Advances in Dynamical Systems and Applications.
ISSN 0973-5321, Volume 16, Number 2, (2021) pp. 1705-1723

https://www.ripublication.com/adsa.htm

Toward Safety of Wireless Sensor Network Based on Blockchain

Hafsa Benaddi[1], Khalil Ibrahimi[1], Haytham Dahri[1], Abderrahim Benslimane[2]

[1]Ibn Tofail University, Faculty of Sciences, Computer Science Research Lab, Kenitra, Morocco.

[2]University of Avignon, CERI/LIA, France.

Abstract

The Internet of Things (IoT) has received a lot of attention worldwide in terms of security and private grants. Due to the increase of massive data transmission, many challenges arise in IoT applications including the database of critical infrastructure which has become vulnerable to a lot of legitimate and malicious activities in centralized as well as decentralized architectures. This paper proposed a new model called (SCHD) to Secure Cluster-Header Decision for sensing Wireless Sensor Network (WSN) data as a prominent paradigm in IoT. Our approach is based on Blockchain aiming to improve the safety of transactions and efficiency to build a low-cost system. Furthermore, it applies the concept of Ethereum's smart contracts to check and validate the confidentiality of each node in a decentralized mechanism. Afterward, the framework's implementation is deployed on the Ganache environment using solidity as a programming language, used to write smart contracts to reach and resolve for each node the flexibility and efficiency for proof of work. The simulation results of our proposed model outperform the existing approaches, such as this study achieves integrity, agreement, validity, availability, and termination, which make the whole system a promising solution by decreasing energy consumption and building a low cost of the smart contract. As a result, it improves the architecture's effectiveness and trustworthiness.

Keywords: Blockchain, Internet of Things; Wireless Sensor Network; Smart Contact; Decentralized Control; Mining; Cryptography; Ethereum.

[1] hafsa.benaddi@uit.ac.ma

1. INTRODUCTION

In the last few years, electronic transactions based on Blockchain have become very popular due to the good reputation of this technology. Although, this electronic transaction cannot deal with the dangerous activities and the risks that may require the use of cryptocurrency [1] [2]. In this context, a centralized network of the election systems has been hacked, where the governments have intimidated in an illegal way citizen in order to give their elect to a specific candidate using violence [3], this occurred several times in USA and RUSSIA elections. Thus, with the appearance of Blockchain as a novel technology, it may solve and facilitate these issues of network management in a dynamic way.

The Blockchain is sharing ledgers between users in the network, while it is not controlled by a centralized method and distributed over nodes. These ledgers are organized as a record book where each transaction performed between users is stored in chronological order. Firstly, the public Blockchain is employed based on the currency or token programmable, contrary to the private Blockchain which is considered by a predefined set of users. Secondly, they are two levels of permissions required to add the information to the Blockchain: the permissioned Blockchain that permits just a selected group of users to write (i.e. generate transactions for the ledger to record) and commits (i.e. verify new blocks for addition to the chain) [4]. In contrast, permissionless Blockchain prevents anyone to contribute and add data to the ledger [5]. As a subfield of applications, Bitcoin is considered as a public programmable currency, such as transactions occurring between nodes among the network being grouped in blocks. Each one of these nodes is validated by a master node named "miner" that follows a mechanism relying on the Blockchain type for the validation. In the process followed by the Bitcoin called in the literature by the "Proof-of-Work" (PoW), a mathematical problem should be solved for validation at each time-stamped, and the winner block is added to the Blockchain network.

Afterward, the transaction of receivers will be seen either by the receiver or the entire network. The delay spent in this process depends in general on the type of Blockchain as seen in bitcoin and Ethereum where their time-stamped are respectively 10 and 15 seconds [6][7]. More precisely, Ethereum is the ultimate and well-established, openhanded decentralized software platform that allows Decentralized Applications (DApps) and smart contracts to construct, build and run without any third party to prevent and exclude any downtime, fraud, control, or interception permanently. It uses (Ether) as a cryptocurrency, which is widely used to run applications, enhance the work and monetize it [8]. Consequently, Ethereum outperforms bitcoin regarding its time-step.

