IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.Doi Number

SOTF: Secure Organizational Transactions Framework Based on Bitcoin Payment Bridge

Shereen M. Mahgoub[1], I. I. Ibrahim[1], and Fatty M. Salem[1]
[1] Department of Electronics and Communications Engineering, Faculty of Engineering, Helwan University, Cairo, Egypt

Corresponding author: Fatty M. Salem (e-mail: faty_ahmed@h-eng.helwan.edu.eg).

ABSTRACT Bitcoin is a decentralized cryptocurrency where all the transactions are saved in a ledger. Blockchains have good features to be used in different finance applications by using smart contract. Also, blockchains are attractive platform for many industrial domains, including the logistics and supply chain industries. Supply chain technology contributes to record every single asset through its flow, tracking orders, receipts, and payments. There are several relay protocols proposed in previous research; these relay protocols are used to connect different types of blockchains. In this paper, we present a Secure Organizational Transactions Framework (SOTF) based on bridge chain to connect private chain or consortium with public chain which reduces interaction with public blockchain and saves the organization details privately in private chain or consortium. The proposed framework makes it possible for smart contract services to acquire means of payment in the consortium and private chain. Moreover, the proposed framework automates the process of updating the payment non-interactively. In this paper, we validate the communication between organizations blockchain and bitcoin, and find out the development cost of the proposed framework which ensures the efficiency and feasibility of the proposed framework. The code of framework implementation is publicly available at GitHub.

INDEX TERMS Bitcoin, Organizations, Blockchain, Consortium blockchain, Supply chain, Smart contract.

I. INTRODUCTION

Bitcoin is an online payment method proposed by Satoshi Nakamoto in 2008 [1] and published as software in 2009. A turnover of 500 transactions per second would require 10 TB of additional disk space per year, which is at the limit for a consumer. This public blockchain stores all transactions with average block size 1MB [2] contains about 2.000 transactions [3] and generates a block in about 10 minutes; this makes database grows about 50GB in a year [4].

Research starts using Off-blockchain transaction protocols which use cryptocurrency contracts to guarantee agreed upon action to be used in different applications as medical records [5], IoT platforms [6], online voting [7], energy trading [8] and transportation systems [9]. First off- blockchain is Duplex Micropayment Channels, proposed by Decker and Wattenhofer [10], which is bidirectional channel for transaction. The second one is called Lightning Channels (HTLCs), proposed by Poon and Dryja [11], which opens the channel indefinitely. HTLCs, are contracts that require the recipient to reveal a secret to gain an output before they are refunded to the sender using a single or a multi-hop scenario.

One of the powers of bitcoin is its stack-based scripting language which allows specifying how funds can be transferred by creating scripts. In bitcoin scripting language, Alice creates a transaction to pay to Bob that signs prior sending it to him. There are different types of blockchains [12, 13, 14, 15]:1) Public blockchain as bitcoin and Ethereum [16, 17, 18] where anyone can make a transaction and access the block, but the problem is transaction expansion and speed delay. 2) Private chain in which one management entity can control and have the authority for all the chain service.

Consortium is a semi-centralized blockchain controlled by selected nodes business agreement among entities as corda [19], Hyperledger [20], Ethereum Enterprise [21] Alliance and nexledger [22]. Consortium chain can focus on services operation rather than economic issues. Consortium chains are used when we need to share database with multiple participants based on predefined permission. Supply chain is a consortium example which is used for tracking products as farm, transformation, manufacturing, retailer, and distribution; all of these can be tracked in the chain [23, 24, 25, 26, 27, 28].

Supply chain is fundamental for gaining financial, environmental and social benefits, but it needs good management. The authors in [29] have proposed a smart contract design based on ontologies of an incepted traceability supply chain system using blockchain technology [30, 31, 32].

A. RELATED WORK

Several cross-chain solutions have been proposed to address interoperability [33] and scalability issues. Cross chain is used as intermediator node like Federated sidechain [34], Cosmos [35, 36] and Polkadot [37]. In this interconnection, a blockchain that has the ability to import and export digital property connects different blockchains by an intermediate chain, or it may have individual participants who watch for fraud like Atomic Swap [38] and Plasma [39]. In [40], a side chain using trusted or semi-trusted intermediator is used. The AION network [41] connects independent blockchains by intermediators. The authors in [42] introduce a framework to switch users from blockchain to another. An atomic move operation is proposed in [43] using migration from blockchain to another and locking the smart contract in the smart contract in the source, and both blockchains have the same virtual machine. Authors in [44] proposed atomic loans using atomic swaps communication protocol. In loans systems transaction fee as in [45], it may end up very costly to the borrower.

The project in [46] uses provider nodes, while the platform in [47] uses distributed nodes to hold private keys. Smart Contract Invocation Protocol (SCIP) [48] enables the interaction of smart contracts where the interface acts as an intermediator. A cross-chain for multi micro-grids [49] is proposed to represent independent micro-grids that trade power to an external network.

As shown, the interprobability work depends on the purpose of doing it, it may be done for some applications as microgrid, IOT, and token exchange. Most of existing cross-chains use one of the following: intermediates, centralized exchange, token concept, trust, or semi-trust party, or convert data format. In this paper, we proposed a framework which has also its own purpose to make interconnection between public and private blockchains to have a local database of all transaction data in the organization to keep the privacy of organization and to reduce interaction with public blockchain with low cost and overhead.

B. OUR CONTRIBUTION

Organizational transactions with core idea are proposed to automate the process of updating the payment non-interactively. The proposed Secure Organizational Transactions Framework (SOTF) succeeds in achieving the following important contributions:

- Managing the relation between product, service and bitcoin using bridging contract.
- Saving the organization details privately in consortium chain or local database or storage and reducing public interaction with public blockchain.
- Keeping local database in organization blockchain connecting data from supply chain and bitcoin.
- Allowing smart contracts as there are two smart contracts deployed on the organization blockchain; 1) The bridging contract which has functions to manage the organizations' payments and approve bitcoin blockchain status; 2) The service contract that deployed in supply chain to interface and follow the service condition.
- Providing awareness and listening by allowing all nodes to have knowledge of the full blockchain and users who can access chain and approve transaction.
- Decreasing delay and contacts with public ledger as the update will be done automatically without need to update secret keys for each bitcoin payment update.
- Efficient cost where organization can buy their services and material with low cost as shown in our results.
- Scalability to large numbers of organizations without adding overheads on the bitcoin network as they can be managed by the bridging contract.

C. ROADMAP OF THE PAPER

The rest of the paper is organized as follows: Section II reviews some basic knowledge. The proposed SOTF framework is presented in section III. Section IV describes the implementation of the proposed framework including the algorithms for organizational blockchain transactions. The proposed framework is evaluated in section V. Finally, the paper is concluded in section VI.

II. PRELIMINARIES

In this section, the organizational structure and blockchain will be described.

A. ORGANIZATIONAL STRUCTURE

Organization is the process of performing, establishing, arrangement, ordering, or making a structure for classifying things. Organization transactions represent financial actions that affect the resources of a company. Organization transactions may be represented as business to business, consumer, or government. To make a payment in exchange, organization needs to create a legal contract. For example, an organization rent cars, it needs to schedule its car by order for different organizations as per organizations request. The scheduling system is crucial to service efficiency between them. This needs a storage system to and away to paid like bank or any other third party or coin-based system which is popular today.

