architecture with its main algorithm and describes the experimental designs and discuss the results in section fourth. Finally, Section 5 presents the conclusion and future works.

II. Related Works

One of the major problems of e-commerce globally is the selling and buying of goods among the parties over the Internet in which the traders may not trust their partners. Krishnamachari et al. [8] proposed the mechanism that executes a transaction with any kinds of assets by using the digital key and these processes do not need a trusted third-party. Additionally, the authors describe a transaction method which signs dual deposit for anti-fraud payment transactions and the delivery between two parties in which the trader can use the digital signature to verify. The seller and the customer use a pair of symmetric keys to verify goods. They use smart contracts to decide and handle sellers and customers by increasing deposits. But this paper has not yet analysis on a problem of shipping, if it is a physical product and the shipper fails to comply with the commitment, then the system is not resolved. For our article, it is recommended that the shipper join the system and mortgage a sum of money to ensure the reliability of the system. Our process is given to not only ensure the benefits of the seller but also prevent shipper's fraudulent. If the shipper has problems, such as loss of goods then the goods of the seller sent at the carrier will still be refunded in cash to the seller.

HR Hasan, K Salah [9] provides a delivery process, participants (sellers, carriers, and buyers) must mortgage a value. This value is double the value of the goods shipped if the successful contract value will be returned for parties. If it fails, Arbitrator will solve the problem when disputing. We provide a time-bound solution to complete the contract if it fails on time, the system will automatically resolve the dispute, based on the contract without Arbitrator.

Two party contracts [10] propose the process using hash and key that is shipped by the seller with goods. The seller will be using the key and hash to compare whether it matches my product. Participants will make a deposit to the contract. But when goods are delivered by the shipper with the same key then it is possible for the carrier to find a way to manipulate or steal a key. Therefore this trust is not guaranteed to affect seller very much.

Deposit on `localethereum.com` [11] is an environment that is implemented by Ethereum smart contract. They introduced a process of three agents including sellers, buyers and third parties funded to deposit. Sellers and buyers trade with each other when they accept the terms of trading. A third party will ensure agreement for the transaction if a dispute happens. Currently, a reliable mediator is always `localethereum.com`. In our approaches, the mediation section is smart contract handling without an additional third party to stand out and such third party generation also increases costs more than primarily.

BitBay [12] is a market that uses decentralized platforms. Sellers and buyers join the transaction and it eliminates the trusted third-party. Therefore, the third-party is its contract. The smart contract will be responsible for receiving the deposit, all funds will be kept in the contract until the transaction is completed. But when the seller and the buyer sign the contract, they will have to deposit a double amount of that product, it means that the item costs $1, the buyer will put in the contract is $2 and the seller is also $2. So the cost incurred is a consideration.

Our article applies blockchain technology [13] to create an operational chain, and we use distributed networks to ensure the consensus of all participants. The blockchain is applied to Bitcoin [14] to solve double spending problems. Participants trade on the blockchain network using private and public key. With the deployment of blockchain networks, smart contracts are included to make transactions between different users faster and more efficient [15]. We use hyperledger fabric [6] as an open source distributed ledger platform for building applications so the blockchain platform is effective and secure.

III. Our Approach

A. Blockchain and Applied Techniques

1) BlockChain: is a technology that enables secure data transmission based on an extremely complex encryption system, similar accounting book of a company where cash is closely monitored. In this case, Blockchain is an accounting ledger operating in the digital field. A special feature of blockchain is that transactions are executed at a high level of trust without disclosing information; security and reliability mechanisms are achieved through special mathematical functions or coding. By 2016, most Blockchain networks were used for digital money transactions. Smart contracts based on blockchain are being considered for a variety of different transaction types, from omnipresent devices to real-time operation management structures for industrial products and in-app data transmission including transaction finance. All types of business and management can participate in the network and use the properties of the blockchain system to ensure the transparency of stakeholders.

2) Consensus Protocol: allow parties in the network have permission to validating and maintaining the network. A transactions before submitted to ledger have to send to all node in network to ensure that the transaction is valid, so that transactions can't be tampered without parties's agreement. In addition, consensus protocol keep all node in the network synchronized with each other.

3) Smart Contract: is the source code stored within blockchain system and execute predetermined terms and conditions are met. Smart contract allow parties exchange goods, make payment, while avoiding the services of a trusted third parties. Basically, smart contract is developed by programmers and install by administrators. Smart contract of Hyperledger called chaincode written by golang, node.js or java language that interact to database in each block, functions in chaincode are called by client application.

