논문 20-45-06-17 The Journal of Korean Institute of Communications and Information Sciences '20-06 Vol.45 No.06
https://doi.org/10.7840/kics.2020.45.6.1072

Black Money Usage Tracker Mechanism Based on Hyperledger Fabric

Farkhod Abdukodirov◆, Ki-Hyung Kim°

ABSTRACT

Black money and fake money are two major conundrums to the development of the third world and particularly less economically developed countries worldwide.

In the paper, we propose the mechanism which records money exchange information including the particular money with serial number, date, owner on the ledger and informs about black money when serial number does not match with the Public Ledger information.

Key Words : Hyperledger Fabric, Chaincode, Golang, Node.js, Web API, Smart Contract, Blockchain, Transaction, QR code

Ⅰ. Introduction

Major problems which are need to be addressed in less economically developed countries in the Global Village are black money and counterfeit of the money. This sort of issues cause more money to be circulated in the economy of the nation directing to a general increase in prices, inflation and even variations in foreign currency exchange rates. People are given top priority at first defense, if people have secure method to check their money, then huge amount of counterfeiting could be detected and prevented, what is more government could also control the foreign currency exchange rate compared to black market price.

In this paper we propose a mechanism which creates the Money exchange transactions Ledger and keeps the ledger run in the Hyperledger network, as well as the mechanism enables the users to check for the black money through the money's serial number. In fact, Hyperledger helps to make record of money exchange information including its serial number, date, owner information on the ledger, then one it is created, it can only be changed by the admin, in our case it would be the authority (e.g. Central bank). We run sample Hyperledger network with real demonstrative web application, which shows all transactions on the ledger. Ledger includes bank information, ID number of a particular transaction packet, date of when it was added to the ledger, current holder and serial number of particular money. Application also enables to call particular transaction individually from out of many transactions on the ledger. More importantly, we can also make new transaction on the ledger and assign the new owner to the existing packet. And last but not least important, we can check for the Black money by calling the specific transaction number, if the serial number of the money is not listed in the ledger, then Hyperledger

※ This research was supported by "Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education, Science and Technology(2018R1D1A1B07048697)" and Korea Institute for Advancement of Technology(KIAT) grant funded by the Korea Government(MOTIE) (P0008703, The Competency Development Program for Industry Specialist).

◆ First Author : Computer Engineering department, Ajou University, Ph.D, af4092@ajou.ac.kr, 학생(박사), 학생회원
° Corresponding Author : Department of Knowledge Information Engineering, Ajou University, kkim86@ajou.ac.kr, 정교수, 종신회원
논문번호 : 201912-339-C-RN, Received December 18, 2019; Revised February 13, 2020; Accepted March 31, 2020

network informs about the Black money, so that it could be stopped from being used in the bank transactions.

The reason that we have used Hyperledger Fabric because it has several useful criterias.[3] In a *public* or *permissionless* blockchain anyone can participate without a specific identity. Public blockchains typically involve a native cryptocurrency and often use consensus based on "proof of work" (PoW) and economic incentives. *Permissioned* blockchains, on the other hand, run a blockchain among a set of known, identified participants. A permissioned blockchain provides a way to secure the interactions among of a group of entities that have a common goal but which do not fully trust each other, such as businesses that exchange funds, goods or information. By relying on the identities of the peers, a permissioned blockchain can use traditional Byzantine Fault Tolerant (BFT) consensus.

Another reason is the *endorsement policy* that is evaluated in the *validation phase*.[3] Endorsement policies cannot be chosen or modified by untrusted application developers; they are part of the system. An endorsement policy acts as a static library for transaction validation in Fabric, which can merely be parameterized by the chaincode. Only designated *administrators* may run system management functions and have the right to modify the endorsement policy.

One of the goal of the Permissioned blockchain is to provide *neutral*, and community-driven infrastructure supported by technical and business governance. Hyperledger fabric also uses the Smart contract which are called chaincode in the system, by means chaincodes of the Hyperledger Network can also be implemented in the Ethereum Network.

Technical summary

	Hyperledger Fabric	Ethereum	Private Ethereum
Type	Private *(only the persons // entities selected can join the blockchain)*	Public *(everyone can subscribe and read the blockchain)*	Private
Permission	Permissioned *(administrators can decide who can read and write, and which data of the blockchain)*	Permissionless *(everyone can make new transaction and manipulate data)*	Permissionless
Speed	Very fast	Fast	Very fast
Scalability*	Fast	Medium	Okay

Unlike Ethereum, Fabric doesn't require a built-in cryptocurrency, but in fact it's possible to develop a native currency or a digital token with chaincode.

The rest of this paper is organized as follows. In Section II, related works are presented and discussed. Section III, gives background information on distributed ledger, chaincode, consensus, blockchain, hyperledger fabric. In Section IV, Proposed Mechanism is given. Experimental results are given in Section V. Section VI analysis the security related issues. And finally, Section VII Concludes the topic and gives the Future Research Scope.