Organization may be multinational company which contains a lot of branches in different geographical locations. Although most of money transaction will be separate for each location, but sometimes transaction may need money approve from different locations and sharing transaction details for review. For example, company material from one organization to another different one, material and money transfer from one organization to another may need to find a way to transfer cost

of material and to save transaction details and manage who can access it. These scenarios don't need anonymity as they know each other and want to make a transaction, but it needs to preserve privacy against public network and avoid its tracking in public blockchains. The same issue will be required in case of different organizations that have a common work with each other.

Organizations' transactions have three forms: 1) Service transaction in which an organization offers a service for another organizations as installation of material or technical device, specific type of learning courses, parking space, car rent, ….etc.; 2) Material transaction where organization may offer material that may be hardware or software as mobile network sites, building sites, computer material, program software….etc.; 3) Coin transaction which may be as bitcoin, Ethereum, zero-coin, altcoin or any other coin base which will be paid for martial or service one. For example, authors in [50] define seven laws of information explaining the difference between information commodity and traditional goods. Price model standard should depend on the authenticity, completeness and coherence of the data that contains valid date, price, service or material description as RFID/NFC module and specified APP usage.

B. ORGANIZATIONAL BLOCKCHAIN

Organization may contain public, private, supply or consortium chain need to make transaction to and from any type of ledger and safe trace information. Organizations use consortium chain with each other to have shared business data to get business goals with each other. As a part of consortium chain, organizations also contain supply chain information for material trace. We assume that there is a peer-to-peer network among these organizations and different peers want to connect to each other.

Networks can be classified into two types: centralized and decentralized networks. Centralized networks need a central authority for the flow of data among the nodes, while decentralized networks are not. Blockchain consists of ledger technology which is decentralized and don't need to have a central administration for managing the flow of data. The distributed database introduced by blockchain technology fundamentally changed the way of information processing. With the blockchain, the information can be entered into record and a community of users can control the way the information will be updated and amend, every node in the blockchain has an equal status. The consensus among the nodes is achieved through rules and protocols based on majority agreement. There are a lot of consensus mechanisms for replicating data among blockchains as Proof of Work (PoW), Proof of Stake (PoS), and RIFT; Each mechanism has different methods for collecting and approving blocks.

Supply chain maintains the lifecycle of the product from the production to the consumption. The data generated in every step can be documented as a transaction creating, and thus a permanent history of the product. It can record every single asset, track orders, receipts, invoices, payments, and track digital assets as warranties, certifications, copyrights, licenses, serial numbers, bar codes and others. Supply chain actors are parts of different organizations or different organization branches which use this chain to track products that move from organization to another as shown in Figure 1.

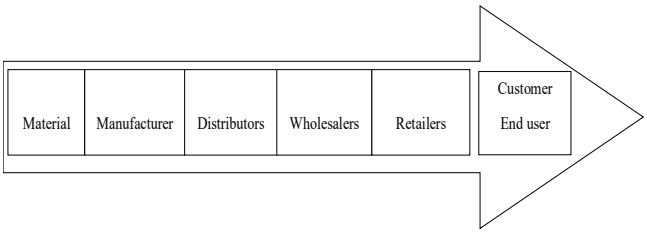

FIGURE 1. Supply chain actors

III. THE PROPOSED ORGANIZATIONAL TRANSACTIONS FRAMEWORK

In this section, the proposed organizational blockchain transactions framework will be introduced. The proposed framework allows organizations to have three transaction parts: First, transaction part with consortium which will save transaction details and confirm its correctness. The second transaction part is supply chain containing material and services details in consortium. The third transaction part is bitcoin network to make a payment.

In the proposed SOTF framework, the consortium will relate to bitcoin and supply chain to approve and save transaction and without the need of third party. Organizations that work together will have a common contract, payments, and transaction condition that will be saved in their consortium, or private chain within the payment channel to trigger automatic execution of the service deployed on these chains. Organization's consortium chain will be accessed only by authorized users who have the right to read and write data (i.e., issue transactions) to the blockchain.

In the proposed SOTF framework, public blockchain receives only the transaction contract in the start and the settlement at the end, while all updates are done in consortium, and bridging only confirms reservation and settlement. This will not take the same time as if all updates are done by bitcoin blockchain. Hence, the system doesn't grant the decrease congestion in public blockchain only, but it also makes a complete system to interconnect between different types of blockchains, ease the organization work, and reserve a full database system for tracking all transactions in the organization blockchain.

One of organizations represents the payee node; let's assume that organization1 which makes service and material transaction, and the second organization (organization2) represents the payer which pays coin for service. The payer organization creates a service contract for providing the service and shares it via the consortium or supply chain for the two organizations. Organization1 creates a service contract for

providing the service, where organization2 makes bitcoin payments which triggers execution of a smart contract on the consortium chain while the payment channel is open. The last state of payment is broadcasted as a settlement transaction by organization2 and is eventually stored in the blockchain after the two organizations signature.

There are two smart contracts deployed on the organization blockchain; 1) The bridging contract which has functions to manage the organizations' payments and approve bitcoin blockchain status; 2) The service contract that deployed in supply chain to interface and follow the service condition.

A. ORGANIZATIONAL TRANSACTION

The proposed SOTF framework is used for blockchain with a heterogeneous network that includes multiple sub-networks to handle a wide variety of business logic, with less transaction fees, and good scalability. Figure 2 represents the consortium model of our framework. The model consists of three layers: the first layer is a consortium blockchain running as an organization blockchain. The second layer is the bitcoin blockchain that spends coin for a service. The third layer is the supply chain to trace product. Finally, notary nodes to confirm bitcoin transaction and supply chain tracing through the bridging contract.

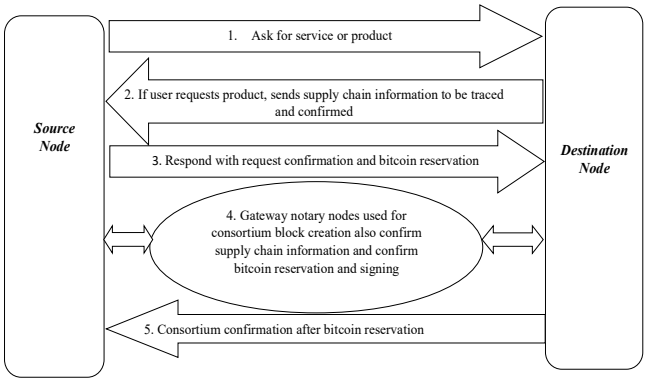

FIGURE 2. Organizational transaction steps

The steps of organization transactions are explained as follows:

1. Source node asks for a service or product from destination node.
2. Destination node sends authorization for supply chain to trace the product and find and send to source node the prices and all required data. If source node needs only service, destination node sends to him the price sheet and all needed data.
3. Source node sends to destination node confirmation for the needed service and creates the bitcoin contract data for bitcoin transaction.
4. Gateway notary nodes are used for consortium block creation, confirming supply chain information, and confirming bitcoin reservation and signing.
5. Bridging nodes either returns the signature of the transaction or returns a rejection error if no coin reserved or if smart contract not published to bitcoin blockchain.