4) Hyperledger Fabric: is an open source distributed ledger platform, design for developing permission application enterprise-grade, Fabric provide a platform to building fast, efficient and secure enterprise blockchain applications.

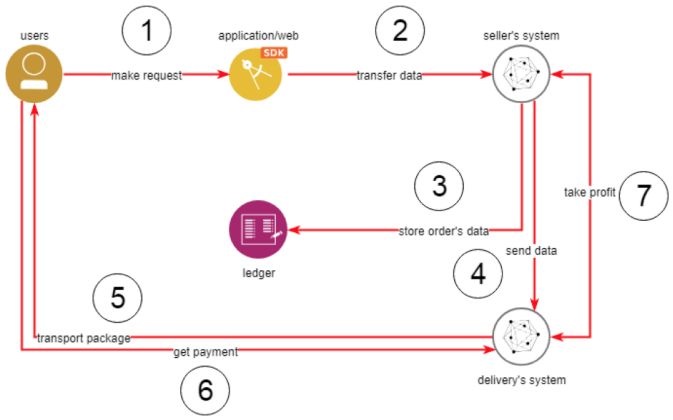

Fig. 1. The COD mechanism for protecting the seller's interest

B. General Model

COD system consist of some object: user (customer), application or website, seller, delivery, smart contract, order, distributed ledger is described in Fig. 1. In the first step, an order is made through website/application, order's information are send to seller's system and store to distributed ledger in step 2 and 3. These information is also send to delivery's system and stores again to the ledger in step fourth. After that delivery packages asset of seller and transport to customer in step 5. Next, the customer pay for asset and finally, seller and delivery take theirs profit in step 7.

IV. The COD Mechanism for Protecting the Seller's Interest

A. The COD Model for Protecting the Seller's Interest

In the implementation aspect, COD system contain customer (buyer), application or website, seller's system (seller), delivery system (delivery company), smart contract, order, distributed ledger. All object of COD system are described in Fig. 2. In this picture, the black lines denoted main steps; the green dashed lines describe the delivery companies can't transport package to customer in limit time; and the black dashed lines describe the customer refuse receive package. The customers make an order on website or application (step 1), order's information are send to the seller to confirm. The seller can accept or refuse this order (from step 2 to step 4). If sellers accept the order, a contract is created with a deal to determine the price of goods and the time, location delivery, step 5. After that, the seller finds the delivery company according to the delivery fee, the previous activities of the delivery company. The seller and delivery company will mortgage money including a delivery fee from the seller and the deposit from the delivery company (equal to the goods' value) to virtual account if they agree with the requirement in the contract including time, location, and fees of delivery. This money must be locked at the virtual account in the blockchain system after that the employee of the delivery company (shipper) transports the goods to the customer. Then the seller announces to the customer that the goods are being delivered (step 6 and 7). In the case of the goods are transported within the delivery time period, the locked mortgage money is refunded to the account of the delivery company. Delivery takes customer's payment and takes profit with seller, seller announce to the customer that the goods have delivered (between step 8 and step 13). If the customer refuses the goods, the shipper resends that goods back to the seller. Then the seller also announce to the customer (7.1 – 7.3). In case of shipper do not complete delivering within delivery time or there some trouble during delivering such as lost goods, damaged goods. The mortgage money is sent to seller's account and seller notify to the customer that the goods have been lost by the shipper from sub-step 6.1 – 6.3.

B. Algorithms

Algorithm 1 createOrder() function

```
if seller accept request then
    get information: id, customer's name, seller's name, asset, quantity, price
    store information to distributed ledger
else
    announce to customer that they cannot accept request
end if
```

We present a smart contract of COD model, this have some functions that interact to database blockchain to demonstrate the operation steps of the COD process. Firstly, customer make an order, order's information is send to seller system, seller store these information to distributed ledger (**Algorithm 1**), if seller don't accept order they must announce to customer. Seller choose which delivery company they want to transport the package, information of order is send to them, if delivery company accept the order they make a mortgage to virtual account, if not delivery transport package to customer, seller store order's information with delivering state, customer pay for asset (**Algorithm 2**). In case of customer refuse the package, deliverer transport back to seller delivery and seller take theirs profit, seller announce to customer that package have delivered (**Algorithm 3**).