Ⅱ. Related Works

Considerable work has been done in the area of Hyperledger Fabric and its implementation in various cases. In [1] highlights Black money and fake currency as main factors which destroy the economy of a country. As a solution[1] authors made a QR based technology to track the black money and fake currency, and also implementing cashless transaction using card. Evaluation on adaptability risk in money laundering using Bitmap Index-based Decision Tree (BIDT) technique was proposed[14] in the paper.[14] Bitmap indexing method efficiently categorizes the rows and columns based on the account details of the customer that reduces the risk identification time and greatly improves the adaptability rate. The paper[3] describes Fabric, its architecture, the rational behind various design decisions, its security model and guarantees, its most prominent implementation aspects, as well as its distributed application programming model.[3] further evaluated Fabric by implementing and benchmarking a Bitcoin-inspired digital currency. Video surveillance system based on blockchain system was proposed in [10]. Authors[10] enlightens, the metadata of the video is recorded on the distributed ledger of the blockchain, thereby blocking the possibility of forgery of the data, their proposed architecture encrypts and stores the video, creates a license within the blockchain, exports the video. A.C.Arize

and S.S.Shwiff in [11] studied the hypothesis that it is the black market exchange rate, not the official rate, that should enter into the demand for money function of countries where there is a black market for foreign currencies. [11] using several co-integration methods and Hausman test, they have shown that this hypothesis is strongly supported for most of the countries. In [2] authors proposed Medical chain storage using permissioned blockchain and how counterfeit drugs will be tracked.[2] Each participant will share their public key, hash value of previous transaction, encrypted QR code by manufacturer, QR code consist the details of medicine which is manufactured by pharmaceuticals agency. Paper proposed[5] an operations execution method for permissioned Blockchain systems, a primary idea is to define operations as System chaincode so that unified and synchronized cross-organizational operations can be executed effectively by using Blockchain native features.

We claim that the use of Hyperledger Fabric to record money and exchange transactions on the ledger enables to prevent money counterfeit and black money effect on foreign currency exchange rate. And tracks the money transaction details, and in case of fake money usage in the network, automatically informs the system, and blocks it from the ledger.

Ⅲ. Background

3.1 Distributed Ledger

The heart of a blockchain network is a distributed ledger that records all the transactions that take place on the network.[3] A blockchain ledger is often described as decentralized because it is replicated across many network participants, each of whom collaborates in its maintenance.[3] We will see that decentralization and collaboration are powerful attributes that mirror the way businesses exchange goods and services in the real world.

In fact, the information recorded to a blockchain is append-only, using cryptographic techniques that guarantee that once a transaction has been added to the ledger it cannot be modified[3]. This property of "immutability" makes it simple to determine the provenance of information because participants can be sure that information has not been changed after the fact.

3.2 Smart Contract(Chaincode)

To support the update of information consistently(Figure.1) – and to enable a whole host of ledger functions (transacting, querying, and so on) a blockchain network uses smart contracts to provide controlled access to the ledger[15].

Smart contracts are not only a key mechanism for encapsulating information and keeping it simple across the network, they can also be written to allow participants to execute certain aspects of transactions automatically. In fact, many smart contracts run concurrently in the network and they may be deployed dynamically in many cases by anyone. Fabric one of the better performing platforms[5] available today both in terms of transaction processing and transaction confirmation latency, and it enables privacy and confidentiality of transactions and the smart contracts what Fabric calls "chaincode" that implement them[3]. The chaincode is the central part of a distributed application in Fabric and may be written by an untrusted developer[3]. Special chaincodes exist for managing the blockchain system and maintaining parameters, collectively called system chaincodes. Chaincode is executed within an environment that is loosely coupled with the rest of the peer and that supports plugins for adding new languages for programming chaincodes[15].

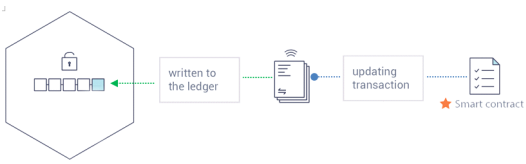

Fig. 1. Smart Contract

3.3 Consensus

The process of keeping the ledger transactions synchronized across the network – to ensure that

ledgers update only when transactions are approved by the appropriate participants, and that when ledgers do update, they update with the same transactions in the same order – is called consensus. Smart-contract existing blockchain platforms, follow an order-execute architecture in which the consensus protocol validates and orders transactions, then, propagates them to all peer nodes, each peer then executes the transactions sequentially[3]. The ordering of transactions is delegated to a modular component for consensus that is logically decoupled from the peers that execute transactions and maintain the ledger[7]. Since consensus is modular, its implementation can be tailored to the trust assumption of a particular deployment or solution.

3.4 Blockchain

Blockchain is an immutable transaction ledger, maintained within a distributed network of peer nodes. These nodes each maintain a copy of the ledger by applying transactions that have been validated by a consensus protocol, grouped into blocks that include a hash that bind each block to the preceding block. Blockchain is a shared,[10] tamper-proof replicated ledger in which records are made irreversible and nonrepudiable thanks to one-way cryptographic hash functions.

3.5 Hyperledger Fabric

One key point of differentiation is that Hyperledger was established under the Linux Foundation, which itself has a long and very successful history of nurturing open source projects under open governance that grow strong sustaining communities and thriving ecosystems.