B. SYSTEM STRUCTURE

The payer (Organization1) will put his service or material containing public key, service description, material RFC, RFID, or serial number (ser) as transaction information in supply chain. Organization2 which represents payee will ask for service information from orgnization1. It is mentionable that organization1 represents the payer party and organization2 represents the payee of the transaction.

The sequence of transactions is shown in Figure 3 and explained as follows:

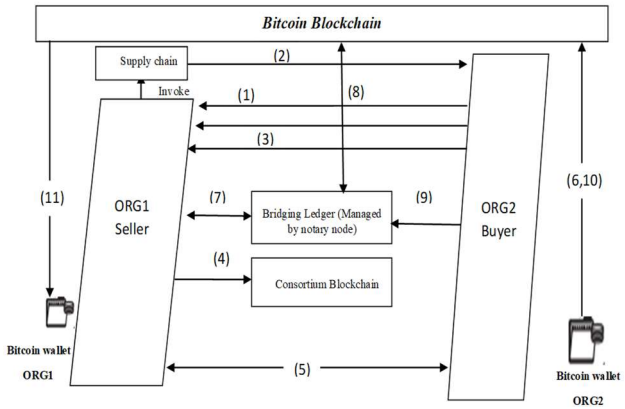

FIGURE 3. Organizational blocks interconnection

(1) Organization2 first requests a transaction.

(2) Organization1 invokes supply chain to track and get prices and quality (note, this step only exist for material buying part, if not, organization1 will creat service contract and detail from consortium directly) T_S supply chain transaction which contains material, tracking and price information in supply chain sent by organization1 to organization2.

(3) Organization2 confirms the needed service and prices, and sends a confirmation for T_S details.

(4) Organization1 creates a transaction template T_T containing material or services details as id, serial, price, and each organization address, and organization1 supply chain transaction requests T_S that will be shared with the organization chain. This transaction information depends on organization type; if for example organization chain is consortium for agricultural machinery scheduling system instead of material detail, it will contain distance informaion, available time, scalability, and so on, and contract will be published in consortium chain.

(5) The two organizations create a multi-signature address α_{12} to be used in the bitcoin funding

transaction T_B which will be used to put deposit in bitcoin platform $T_B^{\alpha_{12}} = (\alpha_{12}, \sigma_{12})$.

(6) Organization2 opens the bitcoin payment channel and broadcasts a funding transaction $T_B^{\alpha_{12}}$ to the bitcoin network which refers to $T_T\ (ID_2\ , Cert_2\ , ID_2, Cert_1, RFID, Ser, T_S)$. Where $T_B^{\alpha_{12}}$ sends organization2 bitcoins to the multi-signature accounts α_{12}. For a refund on $T_B^{\alpha_{12}}$, time-locked is used for the multi-signature accounts as in [51]; where when time expires, organization can return his deposits back to his/her wallet. The transaction $T_B(\alpha_{12}, \sigma_B\ , \alpha_2)$ saves bitcoins amount σ_B for the multi-signature accounts α_{12}. T_T specifies $T_B^{\alpha_{12}}$ as input and corresponds to the complete transaction T_{12}^{σ} as output; the subscripts 12 represents that it has the signatures of the two organizations which represents a valid bitcoin transaction sending σ bitcoins from the multi-signature address α_{12} to organization1's address α_1.

(7) Organization2 sends $T_B^{\alpha_{12}}$ to prove opening the channel in bitcoin blockchain, hence, organization1 creates a transaction $T_C\ (deposite(T_T, T_B^{\alpha_{12}}))$ where T_C (func) is a consortium chain transaction calling a function of the bridging contact. The function $deposite(\alpha_{12}, ID_2, Cert_2, ID_2,\ Cert_1,$ $\sigma_B, RFID, M_{ID}, \alpha_2, \sigma_{c1}, \sigma_{c2})$ stores deposit information including the multi-signature address α_{12}, deposit amount σ_B in the consortium chain, organization2 bitcoin wallet address α_2 which represents the input addrees of bitcoin transaction, organization2 identity and certificate and concertium address $ID_2\ , Cert\ _2, \sigma_{c2}$, organization1 identity and certificate and organization1 concertium address $ID_1\ , Cert\ _1$, $RFID\ , \sigma_{c2}$, a material description M_{ID}, and ID number or S_{ID} if it is a service in organization1.

(8) The bridging contract has a function getTemplate() for retrieving the stored $T_B^{\alpha_{12}}$. Bridging node confirms transaction in the bitcoin blockchain by using bridging contract in cosurtium chain, and invokes concertium chain contract T_C each time to update and confirm new payment update till the last one.

(9) Organization2 provides its signature sig_2 and update(Sig, σ) for updating signature each time he updates his request. The standard for bitcoin is ECDSA (the Elliptic Curve Digital Signature Algorithm) [52]. This updated signature is sent by organization2 and validated by bridging contract.

(10) Settlement: Organization2 finally obtains the last updated transaction signed by organization1 from the bridging node, then he signs it, and broadcasts it to the bitcoin network as a settlement transaction after bitcoin network validation.
The validation includes confirming whether the value of σ is less than σ_{12} and verifying that the signature Sig_2 is correct. Then, after bitcoin network enough confirmations (six confirmations), a complete bitcoin transaction equation is represented as follows: $T_B^{\sigma} = (\ T_B^{\alpha_{12}}, sig_1, sig_2, \sigma, \sigma_{1r})$ where sig_1 represents organization1 signature, sig_2 represents organization2 signature, σ_{1r} indicates that bitcoin is returned to organization2 account, σ bitcoin from multi-signature address to organization1. Bridging node confirms and puts the final tranaction state in consortium block.

(11) Now, bitcoin is transfered to organization1 bitcoin wallet.

It is mentionable that in a supply chain, ownership of products changes several times among participants until they are delivered to consumers without interrupting bitcoin network each time. In previous systems, if the consortium is a supply chain to use a valuable token for a consortium chain, you may need external authorities (e.g., banks) to ensure their values; in such case, final settlement is done via third authority outside the blockchain which may delay and cost more than blockchain time needed for approval and to be saved in a block. Hence, bridging automates the payment process with less contact with public blockchain, facilitates interconnection between different types of blockchains, and saves the organization details privately.

C. CONSENSUS SYSTEM

Deployment and transactions of contract in consortium blockchain cost less energy and time than in bitcoin. Consortium chains have faster times and use less energy than testnet and mainnet chains that we use for implementation because minnent and testnet don't use Proof of Work sybil resistance protocols. Proof of work [53] is used in bitcoin and Ethereum for mining and to prevent fork. However, consortium platform is more flexible as the number of validators in public blockchain leads to troubles with synchronization and mutual agreement that cause delay. The consensus in consortium is reached by a relatively small number of nodes, and hence, it is easier to be reached. These factors have a direct effect on transactions throughput leading to better scalability and executing operations faster. For example, bitcoin's block size is limited 1 Mb per 10 minutes [54], while consortium blockchain can optimize it to 1000 and more transactions per second [55].

