Money Transfer System Using Blockchain Technology: A Case Study of Banks in Iraq

Mohammed Sabri Deer\textsuperscript{1*}, Intisar Al-Mejibli\textsuperscript{2}, Alla Talal Yassin\textsuperscript{3}

\textsuperscript{1}Student MSc, Iraqi Commission for Computers and Informatics, Baghdad, Iraq
\textsuperscript{2}Assistant Professor, University of Information Technology and Communications, Baghdad, Iraq
\textsuperscript{3}Lecturer, University of Information Technology and Communications, Baghdad, Iraq

Email: \textsuperscript{1}pr.moh2002@gmail.com, \textsuperscript{2}dr.intisar.almejibli@gmail.com, \textsuperscript{3}dr.allatalal@uoitc.edu.iq

\* Corresponding Author

Abstract - Expansion of financial markets and diversity of banking activity increased a need for robust banking systems that can be used to provide the best service to client. However, the banking systems in Iraq still substandard. In the last few years, Blockchain technology has emerged as one of the most promising technologies that has attracted the attention of those interested in the banking industry around the world. Blockchain technology has the ability to eliminate the threat or the risk of fraud in all banking operation.

This paper briefly investigates the blockchain architecture, its types and some of the most famous platforms currently used. Additionally, it integrates the open source Quorum Blockchain with the banking procedures to propose an electronic funds transfer system, which secures and facilitates the banking operation in Iraqi banks.

The design and implementing of the proposed system depends on two smart contracts. The proposal presented the design of different algorithms that are executed in bank-end and client-end. Friendly and easy to use interface has been implemented to facilitate the use of the proposed system. Implementing the proposed system achieved promising results in term of high level of security, maintaining the client's privacy, reducing the required time to perform the banking service.

Keywords – Blockchain, e-transfer money, Asymmetric encryption, Hashing

I. INTRODUCTION

Digitalization in information technology (IT) and communication in recent decades has triggered important economic and social changes worldwide, which in turn become one of the most considerable source of development of banking industry [1]. However, the latest transformation in Iraq has led to a major expansion in financial markets and diversification of banking activities, thus the need for robust banking systems has increased.

The banking sector in Iraq currently consists of (63) banks; including almost (1068) branches [2]. These banks are associated with the Central Bank of Iraq (CBI), which takes over responsibility of currency issuance, retaining cash reserve, supervision and monitoring the banks, and etc. [2].

Despite many of these banks use various banking systems for management of client services, most of them still didn't get rid of dependence on papers, as well there is no direct connect among these systems. So, the traditional method of transferring money between clients in different banks is using checks and remittance forms which takes at least three work days to be achieved. In addition, it is susceptible to tampering and forging.

However, there are two payment systems used in Iraqi banks. The real time gross settlement (RTGS) designed for large value payments between banks; and the automated clearing-house system (ACH) where payments are much smaller amounts. Both systems are operated in servers of CBI that give permissions of use to all banks’ general administrations and branches. It should be noted there is no direct connection between the transferring systems on one side and the other banking systems used to manage clients' accounts and services on other side. Rather, data are entered or updated separately in each system. In other word, data are transferred offline between these two types of systems by users themselves according to authorization level granted to each user. Figure 1 shows a general structure of banking systems used in Iraqi banks.
Since 2009, when Nakamoto succeeded in constructing and using the first cryptocurrency based on the blockchain technology, it has become the buzzword [3], not just in financial area rather in many of different fields like smart contracts [4], Internet of Things [5] [6], security services [3] [7], health care [6] [8], energy [9] [10], etc. So, blockchain technology has emerged as one of the most promising technologies that has attracted the attention of those interested in the banking industry around the world [11] [12] [13] [14].

Security is a core issue in any financial system especially those which are implemented over network. The conventional method for funds transfer always being subject to frauds, hacking attempts and data tampering. In addition to, it consumes more time, effort and cost.

This paper addressed a solution for funds transfer system employing the Blockchain technology to present secured money transactions that clients can trust using in. In addition to, reduce the required time and effort in performing the banking services.

The rest of this paper is arranged as follows. Section II presents related researches and studies. Section III gives a preview of blockchain architecture. Section IV presents the proposed model. Section V discusses the model application and section VI concludes the paper.