Nodes are logical entities running on a physical server that can be maintained by participants.[18] They can be categorized into client, peer, and orderer nodes. Client nodes invoke transactions (Figure.2) and are connected to both peers and orderers. Peer nodes maintain the ledger and receive state updates in the form of blocks. They can also act as endorsers for verifying and validating a requested transaction. A transaction is

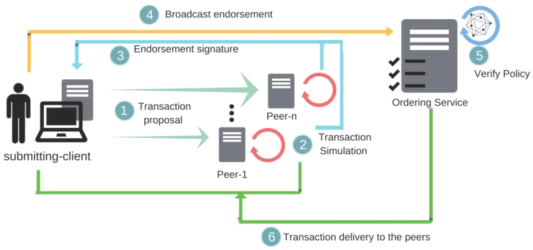

Fig. 2. Outline of transaction flow in Hyperledger Fabric

approved only when it acquires endorsement signatures from designated endorsers. Orderers support communication between clients and peers. When a client invokes a transaction request, the message is broadcasted to all peers. On receiving signature from all the endorsers, the orderer broadcasts a message to all peers to update their copy of ledger.

Ⅳ. Proposed Mechanism

We proposed the mechanism (Figure. 27) which detects the counterfeit money in the network and informs participants about its existence. The newness of the idea relies on the usage of Permissioned Blockchain particularly ***Hyperledger Fabric*** which simultaneously makes it different from other similar works where Serial number was used to detect the Fake Money. And our specific contribution in this paper is implement the Counterfeit Money detection mechanism with its Serial Number key point on Hyperledger Fabric Network. In fact, Fabric network is permissioned, meaning that, unlike with a public permissionless network, participants are known to each other, rather than anonymous and therefore fully untrusted. In our system participants are familiar with each other, that is way it is fully reliable. Figure.27 illustrates the process of proposed mechanism where User1 can get the Ledger from the Hyperledger Fabric Network by querying, and User2 can add new transaction to the Ledger with recordMoney function, money holder can change the owner of particular transaction with changeMoney function and last but not least User3

can check the money for its counterfeit by entering its serial number with checkMoney function. checkMoney function iterates and compares all valid transactions in the Fabric network, if it finds the serial number valid then it informs the User3 about its reliability otherwise informs about its fakeness. We wrote application and chaincode which interacts with Hyperledger Fabric network. More clear explanation of our contribution would be enlightened in the following paragraph "V. Experimental Results".

In Figure 3 we can see the members of the network: Central bank (authority) plays the role of the government which has the main power. Local banks bank(1), bank(2), bank(n) are controlled by the Central bank, and they interact with the Hyperledger Fabric network with the help of WebAPI. Other participants are user and money holder, when new user is submitted, its transaction details will be listed in the Ledger, later we can change the particular money holder to another one.

Our Fabric network can leverage consensus protocols that do not require a native cryptocurrency to incent costly mining or to fuel smart contract execution. Avoidance of a cryptocurrency reduces some significant risk (or) attack vectors, and absence of cryptographic mining operations means that the platform can be deployed with approximately the same operational price as any other distributed system. One of the flexible sides of Hyperledger Fabric is that it can be configured in multiple ways to suffice the diverse solution requirements.

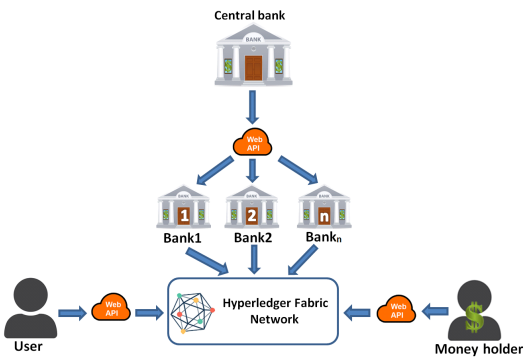

Fig. 3. Members of the network

V. Experimental Results

5.1 Setting up the Development Environment

Before starting the demonstration we need to install required prerequisites on the platform in which we are going to develop fabric network. In contrast, Hyperledger Fabric offers several SDKs to support different programming languages. Additionally, there are three more SDKs that have not yet been officially released (for Python, Go and REST).

Hyperledger Fabric also supports a certificate authority service on choice that, you may pick up to generate certificates and key material to configure and manage identity in blockchain network.

Our PC working development environment:

- OS: Windows 10 Enterprise
- Processor: Intel Core i7 @ 3.40 GHz
- RAM: 10 GB
- System type: 64-bit Operating System, x64-based processor

Installed Required Programs:

- Docker-version 18.09.2
- Docker-compose-version 1.23.2
- Curl-version 7.64.0
- Git Bash-version 2.21.0.windows.1
- Golang-version go1.12.5 windows/amd64
- Node.js-version v10.16.0
- Hyperledger Fabric Go SDK
- Java SDK
- Python-version 2.7.15
- Visual Studio editor

We have to note that, all programs stated above, should be in new updated version, otherwise there might be technical errors while running up the network. Chaincodes are mainly developed in Golang, but there are also other options to develop chaincodes in Java. In our test, we used Go for writing chaincode, for other source codes we have used JavaScript.

5.2 Network setup and web application run

Firstly, we will bring up our Hyperledger network, for this we have to be in money-blockchain/money-app directory.

As we can see from the Figure.4 we call script file, which brings the network up by installing composer module and fabric framework of Hyperledger network. It creates orderer.example, couchdb, peer0.org1.example. When it runs successfully it says “done”, otherwise it would have shown the error message.

Next comes, admin registration to our Hyperledger network (Figure.5). As we can clearly see on the console log: Successfully enrolled admin user “admin”, and assigns the admin user to the fabric client.