Consortium blockchain uses voting-based algorithms for consensus as Paxos [56], Raft [57], PBFT (Practical Byzantine Fault Tolerance) [58], and RFBT (Redundant Byzantine Fault Tolerance) [59]. A consensus algorithm used for consortium blockchain uses Proof-of-Vote type of consensus which doesn't require much energy as there is no mining. In voting-based algorithms, the commitment depends on which committed result wins the majority of votes.

The consensus algorism using raft is easily for understanding and it is more popular in consortium. We will

illustrate the consensus algorithm using Raft as it is designed for ease of understandability and implmentability for industry applications, but the test will be done in Ethereum which uses proof of work consensus algorithm as it is the easiest way to test and get the approximate cost. The prices are approximately the same for them may be it costs more in Ethereum and bitcoin test than in consortium.

Consensus flow chart is ilusturated using raft in Figure 4 for the proposed SOTF framework. Block consensus approach for concertium organization blockchain consists of three stages: election stage, production stage, and verification stage.

At the election stage, the system selects a node from the candidate set that composed of all organizations as the bridging contract and block producer node, and each node in the candidate set has the same probability to be selected. Here, to prevent probability of inconsistencies, an organization creates multiple nodes into the block producer set, and only one node per organization can enter the candidate set which may be the head office of organization. After producing Q blocks, a new election would be held, and the new election starts, and then, turning to the production stage. If this node cannot produce a block within a certain period of time or it generates an illegal block (the appearance of a 'fork' also indicates that the block is illegal), a new bridging node would be selected. The other bridging nodes that don't create the block will act as a supervisor node to ensure the created block legality.

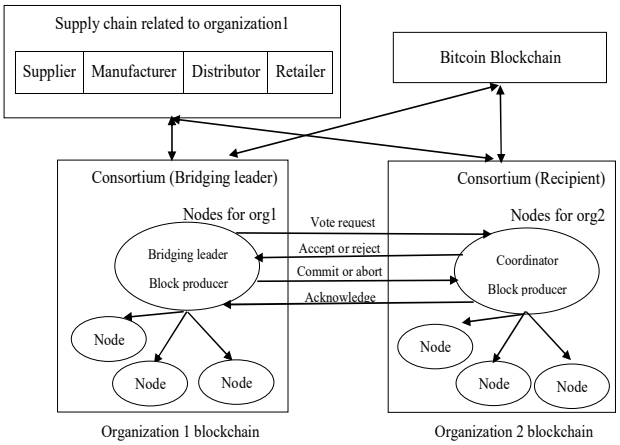

FIGURE 4. Consensus flow chart

After accepting and approving bitcoin transaction by bridging contract, the block will be saved in consortium blockchain. Algorithm 1 describes the block creation for the same blockchain.

Algorithm 1. Each block consensus algorithm

1. **Init**
2. Round=0
3. Block=0
4. Upon start do start Round (0)
5. Select block producer node
6. Input data (transaction request)
7. Data approved and available
8. Create new block
9. supervisor nodes approve new block
10. If block is legal then
11. Output new block, block ++
12. **If** q < Q, then
13. Input new data (transaction request)
14. **Else**
15. Round ++, StartRound (Round + 1)
16. **Repeat**

The above algorithm and chart are for block creation for the same blockchain; but to interact between blockchains [60, 61, 62], this needs a different algorithm for cross blockchain transaction as shown in Figure 5.

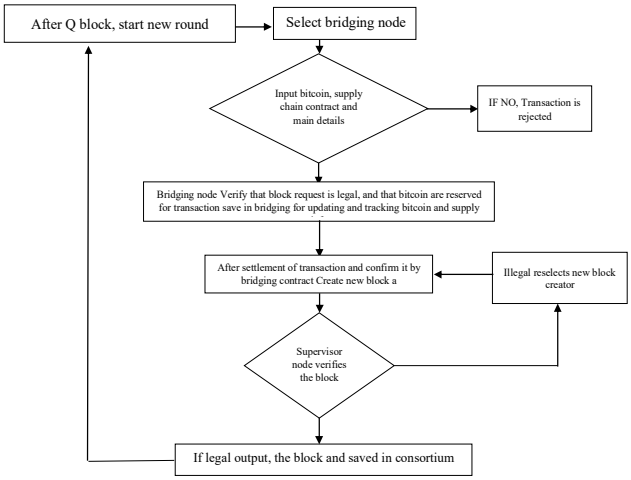

FIGURE 5. Inter blockchain connection

We need one of the blockchains to be elected as the bridging node by using election algorithm; the election algorithm elects the leader for the different blocks which will be responsible for bridging contract. Let's say that organization1 blockchain will send vote request to be the leader bridging node, organization2 and any other connected organization will represent the participants (this can be exetended to multiple organizations interaction). If receipient organizations agree on the transaction, the transaction will be committed; otherwise, it will be rejected. As shown in Algorithm 2, the inter connection will be as follows:

1. The Bridging Request Phase:
   a. Each blockchain node or organization node selects its leader node as shown in Algorithm1, and then, the leader bridging node sends a VOTE-REQUEST to all participants.

b. When the participants receive the VOTE-REQUEST, they will respond by accepting or rejecting the bridging leader; If it is rejected, the system will be terminated.

2. The Commit Phase

a. The bridging node collects votes from all participants. If all participants accept, the bridging node sends a COMMIT message to all participants; otherwise, it aborts and sends an ABORT message to all participants who have voted by accept.

b. Each participant, who voted by accept, needs to wait for a COMMIT or an ABORT message from the bridging leader node. Any node can abort in the waiting period for decision (COMMIT or ABORT) by the bridging node except the node which elects accept must wait for the bridging decision.

Algorithm 2. Inter blockchains consensus algorithm

INIT after each block leader node chooses
(Bridge leader node):
Bridge leader node sends VOTE-REQUEST to all participants
Wait for response messages from all participants
If all nodes voted by accept before time out, then write commit record in database log and send COMMIT to all participants
else write abort record in database log and send ABORT to all participant
end if
(Participant nodes):
Participant nodes wait for VOTE-REQUEST from bridging node
If participants vote by accept before timeout, then write an accept record in database log
Wait for decision message from bridging leader; if decision message is COMMIT, then write COMMIT record in database log
else write ABORT record in database log
end if
(Commit):
Receive message from the leader, then write message to log
If leader crashes, then
Run for the new leader
If elected, then synchronize the log
else skip
end if
end if

Here in our scenario, the bridging node takes the update and confirms information from bitcoin blockchain and supply chain. Finally, bridging node updates all blockchains with the last status and creates block to be saved.

IV. IMPLEMENTATION

The proposed bridging framework is implemented on oracle vm virtual box with linux(ubuntu-64bit) and 6 GRAM. The system installed the vm on hp laptop Intel(R) Core(TM) i7-9750H CPU @ 2.60GHz 2.59 GHz, 16.0 GB RAM and 64-bit operating system, x64-based processor Windows 10 Home Single Language. We use testnet and mainnet for system evaluation as they give a good estimation on the cost of using the proposed framework on the network. The prices are approximately the same for Ethereum and Bitcoin. By deploying to a testnet, we can approximate how much it will cost to deploy to the real network. The source code will be accessible at (https://github.com/shereenmo/Bitcoin-Supply-Bridge.git).

To implement our code, we used Python 68.2% and Solidity 31.8%. The code base has three major components: (i) the supply chain implementation; (ii) the bridging contract; and (iii) the bitcoin wallet. For simplicity, Ethereum blockchain is used for implementation and bitcoin testnet.