Algorithm 2 delivering() function

```
get information of order
if delivery accept order then
    announces to seller that they accepts the order
    transfers money to virtual account
    transport package to customer
    if customer don't receive package within the limit time then
        money in virtual account transferred to seller's account
    end if
else if delivery refuse the order then
    announces to seller that they refuse the order
    choose other delivery
end if
```

C. Set Up Environment

In this section, we provide describe the environment used to implement the algorithm described above The system configuration for the experiments is a 64-bit machine with 8GB of RAM and 2.5 GHz Intel Core i5 CPU running Ubuntu

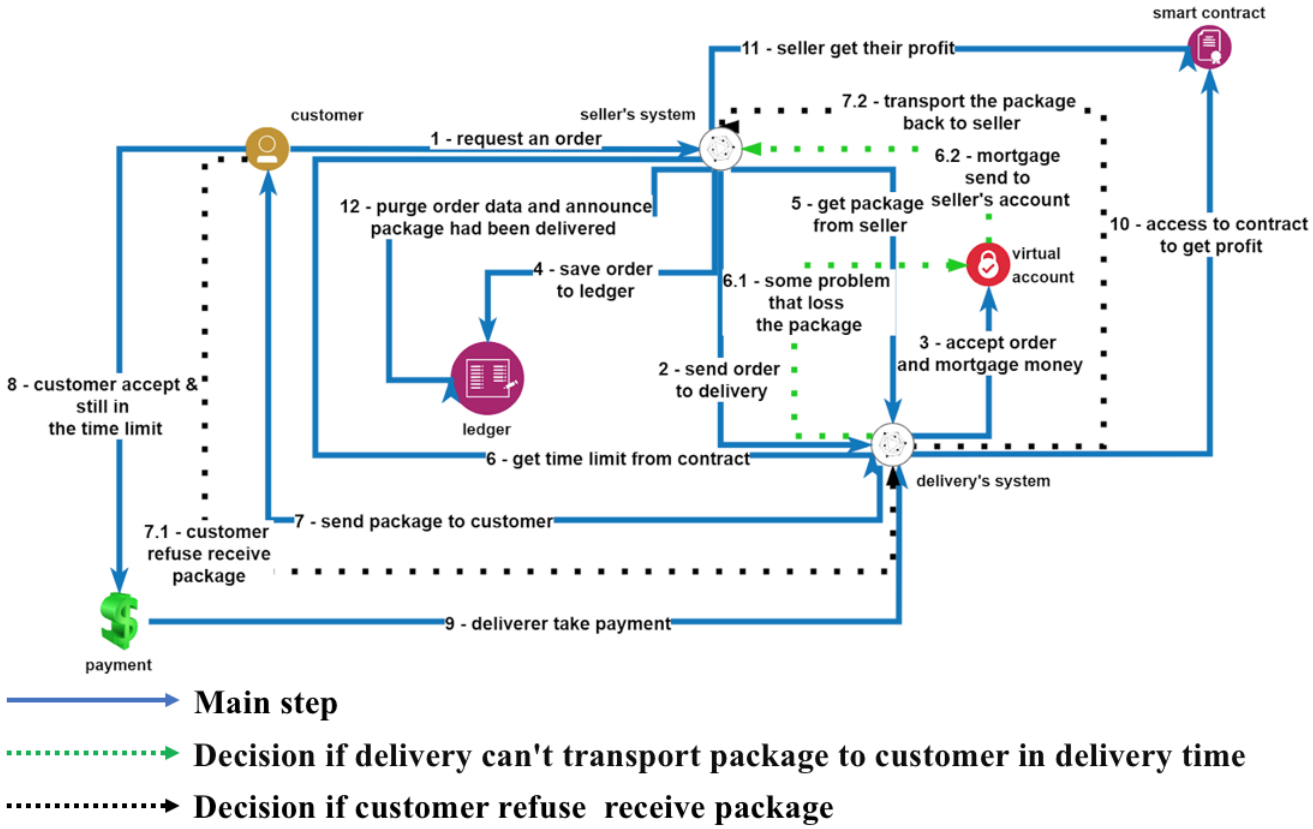

Fig. 2. The COD mechanism for protecting the seller's interest

Algorithm 3 takeProfit() function

```
if customer accept package then
    deliverer receive money
    delivery ad seller take theirs profit
else
    deliverer take package and give back to seller
end if
```

18.10 Cosmic Cuttlefish. We have implemented the DeM-CoD framework by using Hyperledger Fabric version 1.4[1] for deploying CoD process on the Blockchain and Golang for implementing CoD's businesses and cryptography functions[2]. The main steps of the COD system are described in detail below.