Then comes user registration to the fabric network (Figure.6). On our terminal, we can see that, it successfully registered user1 and enrolled member user “user1”. And now admin and user are ready to interact with our fabric network.

Meanwhile, I should note that, all these commands have been called with node.js.

Finally we run the app by bringing the server to the live port. Figure.7 shows, now we can open

```
Franky@DESKTOP-CCGTGOC MINGW64 /c/GoProjects/src/money-blockchain(v.2)(new)/money-app (master
)
$ ./startFabric.sh

export MSYS_NO_PATHCONV=1

docker-compose -f docker-compose.yml down
Removing network net_basic
Network net_basic not found.

docker-compose -f docker-compose.yml up -d ca.example.com orderer.example.com peer0.org1.exam
ple.com couchdb
Creating network "net_basic" with the default driver
Creating orderer.example.com ... done
Creating couchdb             ... done
Creating ca.example.com      ... done
Creating peer0.org1.example.com ... done
```

Fig. 4. Running up the Hyperledger network

```
Franky@DESKTOP-CCGTGOC MINGW64 /c/GoProjects/src/money-blockchain(v.2)(new)/money-app (master
)
$ node registerAdmin.js
 store path:C:\Users\Franky\.hfc-key-store
Successfully enrolled admin user "admin"
Assigned the admin user to the fabric client ::{"name":"admin","mspid":"Org1MSP","roles":null
,"affiliation":"","enrollmentSecret":"","enrollment":{"signingIdentity":"51f40f40a0ffffd401d1
8343f4fbbf90c475375970d827b794a5f1a3d4d8f48c","identity":{"certificate":"-----BEGIN CERTIFICA
TE-----\nMIICATCCAaigAwIBAgIU3QMui1Xc8o34pQ8uW7CmNvNqnakwCgYIKoZIzj0EAwIwXnczELMAkGA1UEBhMCVV
MxEzARBgNVBAgTCkNhbGlmb3JuaWExFjAUBgNVBAcTDVNhbiBGcmFuY2lzY28xGTAXBgNVBAoTEG9yZzEuZXhhbXBsZ
S5jb20xHDAaBgNVBAMTE1nE2NhLm9yZzEuZXhhbXBsZS5jb20wHhcNMjAwMjEzMDIzNTAwWhcNMjEwMjEyMDI0\nMDAwWj
AhMQ8wDQYDVQQLEwZjbGllbnQxDjAMBgNVBAMTBWFkbwlwMFkwEwYHKoZI\nzj0CAQYIKoZIzj0DAQcDQgAE+QRs+jwO/
fOkC2AIXDFmI2/bkYCyzqFIRqwoeAXl\nOhdMxCJrlSwoZilhrqEUjqqcpG9LJOdaaPdpIxlDkxv3K6NsMGowDgYDVR0P
AQH/\nBAQDAgeAMAwGA1UdEwEB/wQCMAAwHQYDVR0OBBYEFM4rTYRcdyehplw74v34FyTQ\n4V37MCsGA1UdIwQkMCKAI
EI5qg3NdtruuLoM2nAVUdFFBNMarRst3dusalc2Xkl8\nMAoGCCqGSM49BAMCA0cAMEQCICM4UbQ6Jopuf1sNO/ILuJxS
GehBIOUlOU7RgID\norPIA1AlIQTOT5uw7kJGE+uGnIlPRXYvaPDMpcv/2E6F4MhvNQ==\n-----END CERTIFICATE-
-----\n"}}}
```

Fig. 5. Admin registration to the network

```
Franky@DESKTOP-CCGTGOC MINGW64 /c/GoProjects/src/money-blockchain(v.2)(new)/money-app (master
)
$ node registerUser.js
 store path:C:\Users\Franky\.hfc-key-store
Successfully loaded admin from persistence
Successfully registered user1 - secret:uWPyorzZYUfr
Successfully enrolled member user "user1"
User1 was successfully registered and enrolled and is ready to intreact with the fabric netwo
rk
```

Fig. 6. Register user to the fabric network

```
Franky@DESKTOP-CCGTGOC MINGW64 /c/GoProjects/src/money-blockchain(v.2)(new)/money-app (master
)
$ node server.js
Live on port: 2206
```

Fig. 7. Running the server

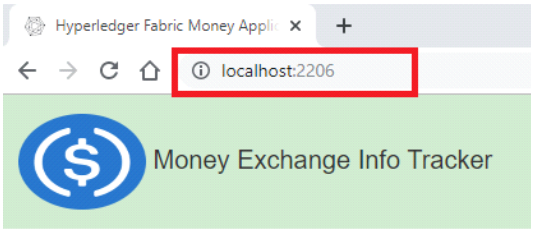

Fig. 8. Web API on live port 2206

Web API in any browser with localhost: 2206 port.

Figure.8 shows, our application is running on web browser localhost:2206 port. It also shows the sample name of our app which is money exchange info tracker, as well as on the right side there is introductory explains what does the web application implements. In the following lines, now we can test our application by interacting with Fabric network.

The application’s database (Figure. 9) is populated with some example money transaction packets. In our case, our ledger includes parameters like: Bank names, ID of a particular transaction packet, date of when it was added to the ledger, current holder and serial number of particular money. To get our ledger, we just need to press the button “Query”, then, it uploads the ledger.