II. LITERATURE REVIEW

Many of literatures like research papers, conference proceedings, journal articles, blogs and forum posts cover blockchain applications in various areas, including banking industry. He Sun et al. [15] proposed a digital currency model for central bank, which is constructed on the permissioned blockchain. Compared to [15], this paper interests in traditional currency transferring among banks instead of digital currency. Haneffa Muchlis Gazali et al. [16] proposed a prototype based on blockchain technology for managing study loans favor the concerned institutions. Compared to [16], this paper focuses on secure money transfer among various bank clients. Haiss Peter and Andreas Moser [17] presented an investigation which include a literature review and a survey among Austrian and German banks to utilize blockchain technology. One of conclusions is that most conventional foundations don't know much
about it until research date. The current study meet with [17] in that most of study sample don’t know about blockchain just the cryptocurrency, “Bitcoin”. Xin Wang, et al. [18] using Hyperledger Fabric blockchain platform, proposed a prototype of inter-bank RTGS system. Contrast to [18], this paper focused on ACH system using Quorum blockchain and smart contracts.

In the present work, a blockchain-based model is presented to eliminate inefficiencies in the money transfer process among Iraqi banks. The proposed model provides secure electronic money transfer and reduces the time of clearing to as little as possible.

III. BLOCKCHAIN ARCHITECTURE

A Blockchain is a sequence of data blocks which are validated, linked and replicated among all the parties (nodes) of a peer-to-peer (P2P) network in a secure and permanent way [3] [19]. Blockchain technology utilizes well-known cryptographic techniques such as asymmetric-key cryptography, hash functions and digital signatures mixed with some of record keeping concepts like append-only ledgers [3] [20]. A blockchain system generally consists of a number of nodes, each one has a local replica of a ledger. To coordinate and validate transactions of the ledger, these nodes do not require a central authority. Instead, they connect with each other in order to get agreement on the content of the ledger [21]. Figure 2 shows a structure of generic blockchain with its main components.

![Figure 2. Structure of Generic Blockchain](image)

In blockchain system, as a distributed environment, the major challenge is how multiple parties can reach to agreement on validating different transactions and blocks before storing them in a distributed ledger. Blockchain platforms use set of consensus protocols (pre-agreed rules) to implement blockchain operations [22] [23]. Many models in their original form or variations, were proposed to deal with consensus issue, like Proof-of-Work (PoW), Proof-of-Stake (PoS), Byzantine Fault Tolerance (BFT), Proof-of-Authority (POA) [3] [23] [24], and others. Blockchains can be classified into three types [3] [25]: public, private and consortium. In public blockchain, all transactions are apparent to the all users and they could participate in the consensus process. If only those who belong to one organization can take part the consensus process, the blockchain is called a private. While in a consortium, only a set of pre-defined nodes would participate the consensus process. From other side, blockchains can classified based on permission model, e.g., who can publish blocks. So, it can be regard public blockchain as “permissionless”, while consortium and private can called a “permissioned” [20] [26].

In the first blockchain (e.g., Bitcoin), the target was to build and use cryptocurrencies. Later with the advent of Ethereum, it becomes possible to use smart contracts to build decentralized applications (DApps) on blockchain networks. Smart contract is an agreement between two or more parties in the form of program code. It is written in high-level language (e.g., Python or Solidity) and recognized by a unique address assigned in the blockchain network. Smart contract includes a collection of state variables and functions, as main components. When some functions are executing, the status of the variables is modified according to the logic written on the contract. Smart contracts can be used for any type of transactions not just financial ones. Once meeting the conditions of the agreement, the contract code is automatically executed on each node in the network as a part of the verification of new blocks [4].

IV. PROPOSED SYSTEM

The proposed system based on using blockchain technology to secure the money transfer system and facilitate the most banking procedures. The proposal provides the clients with a service of secure online transferring and receiving money.
The concept of the proposed system consists of two stages. The first stage involves creating a financial transfer transaction and verifying it by the concerned parties (banks), as appropriate. So, the output of this phase will be either a validated transaction or a notification that the client request is rejected for any (identified) reason. The second stage involves placing all the transactions that have been validated through a specific period of time in a new block, approving this block, and appending it to each copy of the blockchain stored in the network nodes. This stage is completely based on consensus mechanisms that offered by blockchains.