Open the terminal and go to the directory first-network, execute these command to start network:

```
./byfn.sh down
./byfn.sh up -c mychannel -s couchdb
```

after start the network it will create four peer, public key and private of each peer in two org:

```
peer0.org1.example.com
peer1.org1.example.com
peer0.org2.example.com
peer1.org2.example.com
```

We just need `peer0` of each `org` to represent seller and delivery of two organization. enter the CLI container.

```
docker exec -it cli bash
```

In default setting, after starting the `BYFN network`, the active peer is set to: `CORE_PEER_ADDRESS=peer0.org1.example.com:7051`, in detail it is use public key and private key created when you start network above, so you just install smart contract in `peer0` of `org1` to represent a seller company by execute the command:

```
peer chaincode install -n COD -v 1.0 -p
github.com/hyperledger/fabric-samples/
chaincode/COD/COD_chaincode/
```

Switching to `org2` to install smart contract, note that we just install to `peer0` of `org1` and `org2` because we use only one peer of each organization to represent seller and delivery:

```
export CORE_PEER_LOCALMSPID=Org2MSP
export PEER0_ORG2_CA=/opt/gopath/src/git
hub.com/hyperledger/fabric/peer/crypto/peer
Organizations/org2.example.com/peers/peer0.
org2.example.com/tls/ca.crt
export CORE_PEER_TLS_ROOTCERT_FILE=$PEER0
_ORG2_CA
```

[1]https://hyperledger–fabric.readthedocs.io/en/release–1.4/
[2]https://github.com/xuansonha17031991/CashOnDelevery-Chaincode

```
export CORE_PEER_MSPCONFIGPATH=/opt/gopa
th/src/github.com/hyperledger/fabric/peer/
crypto/peerOrganizations/org2.example.com/
users/Admin@org2.example.com/msp
```

Change the active peer to `peer0` in `org2` and install the chaincode:

```
export CORE_PEER_ADDRESS=peer0.org2.exam
ple.com:7051
```

```
peer chaincode install -n marblesp -v
1.0 -p github.com/hyperledger/fabric-sampl
es/chaincode/COD/COD_chaincode/
```

```
peer chaincode instantiate -o orderer
.example.com:7050 --tls --cafile $ORDERER
_CA -C mychannel -n COD -v 1.0 -c '{"Args"
:["Init"]}' -P "OR('Org1MSP.member', 'Org2
MSP.member')" --collections-config $GOPATH
/src/github.com/chaincode/COD/COD_chaincode
/collection.json
```

Now the smart contract have installed in two organization represent for `seller(org1)` and `delivery(org2)`, you can run the functions directly in the smart contract

D. Workflow

Step 1: Create prerequisites information. To participate in the network, companies need to initiates prerequisites information that is assets of sellers, accounts to make payments of seller and delivery, suppose that this information is ready in the network.

query seller's asset

```
{
  "docType":"Seller",
  "name":"seller_1",
  "asset":"Dell_inspiron_5000",
  "quantity":2,"price":1560
}
```

query seller's balance

```
{
  "balance":5000,
  "docType":"balance",
  "name":"seller_1_balance"
}
```

delivery's balance

```
{
  "balance":5000,
  "docType":"balance",
  "name":"delivery_1_balance"
}
```

Step 2: when customer make an order, customer's information and order's information is created customer's information

```
{
  "docType":"customer",
  "email":"someone@email.com",
  "location":"Vietnam",
  "name":"customer_1",
  "number":"01234567899"
}
```

order's information

```
{
  "docType":"Order",
  "asset":"Dell_inspiron_5000",
  "customer":"customer_1",
  "delivery":"delivery_1",
  "orderid":"order_1",
  "price":"1560",
  "quantity":2,
  "seller":"seller_1",
  "status":"waiting"
}
```

Step 3: after order accepted, seller contact to delivery company to make a deal with limit time. The limit time is set up at application layer so it isn't mentioned here, a virtual account is initialized if delivery company agree with that dealing. Finally, delivery make a mortgage to that virtual account, note that the mortgage must equal to asset's price.

query after create virtual account with balance 0

```
{
  "docType":"balance",
  "balance":0,
  "name":"virtual_account"
}
```

The balance properties of the virtual account is updated to 1560, equivalent to asset's price when delivery mortgage. Now, delivery transport the package to customer the status of package is changed to present that it is delivering.

query's result

```
{
  "docType":"Order",
  "asset":"Dell_inspiron_5000",
  "customer":"customer_1",
  "delivery":"delivery_1",
  "orderid":"order_1",
  "price":"1560",
  "quantity":2,
  "seller":"seller_1",
  "status":"delivering"
}
```

Step 4: Customer make a payment, delivery take money and receive theirs profit with seller. After that seller announce to customer to confirm that the package have delivered and money in virtual account automatically transfer back to delivery account. The status properties also change to `delivered` and store to ledger change status of the package the balance of virtual account is transferred to delivery, as mentioned above the rate is 3/7 so delivery take 30% of payment and seller

receive 70%, in detail the seller's balance receive $1092 (70% of $1560) and delivery have $468(30% of $1560). These rate depend on business mechanism of companies. In case of some problem that causes loss of goods or the delivery company does shipping the goods on time, the amount in the virtual account is automatically transferred to the seller's account.