We can also see the same result on terminal (Figure. 10), which gets all money from database,

Query All Money Packets

Query

Bank	ID	Date	Holder	Serial
AsakaBank	1	17052018	Mr.Farkhod1	1235AF545
NationalBank	2	15062019	Mr.Farkhod2	254845AS
WorldBank	3	02042018	Mr.Farkhod3	545A54515
KhalkBank	4	03062019	Mr.Farkhod4	TF5541564
Infinbank	5	12062019	Mr.Farkhod5	54545F515
TrustBank	6	21062018	Mr.Farkhod6	5487813RJ45
TuronBank	7	22062019	Mr.Farkhod7	F1564875468
HamkorBank	8	25072019	Mr.Farkhod8	784A54545A
CapitalBank	9	30042019	Mr.Farkhod9	1213448T45
AlokaBank	10	02052019	Mr.Farkhod10	54512854AS54

Fig. 9. Query all money packets of the given ledger

Fig. 10. Query ledger result on the terminal

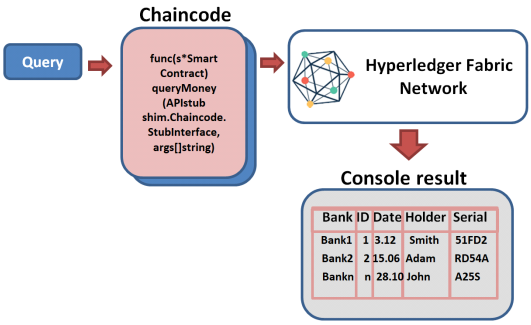

Fig. 11. Query process, how to query the public ledger

completes the query then checks for result, ultimately gives the response of the ledger.

Figure 11 describes the "Query" function of the Hyperledger network. When we query it calls the queryMoney chaincode which takes the all money transaction details from the Hyperledger network, then returns on the console.

Now let's query single packet with its ID which brings the particular packet out of ledger. We just simply need to submit packet number, as an example (Figure.12) we call ID 10 as a result we get all transaction information of 10th packet. Same result can be also viewed in terminal (Figure.13).

And now going on to the next function where

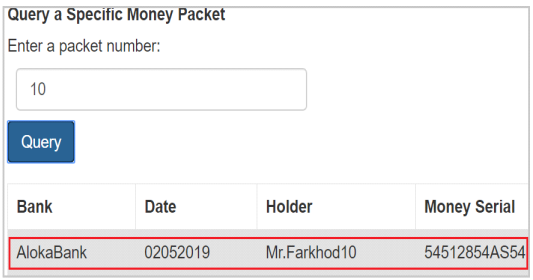

Fig. 12. Query a specific money packet on web

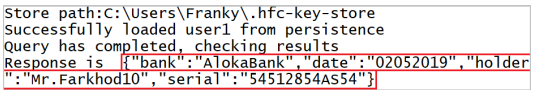

Fig. 13. Specific transaction information on terminal

Fig. 14. Create new packet to the ledger

we create new packet to the registry (Figure.14).

Create money record function enables us to create new packet and adds it to the running network automatically. In our test, we give 11 as a catch id, name of bank is AjouSampleBank, serial number 54515SF54515, date of registration to the ledger 20062019 and name of holder Farkhod, then press the Create button, after that it uploads the information to the ledger.

In Figure.15 network generates a unique transaction ID for this event on the terminal. When the transaction proposal is good, then it sends proposal to the network. Then after getting the Proposal Response, new record appears on the ledger (Figure.16) of web browser and on the

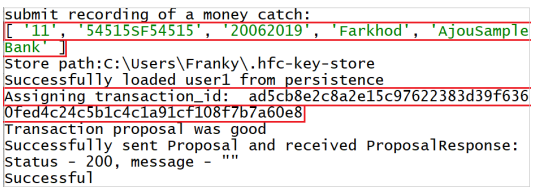

Fig. 15. Generate unique ID transaction

```
Query has completed, checking results
Response is  [{"Key":"1", "Record":{"bank":"AsakaBank","date":"1
7052018","holder":"Mr.Farkhod1","serial":"1235AF545"}},{"Key":"1
0", "Record":{"bank":"AlokaBank","date":"02052019","holder":"Mr.
Farkhod10","serial":"54512854AS54"}},{"Key":"11", "Record":{"ban
k":"AjouSampleBank","date":"20062019","holder":"Farkhod","serial
":"54515SF54515"}},{"Key":"2", "Record":{"bank":"NationalBank","
date":"15062019","holder":"Mr.Farkhod2","serial":"254845AS"}},{"
Key":"3", "Record":{"bank":"WorldBank","date":"02042018","holder
":"Mr.Farkhod3","serial":"545A54515"}},{"Key":"4", "Record":{"ba
nk":"KhalkBank","date":"03062019","holder":"Mr.Farkhod4","serial
":"TF5541564"}},{"Key":"5", "Record":{"bank":"Infinbank","date":
"12062019","holder":"Mr.Farkhod5","serial":"54545F515"}},{"Key":
"6", "Record":{"bank":"TrustBank","date":"21062018","holder":"Mr
.Farkhod6","serial":"5487813RJ45"}},{"Key":"7", "Record":{"bank"
```