Supply chain is created with Chips product, Gate Type: NAND, NOR, Number of Pins: Six, Eight, Twelve, and Fourteen as shown in Algorithms (3-8) which describe our supply chain that defines the working principles of our proposed SOTF blockchain-based framework. We use smart contract to provide a framework for standardized applications, and we can identify the behavior and status of the contract by identifying the key parameters of the smart contract. Smart contract is a programming language description of certain requerments.

Algorithm 3 describes the process of creating chips and sets the default parameters. These parameters represent pins, voltage, name and price. Then, store this data to organization electrical department.

Algorithm 3. Create chips - Set the default parameters for a type of chip

Receive gate, pins, price (in cents), voltage (in mV), name (of item), number of gates from admin
Use gate and pins as **keys** to mapping to store above data.
Additionally **set** manufacturer to "XYZ Electronics Inc.", department to “electrical” and number of items to 1000 by default

Algorithm 4 demonstrates how the organization can change the chips properties as voltage or price.

Algorithm 4. Change properties - Change the price or voltage for a chip

Receive gate, pins, price, and voltage from user

Check to see if gate and pins number is valid
If new price is not zero, **then**
Replace current price of item with new price using gate, pins as **keys** to chips mapping.
End if
If new voltage is not zero, **then**
Replace current voltage of item with new voltage using gate, pins as **keys** to chips mapping.
End if

Algorithm 5 demonstrates how the organization can add new product or more items to existing chips to be saved in supply chain information.

Algorithm 5. Add products - Add more items of a particular type

Receive gate, pins, number of items to add from user
Check number of items to add must be strictly greater than zero
Add number of new items to current number of items using gate, pins as **keys** to chips mapping.
Emit event to display the information on supply chain app

Algorithm 6 describes the ability of removing products from the current supply chain. The Algorithms (3, 4, 5, 6) represent the control part of the supply chain software. All demo photos is attached with full software and bridging contract in GitHub (https://github.com/shereenmo/Bitcoin-Supply-Bridge.git).

Algorithm 6. Remove products - Removes products from current supply

Receive gate, pins, number of items to remove
Check number of items to remove is strictly greater than zero
Check number of items to remove is less than or equal to the available number of items for the product.
Subtract number of items to remove from current number of items available using gate, pins as **keys** to chips mapping.

Algorithm 7 describes how the organization can buy items and specify the properties of items.

Algorithm 7. Buy product - Buy items from an array of 8 numbers

Receive array of 8 unsigned integers representing number of items to buy from each product sold. They are in the following order along with their index: 0: NAND-6, 1: NOR-6, 2: NAND-8, 3: NOR-8, 4: NAND-12, 5: NOR-12, 6: NAND-14, 7: NOR-14.
For index in 0 to 7
If number of items is equal to zero, **then**
Continue to next iteration of loop
End if
// Get Gate Type
If index modulus 2 is equal to zero, **then**
Gate is of type "NAND"
Else
Gate is of type "NOR"
End if
// Get Pins Type
If index divided by 2 is equal to 0 **then**
Pins is of type "Six"
Else if index divided by 2 is equal to 1 **then**
Pins is of type "Eight"
Else if index divided by 2 is equal to 2 **then**
Pins is of type "Twelve"
Else
Pins is of type "Fourteen"
End if
Call Remove Products function passing gate, pins, and number of items from above.
End for
Emit event to show that given

Algorithm 8 shows the ability to remove some requested items from available chips until the end of buying process. The buyer sends request to supply chain to reserve this item and to be removed from supply available items.

Algorithm 8. Defective products - Remove defective products

Receive array of 6 unsigned integers representing number of items that are defective from each product sold. They are in the following order along with their index: 0: NAND-6, 1: NOR-6, 2: NAND-8, 3: NOR-8, 4: NAND-12, 5: NOR-12, 6: NAND-14, 7: NOR-14.
For every integer (number of items) in input array
If number of items is equal to zero, **then**
Continue to next iteration of loop
End if
// Get Gate Type
If index modulus 2 is equal to zero, **then**
Gate is of type "NAND"
Else
Gate is of type "NOR"
End if
// Get Pins Type
If index divided by 2 is equal to 0 **then**
Pins is of type "Six"

Else if index divided by 2 is equal to 1 **then**
Pins is of type "Eight"
Else if index divided by 2 is equal to 2 **then**
Pins is of type "Twelve"
Else
Pins is of type "Fourteen"
End if
Call Remove Products function passing gate, pins, and number of items from above.
End for
Emit event to show that given number of items were removed from the supply chain due to being defective

The bridging transaction contract algoritms are given in Algorithms (9-14); every transaction is characterized by the following data organized into a struct: 1) Transaction state is shown in Algorithm 9, 2) A list of items Ids is created in Algoritm 10, 3) A total amount due (in cents) for all the items in the cart, 4) A list of 3 dates (creation date, completion/payment date, refund date) is created, 5) Seller's signature of approval shown in Algorithm 11, 6) Buyer's signature of approval shown in Algorithm 12.

Algorithm 9. Create transaction - Creates a new transaction

Receive receipt number as input
Increment total number of transactions by one
Set state of new transaction to "Created"
Set first date of dates list for the new transaction as current block time
Emit an event to signal a new transaction

A list of items Ids is created in Algorithm 10 where transaction items list is mapped by the receipt number and item price is mapped by the receipt number, a total amount due (in cents) for all the items is added in the cart, and a list of 3 dates (creation date, completion/payment date, refund date) is created.

Algorithm 10. Add items to transaction

Receive receipt number, a list of item IDs, a list of item prices
Check transaction state of receipt number is set to "Created"
Check length of list of item IDs is equal to length of list of item prices
For item ID in input list of item IDs
Push item ID into transaction items list mapped by the receipt number
End for
For item price in input list of item prices
Add item price to transaction total price mapped by the receipt number
End for
Emit an event to show that the transaction characterized by the given receipt number was updated with a new total.

Seller's signature required for approval is shown in Algorithm 11 in which seller signs the request, then the request is sent by bridging to buyer.

Algorithm 11. Confirm Seller Approval - Seller approves transaction

Receive receipt number as input
Check transaction state of receipt number is set to "Created"
Set seller signature to "true" of transaction mapped by receipt number
Emit an event to let buyer

When buyer receives seller signature of approval, he also can sign, and the bitcoin is reserved as shown in Algorithm 12.

Algorithm 12. Confirm Buyer Approval - Buyer approves transaction

Receive receipt number as input
Check transaction state of receipt number is set to "Created"
Check to see if seller has given their approval
Set buyer signature to "true" of transaction mapped by receipt number
Emit an event to show that both the buyer and seller have approved the transaction and are ready to proceed to payment.

Algorithm 13 demonstrates how the bridging checks the two signatures and bitcoin being quoting to set the transaction with complete date representing current block date to be saved.