Query's result after change status of the package to `delivered`


```
{
  "docType":"Order",
  "asset":"Dell_inspiron_5000",
  "customer":"customer_1",
  "delivery":"delivery_1",
  "orderid":"order_1",
  "price":"1560",
  "quantity":2,
  "seller":"seller_1",
  "status":"delivered"
}
```


E. Discussion

The current COD model still has limitations when a mechanism to protect sellers has not been offered, this paper points out those limitations and suggests an additional process to be implemented on smart contract. However, the objects and information is created is only objective reference and will be further studied to build a complete system. Hyperledger fabric is used to build peer to peer applications on blockchain networks and deploy, install smart contracts on that networks. In this paper we presents an additional process of COD model to solve the delivery side issues that directly affect the interests of seller and indirectly reduces the quality of services. Research results is COD process that has been implemented with smart contracts.

This paper presents an additional process of COD model to solve the delivery side issues that directly affect the interests of seller and indirectly reduces the quality of services. Research results is COD process that has been implemented with smart contracts. This is the basis for developing a complete system, from which businesses can consult to apply to their business models. However, the process is only implemented in the form of smart contract, there is not a complete system and has not yet developed the privacy protection feature.

V. Conclusion

In summary, this paper focuses on analyzing and outlining existing gaps within the COD process and thereby presenting solutions to address through blockchain technology. There is a worthy solution if they decide to apply COD model in theirs business this is the basis for developing a complete system, from which businesses can consult to apply to their business models.

However, the process is only implemented in the form of smart contract, there is not a complete system and has not yet developed the privacy protection feature, so we will continue to develop a complete system based on this smart contracts. In the future, we are confident that the number of companies applying COD model will be increased hence the need for blockchain applications in business has become more necessary than ever. Smart contract is the heart of the blockchain system and the infrastructure for businesses to develop their own blockchain-based system.

References

[1] "Cash on delivery the biggest obstacle to e-commerce in uae and region," 2014. [Online]. Available: https://www.thenational.ae/business/technology/cash-on-delivery-the-biggest-obstacle-to-e-commerce-in-uae-and-region-1.604383

[2] "The reason city link went bust on christmas day," 2014. [Online]. Available: https://www.forbes.com/sites/timworstall/2014/12/27/the-reason-city-link-went-bust-on-christmas-day/

[3] "The end of hanjin shipping - officially declared bankrupt," 2017. [Online]. Available: http://www.seatrade-maritime.com/news/asia/the-end-of-hanjin-shipping-officially-declared-bankrupt.html

[4] "Gnn: a prime example of cod payment risk," 2018. [Online]. Available: https://www.vir.com.vn/gnn-a-prime-example-of-cod-payment-risk-62270.html

[5] "Ethereum," 2018. [Online]. Available: https://ethereum.org/

[6] "Hyperledger fabric," 2019. [Online]. Available: https://hyperledger-fabric.readthedocs.io/en/release-1.4/

[7] X. S. Ha *et al.*, "Dem-cod: Novel access-control-based cash on delivery mechanism for decentralized marketplace."

[8] 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," 2018.

[9] 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.

[10] "Two party contracts," 2018. [Online]. Available: https://dappsforbeginners.wordpress.com/tutorials/two-party-contracts/

[11] "How our escrow smart contract works," 2018. [Online]. Available: https://dappsforbeginners.wordpress.com/tutorials/two-party-contracts/

[12] "Double deposit escrow," January 2018. [Online]. Available: https://bitbay.market/double-deposit-escrow

[13] J. Bremer and S. Lehnhoff, "Decentralized coalition formation with agent-based combinatorial heuristics," *ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal*, vol. 6, no. 3, pp. 29–44, 2017.

[14] "A peer-to-peer electronic cash system," 2008. [Online]. Available: https://bitcoin.org/bitcoin.pdf

[15] A. González, J. Ramos, J. F. De Paz, and J. M. Corchado, "Obtaining relevant genes by analysis of expression arrays with a multi-agent system," in *9th International Conference on Practical Applications of Computational Biology and Bioinformatics*. Springer, 2015, pp. 137–146.