Therefore, *Wireless Sensor Network* (WSN) is a set of physical sensors deployed as a network resulting in a better environment for effective communication [9]. The sensor node is composed of a radio transceiver connected with an antenna, an electronic interfacing circuit, a microcontroller, and a battery as an energy supplier. One of the most popular architectures of WSN is based on designing a master node to control other devices in the network. Among some of the existing challenges in sensor networks, we

are dealing with the lack of a trusted third party. As a result, a high number of malicious IoT devices have been used to create widespread malware. Financially, these growing threats are costing a significant loss of revenues for the largest companies [10].

In this paper, the suggested approach can verify the following requirements reliability, robustness, efficiency, security, and ergonomics. The WSN environment ought to be adapted to the user without making too much effort. In the technical aspect, we are motivated to provide a newly secured mechanism by taking the advantage of asymmetric encryption, proof-of-work, and the Blockchain network's transactions, representing the necessary enhancement for data validation. Furthermore, our approach allows users to be independent of any third-party or centralized architecture. Various consensus protocols are needed to validate the data to prevent and remove any duplicated entry or fraud. That is achieved using smart contracts, which allow users to set pre-conditions based on business protocol.

The main contribution of this work:

- We are considering a private Blockchain environment to construct a decentralized and secured peer-to-peer network model. This allows connected users to perform transactions using their permission Blockchain’s address as an authorization entity;
- We deploy the smart-contract to build a secured system where a Cluster-Header (CH) is dynamically designated from a WSN based-Blockchain;
- The simulations are conducted to evaluate the network transactions when the test scale of the balance of all nodes achieves a predetermined value of energy. Otherwise, the network transactions will be frozen.

The rest of the paper is organized as follows: Section 2 offers some backgrounds mentioned in this study. Section 3, focuses on the description of the system process details, characterization, and decentralized WSN. Section 4 presents the overview of the components of the methodology and the smart contract concept and their deployment process. Section 5 gives the implementation results and a few tested scenarios of the proposed approach and we provide the energy consumed by this method regarding those in the literature. Section 6 concludes this work by indicating some drawbacks of this proposed model and some feature works.

2. RELATED WORK

Securing WSN claimed to be one of the most interesting research because of their flexibility for problem-solving in different domains obviously for the potential ones, that why it is widely used and involved in many industries such as healthcare, E-commerce, pollution, science, marketing...etc [6]. Because of the sensibility of the sent information from the WSN nodes [11], it is necessary to protect any piece of data using a Blockchain-based solution in order to produce better results and to prevent attacks such as selfish mining and Denial of Service attacks [12]. A blockchain-based solution is suggested in [4] [13] [14] [15] by the authors for large-scale IoT devices. Software-

Defined Network (SDN) is used for making decisions and flexibility of attacks by constructing smart contracts using Ethereum technology in a wide range of networks and Blockchain. The provided model deals with the DDoS mitigation for searching the issues of the transfer of attack activities in decentralized networks. In [16], the authors proposed a new scheme to avoid malicious attacks using Blockchain and symmetric encryption to defend the data integrity and availability, although only symmetric aspect is considered their work while a cluster head selection is not conducted. In [5], a computational resource-sharing framework is investigated for D2D Network using Blockchain. In [14], the authors described recuperation of failed nodes based on Blockchain technology to recover nodes in terms of computational power, energy, and data storage in WSN looking for the state of cluster header by performing the security analysis. In [17], the authors presented a dynamic cluster head in the distributed architecture of the WSN approach in low energy cluster hierarchy (LEACH) aimed to increase energy consumption during the selection of the CH operation. In [18] [19], the authors proposed a new technique of cluster head selection in WSN using fuzzy decision-making by including energy consumption distance of nodes in the areas between neighbors nodes and base station in order to reduce the optimization complexity and decrease the energy consumption per node which consequently increase the network lifetime.