The following subsections explain the system model components, request/transaction structure, system implementation, user interface that can used and finally present a description of the proposed system work using flowchart and number of algorithms.

4.1. System Components–

Figure 3 shows the components of the proposed system model.

The proposed model includes the next components:

1. Number of computers (network nodes) that can be classified into three types:
   - **Bank nodes** are the nodes of the banks (or branches) participating in the proposed system. In order to achieve the transparency and other features that the blockchain builds on, a copy of the blockchain should be maintained in each (or most) of these nodes.
   - **Block-maker nodes** that create and validate each block before they are added to blockchain according to the relied consensus protocol.
   - **End-points nodes** (users’ nodes) that have permit to access the system. Two types of users can interact with this model. First, the clients who have accounts in participating banks and want to use bank services (e.g. for money transfer to other clients inside the same network). Second, bank employees who are classified according to the authority granted them to access the system. Except familiar browsers, this type of nodes is not containing any programs specific to the proposed system. These nodes can be computers devices, smart phones, or etc. They connect to the system via the web server specific for each bank.

2. A peer-to-peer networking that links the nodes mention in point 1 (type A and B). It is based on permissioned quorum blockchain [26] [27] [28].

3. Private smart contract that implements the logic of the system at each bank. It stores the client's data and accounts, conducts the required validating and manages the transactions related to the these accounts. The proposed model uses bankAgent contract where a single instance is deployed at each bank node.

4. Public smart contract that stores the common data for all banks and declares the approved transactions after they are validated by the concerned banks. Some of shared data are banks' codes and contracts' addresses. This contract is also responsible of validating assurance ceiling of bank balances stored in CBI to perform any funds transfer between different banks.

5. A set of back-end programs and front-end (web-based) interfaces that provide pages required for using the system.

It is worth noting that the current model is not support account opening. Opening an account usually requires verification of some documents and information to comply with existing laws.
4.2. Requests/Transaction Structure –

For each participating bank on the network (except CB), there is an instance of the bankAgent contract is deployed. It has a number of functions to store new requests and check their validation according to concerned accounts. Data fields of each request are organized as shown in Table 1.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Id</td>
<td>Request Identity</td>
</tr>
<tr>
<td>Timestamp</td>
<td>Timestamp of the request</td>
</tr>
<tr>
<td>Status</td>
<td>Status of the request after latest process</td>
</tr>
<tr>
<td>Amount</td>
<td>The amount of money</td>
</tr>
<tr>
<td>Currency</td>
<td>The currency code</td>
</tr>
<tr>
<td>initContract</td>
<td>Originator contract’s address</td>
</tr>
<tr>
<td>initAccount</td>
<td>Originator client’s account</td>
</tr>
<tr>
<td>destContract</td>
<td>Beneficiary contract’s address</td>
</tr>
<tr>
<td>destAccount</td>
<td>Beneficiary client’s account</td>
</tr>
</tbody>
</table>

For CB, an instance of the cbAgent contract is deployed. After request validating in concerned banks, the hash value of this request will write as a valid transaction in cbAgent contract. Data fields of valid transaction is as shown in Table 2.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Timestamp</td>
<td>Timestamp of the transaction</td>
</tr>
<tr>
<td>Approved</td>
<td>Approving Status (true/false)</td>
</tr>
</tbody>
</table>

The data of each request/transaction is saved in a state variable of mapping type, which is a table of pairs (key-value). The transaction is accessed directly only via the key and the table cannot be read in sequential manner at all. The key of transaction is got by applying next the hash function:

keccak256(abi.encodePacked(initContract, Id));

The parameters are all send privately inside the request from originator contract to destination contract. Accordingly, no other party can access the transaction easily.