Fig. 16. New record is added to the ledger on terminal environment

Query All Money Packets

Query

Bank	ID	Date	Holder	Serial
AsakaBank	1	17052018	Mr.Farkhod1	1235AF545
NationalBank	2	15062019	Mr.Farkhod2	254845AS
WorldBank	3	02042018	Mr.Farkhod3	545A54515
KhalkBank	4	03062019	Mr.Farkhod4	TF5541564
Infinbank	5	12062019	Mr.Farkhod5	54545F515
TrustBank	6	21062018	Mr.Farkhod6	5487813RJ45
TuronBank	7	22062019	Mr.Farkhod7	F1564875468
HamkorBank	8	25072019	Mr.Farkhod8	784A54545A
CapitalBank	9	30042019	Mr.Farkhod9	1213448T45
AlokaBank	10	02052019	Mr.Farkhod10	54512854AS54
AjouSampleBank	11	20062019	Farkhod	54515SF54515

Fig. 17. New packet appears on the ledger

Change Money Holder

Enter a catch id between 1 and 10:

11

Enter name of new holder:

Franky

Change

Fig. 18. Changing money holder in the ledger


```
Assigning transaction_id:  db9c9356d24c3e92f404587217f2cb7c2930f
17691af5968dcd8281def7af4d1
Transaction proposal was good
Successfully sent Proposal and received ProposalResponse: Status
 - 200, message - ""
Successful
```


Fig. 19. Assigning transaction id

terminal of Git bash (Figure.17).

Now, we will continue to interact with changing (Figure.18) the money holder (e.g. from ID-11 owner “Farkhod” to “Franky” and application assigns transaction id (Figure.19) for this event and sends the proposal to the network.

Query All Money Packets

Query

Bank	ID	Date	Holder	Serial
AsakaBank	1	17052018	Mr.Farkhod1	1235AF545
NationalBank	2	15062019	Mr.Farkhod2	254845AS
WorldBank	3	02042018	Mr.Farkhod3	545A54515
KhalkBank	4	03062019	Mr.Farkhod4	TF5541564
Infinbank	5	12062019	Mr.Farkhod5	54545F515
TrustBank	6	21062018	Mr.Farkhod6	5487813RJ45
TuronBank	7	22062019	Mr.Farkhod7	F1564875468
HamkorBank	8	25072019	Mr.Farkhod8	784A54545A
CapitalBank	9	30042019	Mr.Farkhod9	1213448T45
AlokaBank	10	02052019	Mr.Farkhod10	54512854AS54
AjouSampleBank	11	20062019	Franky	54515SF54515

Fig. 20. Query all money packets after updating the owner information


```
Successfully loaded user1 from persistence
Query has completed, checking results
Response is  [{"Key":"1", "Record":{"bank":"AsakaBank","date":"1
7052018","holder":"Mr.Farkhod1","serial":"1235AF545"}},{"Key":"1
0", "Record":{"bank":"AlokaBank","date":"02052019","holder":"Mr.
Farkhod10","serial":"54512854AS54"}},{"Key":"11", "Record":{"ban
k":"AjouSampleBank","date":"20062019","holder":"Franky","serial
":"54515SF54515"}},{"Key":"2", "Record":{"bank":"NationalBank","d
ate":"15062019","holder":"Mr.Farkhod2","serial":"254845AS"}},{"K
ey":"3", "Record":{"bank":"WorldBank","date":"02042018","holder"
:"Mr.Farkhod3","serial":"545A54515"}},{"Key":"4", "Record":{"ban
k":"KhalkBank","date":"03062019","holder":"Mr.Farkhod4","serial"
:"TF5541564"}},{"Key":"5", "Record":{"bank":"Infinbank","date":"
12062019","holder":"Mr.Farkhod5","serial":"54545F515"}},{"Key":"
6", "Record":{"bank":"TrustBank","date":"21062018","holder":"Mr.
Farkhod6","serial":"5487813RJ45"}},{"Key":"7", "Record":{"bank":
```


Fig. 21. Owner update to the newest one on the terminal ledger

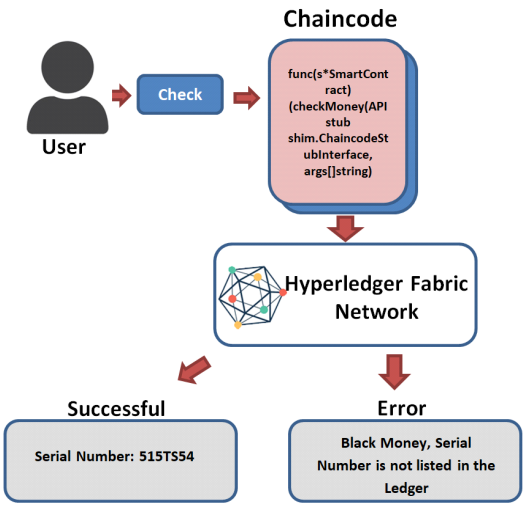

Fig. 22. Checking for the Black money

In the end the ledger now reflects the change for packet 11, and event will be broadcasted across the network (Figure.20) and the terminal (Figure.21).

Checking for the Black money (Figure.22) we

need the transaction number of the particular transaction on the ledger. New user initiates the Check function which runs the chaincode then requests the transaction validity from Hyperledger network, if it is Successful then Console prints out the serial number of the money, otherwise it prints Error message.