To address the drawbacks of existing approaches, we proposed the cluster-Header selection in WSN dynamically based on Blockchain technology using a smart contract to develop decentralized applications using the Ethereum environment. In fact, with all previously mentioned features, we can explore works in literature that collect for us much information including existing Blockchain applications as distinguished by written blocks, submitted transactions, deployed contracts, passed events, and logging information under logs. However, ensuring the stability, security against attacks, energy efficiency, and availability of cryptography mechanisms to avoid any modification of existing nodes due to tractability of each node in WSN is the aim of this work.

3. PROPOSED MODEL

3.1. *System Description*

In this section, we describe the general architecture of our proposed model. However, the idea applies the blockchain technologies to wireless sensor networks to increase the resilience of transactions with them. In order to prevent any kind of security issues [20].

This approach ensures the non-single control by any authority and improves the trustworthiness between parties by exploiting asymmetric cryptography system (private/public key encryption and decryption) to enhance the chances of any data immutability and arrange the way that data is transmitted between nodes over the network. Based on the Blockchain, our model is broken down into clusters and each cluster has various nodes, one of these nodes is a master one dedicated to receiving numerous data. The remaining nodes of a network (Cluster) will carry out multiple tasks at once; one of these tasks is sending the same piece of information to the Master node and mining it on the Blockchain directly. On the other hand, the master node will be

charged with checking the validity of the received information by comparing them with the stored ones on the Blockchain as shown in Fig.1.

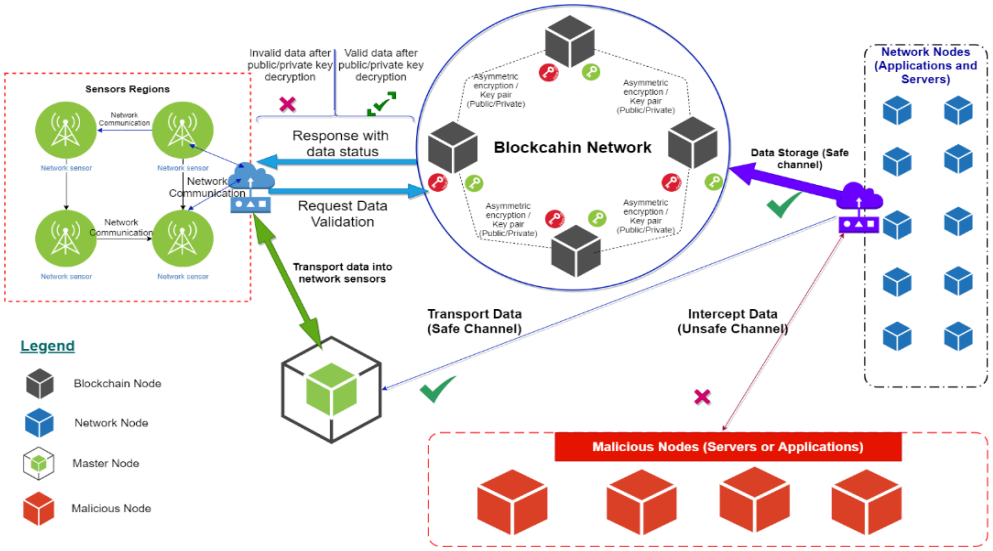

Fig.1. Suggested model to secure WSN based on Blockchain

3.2. *Formulation*

We consider an area composed of one sink node, n nodes with m clusters, and each cluster c is composed with x_c nodes and its cluster-header is denoted by CH_c as depicted in Fig.2. The energy of each node in a cluster is denoted by E_i^c and the minimal energy for the CH is denoted by E_{min}. Let $x_{i,j} = 1$ means that the node i belongs to cluster j else 0. The first estimation brings the top node because at the beginning energies are equivalent.