Algorithm 13. Pay Transaction

Receive receipt number as input
Check transaction state is set to "Created" or "Failed"
Check for seller approval signature
Check for buyer approval signature
Set state of transaction to "completed" mapped by receipt number
Set 2nd date in transaction date list (completed date) to current block time

Emit an event to start the payment process on the bitcoin network quoting the receipt number and the total (in cents)

Hence, the transaction can be mapped from their receipt number (key) to the transaction information (struct). The transaction state can be of the following type: Created, Completed, Failed, and Refunded. When organization needs to refund transaction, the transaction state is set to refunded as shown in Algorithm 14. Then, seller balance in bitcoin wallet can be checked.

Algorithm 14. Refund Transaction

Receive receipt number as input
Check if transaction state is set to "completed"
Set transaction state to "Refunded"
Set 3rd date in transaction dates list (refunded date) to current block time
Emit an event letting buyer and seller know that the transaction was refunded and to begin refund process on bitcoin network

V. FRAMEWORK EVALUATION

Using consortium, transaction fees can be estimated according to [63]: On private or consortium blockchains cost of deployment can be in range between 100-1000 USD which means gase price from 18.9007 to 189.007. Additionally, according to [64], the total cost to operate an application may be lower on mainnet than running a private chain. However, consortium blockchains are hybrids between private blockchains and public blockchains and tend to lean more to the private side. When a bunch of organizations come together and make a blockchain to ease communications between themselves, these organizations can control how much each transaction will cost.

Generally, in privatized blockchains (Private or Consortium), the practice is to not have any transaction cost at all. This is because to join one of these networks, you either have to pay an entry fee or to be a part of that organization itself. All in all, deployment and transactions in smart contracts in consortium cost next to nothing in actual money. As far as time and energy is concerned, consortium chains have faster times and use a lot less energy than testnet and mainnet chains because they don't use Proof of Work sybil resistance protocols. We deploy bridging contract, supply chain contract and bitcoin contract using Etherum and bitcoin testnet. The deployment cost of the proposed SOTF framework is demonstrated in Table I.

Bitcoin testnet execution prices getting transaction information from [65], transaction fees per transfer of BTC: 0.00001122 BTC. In a public blockchain, there are miners or validators depending on the sybil resistance protocol. The amount of gas and gas fees we pay is for these miners who are using their resources to add our transaction to the block they are mining. When mining rewards, e.g. 6.25 BTC per block mined, these miners will only get these gas fees we give them.

As shown in Table II, this cost is estimated by bitcoin testnet and log file. This estimated cost may be affected by the type of blockchain and the network connection, but it will still duable.

TABLE I
DEPLOYMENT COST OF THE PROPOSED SOTF FRAMEWORK

Smart Contract	ABI Name	Amount of Gas Used	Transaction Cost (Eth)	Transaction Cost (USD)
Supply Chain Contract	Deployment	2515454	0.003773181	10.61573161
	Change Properties	30153	4.52295E-05	0.127251842
	Add Products	31512	4.7F8E-05	0.132987101
	Buy Product	97960	0.00014694	0.413411284
Transaction Bridge Contract	Deployment	1026368	0.001539552	4.331483391
	Create Transaction	85115	0.000127673	0.359202751
	Add Items to Transaction	265246	0.000397869	1.119392502
	Confirm Seller	46257	6.93855E-05	0.195214024
	Confirm Buyer	32244	4.8366E-05	0.136076291
	Pay Transaction	53412	8.0118E-05	0.225409591
	Refund Transaction	51381	7.70715E-05	0.216838354

VI. CONCLUSION

Blockchain technology is rich challenge work media related to scalability, interprobability, identity registration, privacy, and regulations. In this paper, a Secure Organizational Transactions Framework (SOTF) is introduced based on bitcoin payment bridge to preserve the privacy of transacted organizations. Additionally, we provided an open-source code of supply chain, bitcoin blockchain, Ethereum blockchain, smart contracts for bitcoin paying, and interpretable bridging code. Furthermore, we presented details and aspects related to the system architecture, design, consensus interactions and implementation algorithms. Generally, this communication happens bi-directionally since in a real blockchain network, nodes keep getting created and become inactive; this is done in order to not isolate any node in the network.

TABLE II
DEPLOYMENT COST OF THE PROPOSED SOTF FRAMEWORK

Bitcoin Blockchain (Hash)	Amount of Money Transfered (USD)	Transaction Fees (BTC)	Transaction Fees (USD)
e30d408509cce20d005fb08dd05d32e4eaf7e5b8dad472962e1699fc90450922	38.01	0.00001496	0.603012168
f2d3f7d07ff94cc97f4213456e1171b90c3fdacfcd9c92bbde647752c119a59a	1,249.41	0.00001496	0.603012168
9027766db0c630d40cc2ed6b24f69529322465f0c446527db70d044ce818e433	39.23	0.00000374	0.150753042
6fd78edf8200669f12def1612dc285c00ac52c5918989a3150065a1a680f5f2f	1,358.69	0.00000374	0.150753042
285fe698081bb92147ca30c86e5c35e505d93efd61250401083bc03aa1e9326f	334.96	0.00000818	0.329721894
d5c3c1f7467a2de6fa1aef6fc5afc3c16431739f2daa69008f5d6edfe53b8b29	1,193.56	0.00000226	0.091096758
fd44f4742db1e3c1055b16c7be8e04f58e1bb993b378bfa0e9d2e1e78114ee22	21.24	0.0000603	2.43059049

In addition, a number of proposed algorithms are incorporated in the proposed SOTF framework to make transparent transactions, and these algorithms have been implemented and shown as open-source software system. Moreover, a software case study is presented to address the interprobability and cost issues. The experimental results demonstrate that the proposed SOTF framework is applicable and has a managed cost.

As mentioned before, the reason we test and deploy to test media testnet and mainnet is because they give a good estimation on the cost of using the proposed framework in the network. The prices are approximately the same for all of them. However, further research on implementation of our bridging system in different blockchains with different consensus algorithms will be considered in our future work to determine the cost.