Ⅵ. Security Analysis

DoS(Denial of Service) attack is prevented in fabric. As its a private network mostly all participants are known and if someone is doing some malicious activity obviously others will know about it and can work to revoke the access of malicious nodes. Moreover, if there are 3 organizations participating and if the endorsement policy is set for approvals from 2 out of 3 then the 3rd org cannot affect the network. And side by side as everyone has the view of what's happening with the ledger, the maliciuos nodes can be kicked out of the network.

In fact, we can configure the Fabric network in a way that any transactions need the approval of all the participants which make 51\% attack impossible by design.

The proposed mechanism produces the secure channel for transactions on the ledger among different participants. Blockchain's distributed ledger is different with its decentralized records compared to other traditional databases. Data storage is approved with having no any failure point, as well as ledger synchronization is organized across the network. We believe in the decentralized blockchain to be secure. Our developed chaincode functions compile as a trusted distributed application and gets security from the blockchain and the underlying consensus among the peers.

On-chair privacy stipulates transactional privacy to be provided against the public (against no contract members) – unless the contractual parties themselves willingly disclose information. Chaincode implementations also rely on trusted servers for security.

Hyperledger Fabric is a permissioned blockchain with a membership infrastructure that enables participants of the network to not only strongly authenticate themselves in transactions but also to prove authorization to perform a variety of system

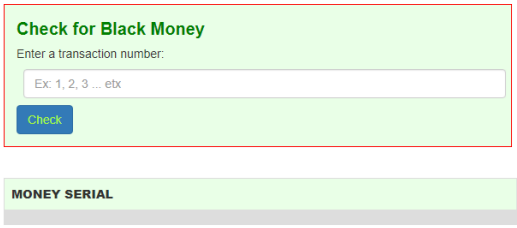

Fig. 23. Enter the transaction number

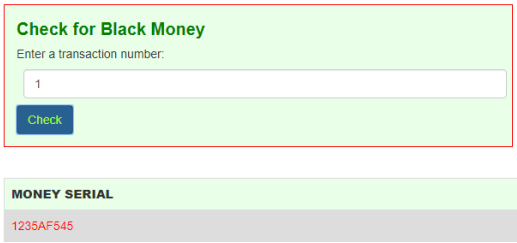

Fig. 24. In case of valid serial number of the money

```
Store path:C:\Users\Franky\.hfc-key-store
Successfully loaded user1 from persistence
Query has completed, checking results
Black Money, serial number is not listed in the ledger
```

Fig. 25. Serial number does not match with the ledger

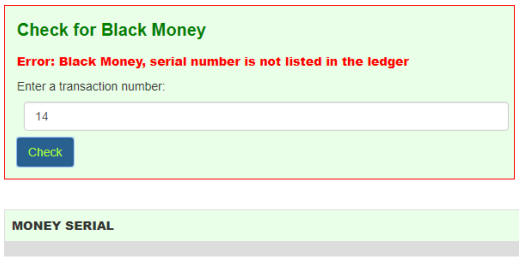

Fig. 26. Error message on the console

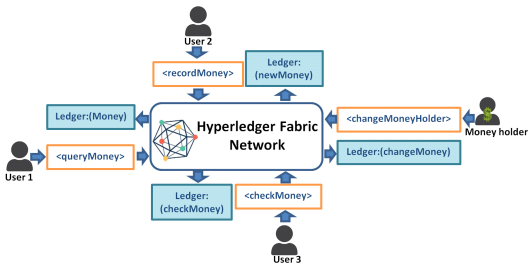

Fig. 27. Proposed mechanism

```go
type SmartContract struct {
}
type Money struct {
    Bank   string `json:"bank"`
    Date   string `json:"date"`
    Serial string `json:"serial"`
    Holder  string `json:"holder"`
}
func (s *SmartContract) Init(APIstub shim.ChaincodeStubInterface) sc.Response {
    return shim.Success(nil)
}
func (s *SmartContract) Invoke(APIstub shim.ChaincodeStubInterface) sc.Response {
    function, args := APIstub.GetFunctionAndParameters()
    if function == "queryMoney" {
        return s.queryMoney(APIstub, args)
    } else if function == "initLedger" {
        return s.initLedger(APIstub)
    } else if function == "recordMoney" {
        return s.recordMoney(APIstub, args)
    } else if function == "queryAllMoney" {
        return s.queryAllMoney(APIstub)
    } else if function == "changeMoneyHolder" {
        return s.changeMoneyHolder(APIstub, args)
    } else if function == "checkMoney" {
        return s.checkMoney(APIstub, args)
    }
    return shim.Error("Invalid Smart Contract function name.")
}
```

Fig. 28. Chaincodes of the network (“queryMoney”, “initLedger”, “recordMoney”, “queryAllMoney”, “changeMoneyHolder”, “checkMoney”)

checkMoney function

```
1:   procedure function checkMoney
2:         var serial=$scope.money_serial
3:         appFactory.checkMoney(serial, function(data))
4:     repeat $scope.query_money=data
5:         If ($scope.query_money==”Error msg”)
6:           console.log()
7:           $(“#error_check”).show() else
8:           $(“#error_check”).hide()
9:         end if
10:    until console.log(“success”)
11:  end procedure
```

Fig. 29. Pseudo code of checkMoney function app.js

operations, e.g. reconfiguration. Starting from its permissioned nature, Hyperledger Fabric offers a variety of confidentiality mechanisms to accommodate varying degrees of managing privacy, depending on the use case.