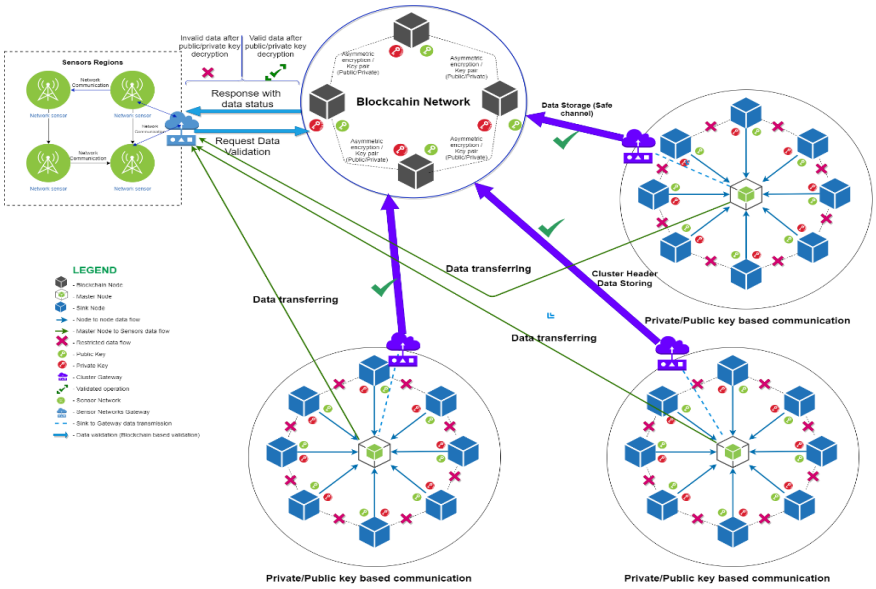

Fig. 2. Illustration of WSN Data Flow with Blockchain

After that, decreasing the energy is related to the time spent and the energy to behave and release demanded transactions over the network, defined as:

$$getEnergy(node.address) > max(E_1^c, E_2^c, \dots, E_{x_c}^c) \quad (1)$$

Besides the energy factor, the distance between the looped nodes and the Blockchain target is also calculated to prevent any additional costs while minimizing transactions over the network as demonstrated in the following formula:

$$node.distance \ < min(D_1^c, D_2^c, \dots, D_{x_c}^c) \quad (2)$$

Each node must exist in one cluster at most. So that, either a single node is found in one place or not. This means that:

$$\sum_{j=1,i}^{m} x_{i,j} \leq 1, i = 1, \dots, n \quad (3)$$

Taking the energy of each node as input and calculating the highest energy value of the network to decide which individual will be selected as a header for a given cluster to perform the execution of transactions, we verified:

$$E_{current} \geq E_{min} \quad (4)$$

A CH is privileged based on its residual energy which must exceed intensity conditions; the total energies of all nodes among the network and the minimal one (4) as well to mine the next block in the Blockchain. As long as the network keeps working, the traffic will proceed its way and the transactions must be validated and executed by the master node named Cluster-Header. Otherwise, freezing the transactions requires setting an end of the data flow over the chain. For this reason, we are assigning a variable named **transactionsEnd** which will be tested in the next appeal for a new transaction that determines if the operation is performed or not. If at least one node satisfies the energy condition, it will perform the next transactions. Otherwise, **transactionsEnd** will be set true. Therefore, **Algorithm 1** shows how the cluster head is selected dynamically based on their residual energy. Such as nodes with the largest residual energy amount are preferred to be cluster heads.