REFERENCES

[1] S. Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System," 2008. Available: https://bitcoin.org/bitcoin.pdf
[2] S. Barber, X. Boyen, E. Shi, and E. Uzun, "Bitter to Better—How to Make Bitcoin a Better Currency," Financial Cryptography and Data Security, 2012.
[3] Ch. Decker, J. Seidel, and R. Wattenhofer, "Bitcoin Meets Strong Consistency," arXiv preprint arXiv:1412.7935, 2014.
[4] M. N. BHUTTA, A. A. KHWAJA, A. NADEEM, H. F. AHMED, M. K. KHAN, M. A. HANI AND H. SONG, "A SURVEY ON BLOCKCHAIN TECHNOLOGY: EVOLUTION, ARCHITECTURE AND SECURITY," 10.1109/ACCESS.2021.3072849.
[5] A. Azaria, A. Ekblaw, T. Vieira and A. Lippman, "MedRec: Using Blockchain for Medical Data Access and Permission Management," In 2nd International Conference on Open and Big Data (OBD). IEEE, pp. 25-30, 2016.
[6] K. Christidis and M. Devetsikiotis, "Blockchains and Smart Contracts for the Internet of Things," IEEE Access, vol. 4, pp. 2292-2303, 2016.
[7] P. McCorry, S. Shahandashti and F. Hao, "A Smart Contract for Boardroom Voting with Maximum Voter Privacy," In 21st International Conference on Financial Cryptography and Data Security (FC). Springer, Cham, pp. 357-375, 2017.
[8] A. Goranovic, M. Meisel, L. Fotiadis, S. Wilker, A. Treytl and T. Sauter, "Blockchain Applications in Microgrids and Overview of Current Projects and Concepts," In 43rd Annual Conference of the IEEE Industrial Electronics Society (IECON), pp. 6153-6158, 2017.
[9] Y. Yuan and F. Wang, "Towards Blockchain-Based Intelligent Transportation Systems," in IEEE 19th International Conference on Intelligent Transportation Systems (ITSC). IEEE, pp. 2663-2668, 2016.
[10] C. Decker, R. Wattenhofer, "A Fast and Scalable Payment Network with Bitcoin Duplex Micropayment Channels," In: Pelc, A., Schwarzmann, A.A. (eds.) SSS 2015. LNCS, vol. 9212, pp. 3–18. Springer, Heidelberg (2015)
[11] J. Poon and T. Dryja, "The Bitcoin Lightning Network: Scalable Off-chain Instant Payments," (2016), https://lightning.network/lightning-network-paper.pdf. [Accessed 1- March- 2022]
[12] C. Fan, S. Ghaemi, H. Khazaei, and P. Musilek, "Performance Evaluation of Blockchain Systems: A Systematic Survey," Digital Object Identifier 10.1109/IEEEACCESS.2020.3006078
[13] M. Stojanović, N. Radović, A. Njeguš, "Opportunities and Challenges of Applying Blockchain technology at Airports,". 5th International Scientific Conference – EMAN 2021, DOI: https://doi.org/10.31410/EMAN.2021.157
[14] U. Jafar, M. J. Ab Aziz and Z. Shukur, "Blockchain for Electronic Voting System—Review and Open Research Challenges," Sensors 2021, 21, 5874. https://doi.org/10.3390/s21175874
[15] S. Yawar, A. Zaidi, M. A. Shah, H. A. Khattak, C. Maple, H. T. Rauf, A.M. El-Sherbeeny and M. A. El-Meligy, "An Attribute-Based Access Control for IoT Using Blockchain and Smart Contracts Sustainability," 2021, 13, 10556. DOI: https://doi.org/10.3390/su131910556
[16] J. C. Baek, "The Loop: Hyperledger Fabric, R3 Corda," (2017). https://blog.theloop.co.kr/2017/02/20/hype rledger-fabric-r3-corda/14.
[17] A. Pinna, A. Pinna, G. Baralla, R. Tonelli, "A Massive Analysis of Ethereum Smart Contracts Empirical Study and Code Metrics," IEEE Access, Vol. 7, 2019, pp. 78194 – 78213, DOI: 10.1109/ACCESS.2019.2921936
[18] U. Khan, Z. Yongan, and A. Imran, "A Blockchain Ethereum Technology-Enabled Digital Content: Development of Trading and Sharing Economy Data," IEEE Access, Vol. 8, 2020, pp. 217045 – 217056, DOI: 10.1109/IEEEACCESS.2020.3041317
[19] S. Kushwaha, S. Joshi, D. Singh, M. K Aur, and H. No Lee, "Systematic Review of Security Vulnerabilities in Ethereum Blockchain Smart Contract," IEEE Access, Vol. 10, 2022, pp. 6605 - 6621, DOI: 10.1109/IEEEACCESS.2021.3140091
[20] M. Vukoli´c, "Hyperledger Fabric: Towards Scalable Blockchain for Business," Technical report, Trust in Digital Life 2016, IBM Research (2016)
[21] G. Wood. "Ethereum: A Secure Decentralised Generalised Transaction Ledger," Ethereum Project Yellow Paper, 2014.
[22] Samsung SDS: Blockchain Platform Case—Samsung Nexledger (2018)
[23] K. Salah, N. Nizamuddin, R. J. Raman, and M. Omar, "Blockchain-Based Soybean Traceability in Agricultural Supply Chain," IEEE Access, 2019, Vol. 7, pp. 73295 – 73305, DOI: 10.1109/IEEEACCESS.2019.2918000.
[24] X. Pham; A. Maag; S. Senthilananthan; M. B. Predictive, "Analysis of the Supply Chain Management using Machine Learning Approaches: Review and Taxonomy," 2020 5th International Conference on Innovative Technologies in Intelligent Systems and