4.3. System Implementation –

The implementation of proposed model includes the following steps:

- Design and write smart contracts in Solidity language and compile each smart contract to generate its binary file. Remix-Ethereum IDE is the tool that used in the current work.
- Specify the number of nodes and prepare the network configuration. It is possible to use auxiliary software to provide a number of virtual nodes such as ganache-cli software.
- Install, build and configure a Quorum node on each machine of the network. Each machine (node) serves as a separate bank. RAFT is the consensus mechanism used in this model.
- Execute the code in previous step in network nodes as it needed.
- Deploy the contract on Quorum network using Quorum client (Geth).
- Run a decentralized app (DApp) composed of front-end and back-end applications (nodejs) that provide interaction with clients, banks and the smart contract.
4.4. User Interface –
The clients interact with the proposed system via simple web application which is programmed using HTML, JavaScript and Web3JS. It mainly includes three pages: login page, money transfer page and account statement page. In order to perform a transfer request, the client should follow these steps: (1) login to system, (2) navigate to money transfer page, (3) enter transfer information, (4) click on "Sign the Request" button to add the digital signature and (5) click on "Send Money" button. The design of money transfer page will take the shape depicted in figure 4.

![Money Transfer Page](image)

Figure 4. Front-end Application: Money Transfer Page

4.5. Description of system work –
The logic of the system implements through smart contracts including bankAgent that deployed at each bank node and cbAgent deployed at CB node. As well through RAFT consensus protocol which is installing by default with quorum blockchain.
The major operations in the system can be explained in more details by three algorithms. Each algorithm represents a part of system work implementing in bank node or CB node, based on role of each node.
Figure 5 shows a flowchart of the proposed system work.
Figure 5. Flowchart of the System Work.

In Algorithm (1), the client logs onto the system via front-end app, fills the transfer form, signs the request using the private key and submits it to concerned bank agent where the client's account is. It is worth mentioning that the client's key is generated when the client opens his account at specific bank.

**Algorithm (1): Client login and place a request for money transfer**

**Input:** Client’s user name, Client’s ID, password, and transfer information

**Output:** encrypted request

1) Begin

2) Enter client’s user name, id (user-id), and password

3) If the entered information are not corrected then Go to step 9

4) Place a request for money transfer. Fill the request form with: transfer amount, currency, client’s bank, client’s account number, beneficiary's bank and beneficiary's account number

5) The client should approve the request by signing it as follow:
a. Encrypt the request firstly using client’s private key
b. Encrypt the result from (a) with user-id using bank’s public key

8) Submit the encrypted transaction to bankAgent in the concerned bank
9) End

Each bank (or branch) node has an instance of bankAgent that receives the request Rx, determines its type (incoming, outgoing or inside same bank transfer), validates it related to current bank, and sends it to bankAgent in counterparty banks. Finally, when Rx is validated in (all) concerned bank(s), it is send to public contract. Algorithm(2) presents steps of work performed by bankAgent.

Algorithm (2) (bankAgent): Validate Request according to concerned bank (node)

Input: Encrypted Request(Rx), user-id
Output: Encrypted Request (Rx) with status “PASS1”, “PASS2”, “VALID” or “ERR-n” where “n” an error

1) Begin
2) Decrypt Rx using concerned bank’s private key
3) If user-id is recognized and originator’s bank matching current bank Then /* Means outgoing trans */
   Continue
   Else
   Go to Step 8
   End If
4) Decrypt Rx using originator’s public key based on user-id
5) Generate Request Id as a reference in current bank, Set timestamp
6) Validate request according to originator’s account, comparing with client account in current bank
7) If validated Then
   Reserve the amount from the active balance of the client account
   Set status as “PASS1”
   Else
   Set appropriate ERR-n
   Go to Step 10
   End If
8) If beneficiary’s bank matching current bank Then /* Means incoming trans */
   a. Validate request according to beneficiary’s account, comparing with client account in current bank
   b. If validated Then
      Set status as “PASS2”
      Else
      Set appropriate ERR-n
      Go to step 10
      End If
   End If
9) If originator’s bank matching beneficiary's bank Then /* Means in the same bank */
   Set status as “VALID”
   End If
10) If Rx status is “PASS1” Then
    Encrypt Rx using public key of beneficiary's bank
    Send to beneficiary's bank
    Else If Rx status is “PASS2” Then
    Encrypt Rx using public key of CB
    Send to cbAgent
    Else If Rx status is “VALID” Then
    Write Rx hash into cbAgent
    Else
    Return ERR-n to originator’s bank if it not the current bank
   End If
11) End
The cbAgent that deployed in CB node, validates Rx for clearing purpose between counterparties. Algorithm (3) illustrates main task in cbAgent.