4. COMPONENTS OVERVIEW

4.1. *Sensor Nodes*

In the proposed methodology, we use sensor nodes n to monitor the activities of data transmitted with a high level of transparency and security powered by the Blockchain to secure peer-to-peer network and IoT-based applications allowing connected users to perform secured transactions using their Blockchain's address as network identifier. However, to choose the Cluster-Header (CH) that can transfer the information from neighbors (nodes) communicated data in the same cluster to the base stations, an election of the master node will start to pick the most trusted node that can handle all data and transfer it effectively. The transaction process will finish when the balance of energy and the distance between neighbors of all nodes achieves an already determined value so that the network will stop working to prevent any unexpected behavior from

all members and clusters of the network.

4.2.*Cryptographic Hash Function*

To guarantee the safety of the communication for transmitted data between the cluster head and the master node, Blockchain provides one of the most features in the security paradigm; the cryptography aspect of data encryption and decryption. More precisely, the hash function of asymmetric cryptography uses two main tools: a public key and private key Fig.3. However, in the Ethereum Blockchain, the sender of a transaction signs the transaction with his private key, and once it's broadcasted on the network, anyone can verify that the signature is valid with the sender's public key.

Algorithm 1 A CH selection method in WSN Based Blockchain

Data: Nodes energy;
Result: Decide Cluster-Header
Initialization: The main parameters
Step1:
- Read minimal required energy (E_{min});
- Calculate distance between nodes;
- Launch environment;

Step2:
while *all nodes energy are greater than the minimal value and data streaming beats alive* **do**
  Step3:
  Read current Cluster-Header energy ($E_{current}$);
  if $E_{current} > E_{min}$ **then**
    Step4:
    - Sleep main thread for a while to prevent system crash;
    - Perform the transaction with the current node;
    - Free up system memory;
  else
    Step5:
    Set *isChanged = false* ◁*to check whether the Cluster-Header is chosen or not*
    while *Cluster-Header not yet selected for next transaction block* **do**
      for *each node in the requested cluster* **do**
        if $getEnergy(node.address) > max(E_1^c, E_2^c, ..., E_{x_c}^c)$ *and* $node.distance < min(D_1^c, D_2^c, ..., D_{x_c}^c)$ **then**
          - Set current loop node as the Cluster-Header;
          - Free up system memory;
          - Break loop;
        else
      end
    end
    if *isChanged == false "Means no picked cluster-header"* **then**
      - Set transactionsEnd = true; "ie. freeze data flow over the cluster"
      - Free up system memory;
      - Sleep main thread for a while to prevent system crash;
      - Distribute final signal;
    end
    - Sleep main thread for a while to prevent system crash;
    - Perform the transaction with the new selected node;
    - Free up system memory;
  end
end