Industrial Applications (CITISIA).” 09 March 2021. DOI: 10.1109/CITISIA50690.2020.9371842.
[25] D. Sathya, S. Nithyaroopa, D. Jagadeesan and I. J. Jacob, “Block-chain Technology for Food supply chains,” 2021 Third International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV), March 2021. Tirunelveli, India. DOI:10.1109/ICICV50876.2021.9388478.
[26] J. Chen, T. Cai, W. He, L. Chen, G. Zhao, W. Zou and L. Guo, “A Blockchain-Driven Supply Chain Finance Application for Auto Retail Industry”, Entropy, 2020, 22, 95; DOI:10.3390/e22010095.
[27] S. Chang and Y. Chen, “When Blockchain Meets Supply Chain: A Systematic Literature Review on Current Development and Potential Applications,” 2020, IEEE Access, Vol. 8, 2020, pp. 62478 – 62494. DOI: 10.1109/IEEEACCESS.2020.2983601.
[28] L. Wang, L. Xu, Z. Zheng, S. Liu, X. Li, L.Cao, J. Li, and CH. Sun, “Smart Contract-Based Agricultural Food Supply Chain Traceability,” IEEE Access, Vol. 9, 2021, pp. 9296 – 9307, DOI: 10.1109/IEEEACCESS.2021.3050112.
[29] H. M. Kim and M. Laskowski, “Towards an Ontology-Driven Blockchain Design for Supply Chain Provenance,” Intelligent Systems in Accounting, Finance, and Management, 25(1), pp. 18-27, 2016.
[30] L. A. Ramasamy, K. F. Khan, A. L. Imoize, J.O. Ogbebor, S. Kadry, and S. Rho, “Blockchain-Based Wireless Sensor Networks for Malicious Node Detection: A Survey,” IEEE Access, Vol. 9, 2021, pp. 128765 – 128785, DOI: 10.1109/IEEEACCESS.2021.3111923
[31] W. HONG, J. YOU, HE. ZHANG, “A SURVEY ON BLOCKCHAIN APPLICATION: A DATA COLLABORATION PERSPECTIVE,” IN 2021 16TH INTERNATIONAL CONFERENCE ON COMPUTER SCIENCE & EDUCATION (ICCSE), 2021, DOI: 10.1109/ICCSE51940.2021.9569248.
[32] M. Pilkington, “Blockchain Technology: Principles and Applications,” In: Research Handbook on Digital Transformations,” Elgaronline; 2016. p. 225.
[33] S. Khan, M. B. Amin, A. T. Azar, and SH. A Slam, “Towards Interoperable Blockchains: A Survey on the Role of Smart Contracts in Blockchain Interoperability," IEEE Access, Vol. 9, 2021, pp. 116672 - 116691, DOI: 10.1109/IEEEACCESS.2021.3106384.
[34] A. Back, M. Corallo, L. Dashjr, M. Friedenbach, G. Maxwell, A. Miller, A. Poelstra, J. Timon, and P. Wuille, “Enabling Blockchain Innovations with Pegged Sidechains,” 2014. [Online]. Available: https://blockstream.com/sidechains.pdf. [Accessed: 1- March- 2022].
[35] J. Kwon, E. Buchman, “Internet of Blockchains - Cosmos Network,” 2016. [Online]. Available: https://cosmos.network/about/whitepaper. [Accessed: 1- March- 2022].
[36] Cosmos Network. https://cosmos.network, [Accessed: 1- March- 2022].
[37] G. Wood, “POLKADOT: Vision for a Heterogeneous Multi-Chain Framework,” 2016 [Online]. Available: https://github.com/w3f/polkadot-white-paper/blob/master/PolkaDotPaper.pdf. [Accessed: 1- March- 2022].
[38] T. Nolan, “Re: Alt Chains and Atomic Transfers,” 2013. [Online]. Available: https://bitcointalk.org/index.php?topic=193281.msg2224949# msg2224949. [Accessed: 1- March- 2022].
[39] J. Poon, V. Buterin, “Plasma: Scalable Autonomous Smart Contracts,” 2017. [Online]. Available: https://plasma.io/plasma.pdf. [1- March- 2022].
[40] S. D. Lerner, “RSK,’’ Tech. Rep., 2015
[41] M. Spoke, ‘‘Aion: Enabling the decentralized internet,’’ AION, White Paper, Jul. 2017.
[42] P. Frauenthaler, M. Borkowski, and S. Schulte, “A framework for blockchain interoperability and runtime selection,” 2019, arXiv:1905.07014. [Online]. Available: http://arxiv.org/abs/1905.07014
[43] E. Fynn, A. Bessani, and F. Pedone, “Smart contracts on the move, ” in Proc. 50th Annu. IEEE/IFIP Int. Conf. Dependable Syst. Netw. (DSN), Jun. 2020, pp. 233–244.
[44] M. Black, T. Liu, and T. Cai, “Atomic loans: Cryptocurrency debt instruments,” 2019, arXiv:1901.05117. [Online]. Available: http://arxiv.org/abs/1901.05117
[45] A. Zamyatin, D. Harz, J. Lind, P. Panayiotou, A. Gervais, and W. Knottenbelt, “XCLAIM: Trustless, interoperable, cryptocurrency backed assets,” in Proc. IEEE Symp. Secur. Privacy (SP), May 2019, pp. 193–210.
[46] COMIT Project. Accessed: June. 10, 2022. [Online]. Available: https:// comit.network/docs/comit-protocol/comit-projects/
[47] A Cryptofinance Platform. Accessed: June. 10, 2022. [Online]. Available: https://uploads-ssl.webflow.com/Whitepaper.pdf
[48] G. FalaziEmail, U. Breitenbücher, F. Daniel, A. Lamparelli, F. Leymann, and V. Yussupov, “Smart contract invocation protocol (SCIP): A protocol for the uniform integration of heterogeneous blockchain smart contracts,” in Proc. Int. Conf. Adv. Inf. Syst. Eng. Cham, Switzerland: Springer, 2020, pp. 134–149.
[49] L. Wang, J. Wu, R. Yuan, D. Zhang, J. Liu, S. Jiang, Y. Zhang, and M. Li, “Dynamic adaptive cross-chain trading mode for multi-microgrid joint operation,” Sensors, vol. 20, no. 21, p. 6096, Oct. 2020.
[50] D. L. Moody, P. Walsh, “Measuring the Value of Information-An Asset Valuation Approach,” [C]//ECIS. 1999: 496-512.
[51] P. Todd, “OP CHECKLOCKTIMEVERIFY,” 2014. [Online]. Available: https://github.com/bitcoin/bips/blob/master/bip-0065.mediawiki. [Accessed: 1- March- 2022].
[52] L. Zih, H. Y Shen, “On the Elliptic Curve Digital Signature Algorithm.,” Tunghai Science Vol.8, pp. 109-126 July. (2006).
[53] Z. Zheng, S. Xie, H. Dai, X. Chen and H. Wang, "An Overview of Blockchain Technology: Architecture, Consensus, and Future Trends," in IEEE 6th International Congress on Big Data, 2017.
[54] A. E. Gencer, S. Basu, I. Eyal, R. van Renesse, E. G. Sirer, " Decentralization in bitcoin and ethereum networks", arXiv preprint arXiv:1801.03998, 2018.
[55] https://blockchain.intellectsoft.net/blog/how-the-consortium-blockchain-works/
[56] L. Lamport. “Paxos Made Simple,” 2001.
[57] D. Ongaro, and J. Ousterhout.” In search of an understandable consensus algorithm,” In USENIX Annual Technical Conference, 2014.
[58] M. Castro, B. Liskov,” et al. Practical byzantine fault tolerance,” In OSDI, volume 99, pages 173–186, 1999.
[59] P. Aublin, S. B. Mokhtar, and V. Qu´ema, “RBFT: Redundant Byzantine Fault Tolerance,” In 2013 IEEE 33rd International Conference on Distributed Computing Systems, pages 297–306, July 2013. ISSN: 1063-6927.
[60] D. Zhao, “Cross-blockchain transactions,” In Conference on Innovative Data Systems Research (CIDR), 2020.
[61] V. Zakhary, D. Agrawal, and A. El Abbadi, “Atomic Commitment across Blockchains,” CoRR, abs/1905.02847, 2019.
[62] M. Herlihy, L. Shrira, and B. Liskov, “Cross-chain Deals and Adversarial Commerce,” PVLDB, 13(2):100–113, 2019.
[63] https://www.leewayhertz.com/cost-of-blockchain-implementation/
[64] https://ethereum.org/en/enterprise/
[65] https://www.blockchain.com/btctestnet/address/mhYd2xPs8opYrvPyZyihQTcSywxZvjifGu

Shereen M. Mahgoub received her B.Sc. degree in Electronics, Communications, and Computers Engineering from Helwan University, Cairo, Egypt, in 2007. She received her M.Sc. degree in network security from Helwan University, in 2013. She is currently pursuing the Ph.D. degree in cryptography and network security with Helwan University, Cairo, Egypt. She is now a network engineer in Egyptian Radio

Television Union. Her research interests include authentication, privacy, access control, blockchain, and bitcoins.

I. I. Ibrahim received his B.Sc. degree with honor in communications engineering from Helwan University, Cairo, Egypt, in 1976. He received his M.Sc. in communications from Cairo University, Cairo, Egypt in 1983 and PhD in communications from Queen University, Belfast, UK in 1987. He is a member of National Communications and Electronics Engineering Promotion Committee, the former Head of Electronics and Communications Engineering Department, and a professor of wireless communications with the Faculty of Engineering, Helwan University. His current research interests include wireless communications, Internet of Things, device-to-device communications, and network security.

Fatty M. Salem received her B.Sc. degree in Electronics, Communications and Computers Engineering from Helwan University, Cairo, Egypt, in 2007. She received her M.Sc. and PhD degree in network security from Helwan University, in 2010 and 2014 respectively. Currently, she is an Associate Professor in the department of Electronics and Communications, Faculty of Engineering, Helwan University, Egypt. Her research interests include authentication, privacy, access control, applied cryptography, blockchain, bitcoins, and mobile security. More specifically, she is working on the design of efficient and secure cryptographic algorithms and protocols.