**Algorithm (3) (cbAgent):** Validate transfer conditions in CB bank (node)

**Input:** Encrypted Request (Rx) with status “PASS2”

**Output:** Encrypted Request (Rx) with status “VALID” or “ERR-n” where “n” an error number

1) Begin
2) Decrypt Rx using CB bank’s private key
3) If both banks (originator and beneficiary) validate conditions and originator’s bank have an enough balance Then
   Reserve the amount from the active balance of the originator's bank
   Set Rx status as “VALID”
Else
   Set appropriate ERR-n
End If
4) End

The second stage of system work involves creating the blocks. These procedures has implemented based on RAFT consensus algorithm whose Quorum provides it by default. It is used for managing duplicated state logs or machines, and it is simple to comprehend and implement [28]. RAFT algorithm provides instant transactions finality, offers faster block times and does not produce unnecessary empty blocks, making them an appropriate option for permissioned blockchain implementations[28].

Transaction is published in bankAgent instances for both originator contract and beneficiary contract. It is stored as uncertain state (approved=false), and it is only final whenever be approved. The hash value of this transaction is published in cbAgent contract to ensure that all concerned parties are aware of the transaction. Transactions can be approved by calling the approveTransaction function to update the transaction state (approved=true). The only party authorized to invoke this function is bankAgent. Thus, the clients’ balances in concerned banks and the banks’ balances in CBI are updated as much as the amount reserved for each transfer operation (adding to the beneficiary balance and subtracting from the originator balance). Just now, the beneficiary client can receive the amount of transfer added to his account.

V. LIMITATIONS AND FUTURE CONSIDERATIONS

Many requirements must be satisfied to implement the proposed system on an actual network of banks. There is a need to induce number of banks for applying and testing this model. It should also be noted that the transfer service fees due to banks have not been taken into account, and it is possible to add them with the fields in transaction structure.

Blockchain convenes the promise to address the complicated issues plaguing the Iraqi banking system presently, that is, traditional way for funds transfer. Conceptually, the blockchain technology supplies a solution to this problem by offering set of features that attract those interested in the banking industry.

However, on a practical part, some challenges stick around. The adoption of blockchain requires a technological and regulatory evaluation, immense collaboration, increased societal awareness, stringent planning and intact technical expertise.

Banks deal with sensitive financial information and clients’ private information requiring rigid data protection standards. Iraqi banks are guided by the Central Bank of Iraq’s guidelines on funds trading and data protection. Information security consciousness is low in the Iraqi banking system, which exposes financial institutions to cybercrimes and data security threats.

In Iraqi banks, one of the most critical information security challenges is the threat to integrity and confidentiality of information when shared with external parties.

Blockchain could be applied as a new method for financial institutions and their clients to transfer the money with each other that does not depend on any third parties. By conducting transfers between banks themselves in addition their clients, banks would be able to save a major amount on costs as well improve the speed and safety of payments. Consequently, blockchain can be used to make transfers and payments in real-time execution, complete transparency, real-time fraud analysis and prevention at a reasonable cost.
IV. CONCLUSION

This paper proposes a model for e-transfer money between Iraqi banks based on the blockchain technology. The system reveals a high level of security applying key features of the blockchain which is designed to be immutable, tamper-proof and irreversible. To ensure that the data is not manipulated, the asymmetric key encryption is used to transfer transactions between the concerned banks, as well signing the transfer requests using digital signature. The system maintains the privacy of clients by separating between the client identity and his financial transactions. The transactions stored in the blockchain are associated to bank codes and account numbers. Thus, there is no indication to the client identity, which is known only at the branch where the account is opened.

Using this model, the transfer time was reduced from at least three work days in conventional banking systems to several seconds. This time represents the waiting time for the new block plus the time it takes to create, approve the new block and update accounts in banks’ contracts.

In addition, the system provides an easy and friendly interface where any client has the ability to transfer money or get an account statement, as long as he/she has an access to the system.

REFERENCES

[14] “Blockchain in banking While the interest is huge, challenges remain for large scale adoption Headline Open Sans Bold Subhead Open Sans Light up to two lines of text,” 2017.