Ⅶ. Conclusion & Future Scope

This paper proposes the black money usage tracker mechanism based on Hyperledger Fabric network with demonstrative web application. Which in fact, keeps the record of all transactions in the fabric ledger, what is more enables the admin to make changes on the ledger by calling particular transaction packet, assigning new transaction to the ledger, and changing the owner of one specific packet and finally enables the user to check for the serial number of the money. In some cases it can also help to reduce the paper work in the governmental stage, by allowing them to keep data of records online.

Distributed ledger technology of Blockchain is entirely different from today’s database types as its records are decentralized. Which makes it impossible to have any point of failure for the data storage, moreover ledger is synchronized across the network. Additionally, we have also made repository in github.com (https://github.com/af4092/money-blockchain) as a useful resource. And our work undoubtedly is a development-ready application, which you just simply implement in the system and experience the outcome.

References

[1] J. Refonaam, G. G. Sebastian, D. Ramanan, and Dr. M. Lakshmi, “Effective identification of black money and fake currency using NFC, IoT and Android,” *IC3IoT*, Chennai, India, Feb. 2018.

[2] R. Kumar and R. Tripathi, “Traceability of counterfeit medicine supply chain through Blockchain,” *COMSNETS*, Bengaluru, India, 2019.

[3] E. Androulaki, A. Barger, V. Bortnikov, Ch. Cachin, K. Christidis, A. De. Caro, D. Enyeart, Ch. Ferris, G. Laventman, Y. Manevich, S. Muralidharan, Ch. Murthy, B. Nguyen, M. Sethi, G. Singh, K. Smith, A. Sorniotti, Ch. Stathakopoulou, M. Vukolic, Sh. W. Cocco, and J. Yellick, “Hyperledger fabric: A distrtibuted operating system for permissioned blockchains,” *EuroSys’18*, Apr. 2018.

[4] N. Zupan, K. Zhang, and H. A. Jacobsen, “Demo: HyperPubSub: a decentralized, permissioned, publish/subscribe service using blockchains,” *Middleware’17*, Dec. 2017.

[5] T. Sato and Y. Himura, “Smart-contract based system operations for permissioned blockchain,” *IEEE*, 2018.

[6] S. Nakamoto, “Bitcoin: A peer-to-peer

electronic cash system," https://bitcoin.org/bitcoin.pdf, 2008.

[7] K. Christidis and M. Devetsikiotis, "Blockchains and smart contracts for the internet of things," *IEEE*, May 2016.

[8] M. Conoscenti, A. Vetro, and J. C. De. Martin, "Peer to peer for privacy and decentralization in the internet of things," *2017 IEEE/ACM 39th ICSE-C*, Jul. 2017.

[9] W. S. Park, D. Y. Hwang, and K. H. Kim, "A TOTP-Based two factor authentication scheme for hyperledger fabric blockchain," *2018 ICUFN*, Aug. 2018.

[10] Y. Jeong, D. Y. Hwang, and K. H. Kim, "Blockchain-based management of video surveillance systems," *2019 ICOIN*, May 2019.

[11] A. C. Arize and S. S. Shwiff, "The black market exchange rate and demand for money in sixteen developing countries," *Int. Advances in Econ.*, May 1998.

[12] M. B. Oskooee and A. Tanku, "Black market exchange rate, currency substitution and the demand for money in LDCs," https://www.sciencedirect.com/journal/economic-systems/vol/30/issue/3, Oct. 2006.

[13] C. Hevia and J. P. Nicolini, "Monitoring money for price stability," *J. Econ. Dynamics and Control*, Apr. 2018.

[14] V. Jayasree and R. V. Siva Balan, "Money laundering regulatory risk evaluation using bitmap index-based decision tree," *J. Assoc. Arab Universities for Basic and Appl. Sci.*, Mar. 2016.

[15] T. Ahram, A. Sargolzaei, S. Sargolzaei, J. Daniels, and B. Amaba, "Blockchain technology innovations," *IEEE TEMSCON*, 2017.

[16] M. H. Eldefrawy and M. K. Khan, "Detecting counterfeit-money using RFID-enabled mobile devices," *ICITST*, 2012.

[17] S. Pongnumkul, Ch. Siripanpornchana, and S. Thajchayapong, "Performance analysis of private blockchain platforms in varying workloads," *IEEE*, 2017.

[18] O. Choudhury, H. Sarker, N. Rudolph, M. Foreman, N. Fay, M. Dhuliawala, I. Sylla, N. Fairoza, and A. K. Das, "Enforcing human subject regulations using blockchain and smart contracts," *Blockchain in Healthcare Today*, ISSN 2573-8240.

Farkhod Abdukodirov

Received Master's degree on Computer science at Ajou university, currently Ph.D student at Ajou university Computer engineering department. Research lab is Internet lab, main research areas are Blockchain, Hyperledger Fabric, Hyperledger Indy, DID.

Ki-Hyung Kim

Received the M.S and Ph.D degree in electronics engineering from KAIST, Korea. He has been working as a professor at Ajou university, Korea, since 2005. He has been Visiting Professor at Stony Brook University, NY, since 2011. His research interests include M2M, ISA100, 6LoWPAN, wireless sensor networks, blockchain tachnologies.