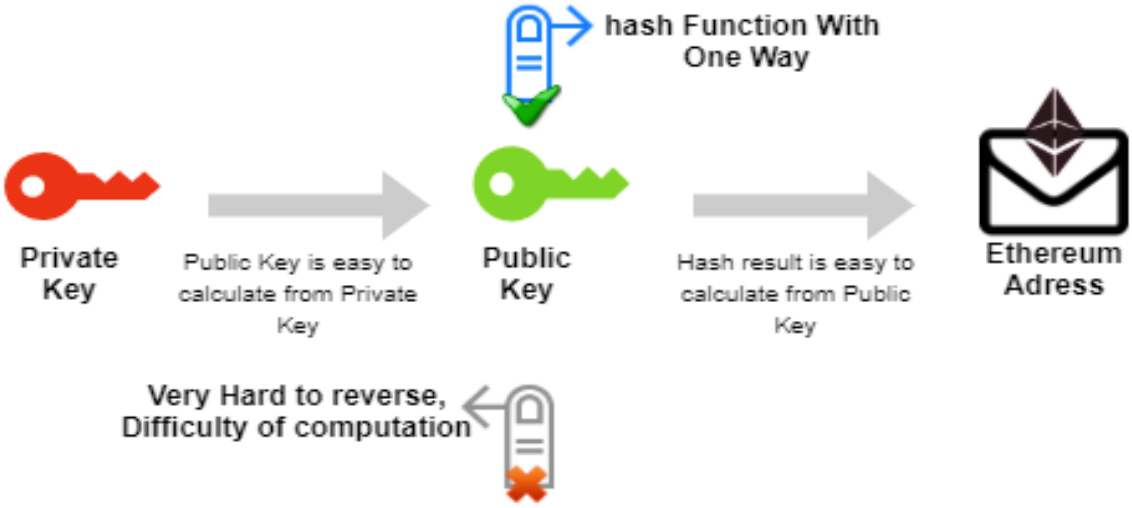

Fig. 3. Hash Function Computation

4.3.*Transaction Process*

Transactions mining is the main operation for the whole process, which when transferring data over the network we are manipulating security aspects of cryptography system such as private/public encryption with the aim of storing data respecting the written rules on the contract and storing data inside the network for later check form of all parties. This action has the goal of preventing any kind of modification or fraud thanks to the immutability principle. The details of the transaction can be summarized as shown in Table.1

Table 1. Transactions Details

Transaction Details of SCHD in WSN account nodes	
Sender Address	0xC39394c9898380a367E0870831169D6ba0Fd6951
Receiver Address	0xE2AE330Ba3B07C4b8882d3d1FCD449BB0055cf49
Mined in Block	37
Gas Used	26868
Date	July15, 2019, 3:22 P.M
Tx Data	0x0b9a4a2bdf966000a128da91166041ee6916fe0930a46f538ef5128730c0034c

4.4.*Block Validation*

The process of adding a new block in the Blockchain network is triggered when a new transaction is requested to be mined. For this reason, the source node broadcasts its public key to all other nodes in the same cluster to participate in mining activity. Afterward, nodes verify if it is a valid node using the consensus procedure. A reward will be given to the leader miner that can validate the block then the verified transactions are added to the network as shown in **Algorithm 2**.

Algorithm 2 Block Data Validation Process Algorithm

Data: PreviousBlock, CurrentBlock, SenderPublicKeyIdentifier
Result: Validity of Block for transactions mining

```
if not(CurrentBlockHash decides true computation through network series) then
    decision ← false;
end
if Node N payloader is validated across processing then
    decision ← true;
end
if CurrentBlock contains event payloader of node N then
    decision ← false;
end
if key encryption/decryption failed then
    decision ← false;
end
if all payloader received validation from all network members then
    decision ← true;
end
return decision
```

4.5.*Consensus Modality*

To ensure the correctness of smart contracts deployment and data storage, the consensus mechanism of Ethereum comes in the field to specify which nodes can be extended in the Blockchain within the network based on the concept of POW and his alternative POS which is defined as follow:

- ***Proof of Work (PoW):*** has been approved to play the role of the consensus algorithm. At that point, PoW becomes the most used mechanism (used by Bitcoin, Ethereum, etc) to validate transactions and avoid double-spending, but it still suffers from the problem of high-level consumption of electricity.
- ***Proof of Stake (PoS):*** another way of consensus algorithm which has the same purpose as PoW aims to validate transactions and avoid doubles pending without consuming energy. In a PoS Blockchain we don't have miners anymore, but something called validators. These validators vote on the next block, and

the weight of each validator's vote depends on the size of his stake.

- ***Hybrid Casper:*** Considered as Ethereum PoS implementation. The main Casper project is Casper FFG [21] that is a hybrid PoW/PoS consensus mechanism. This consensus means some blocks will be validated with the old PoW algorithm and some with the Casper protocol. So it will be a multi-step transition to introducing PoS for the Ethereum network.

4.6.*Smart Contract*

A smart contract is a computer protocol expected to automatically manage, verify, or require the negotiation or performance of a contract. Consequently, contracts ensure the performance of credible transactions without the need of the third parties; smart contracts render transactions traceable, transparent, and irreversible in WSN. The code and the agreements contained therein exist across a distributed, decentralized Blockchain network [8].

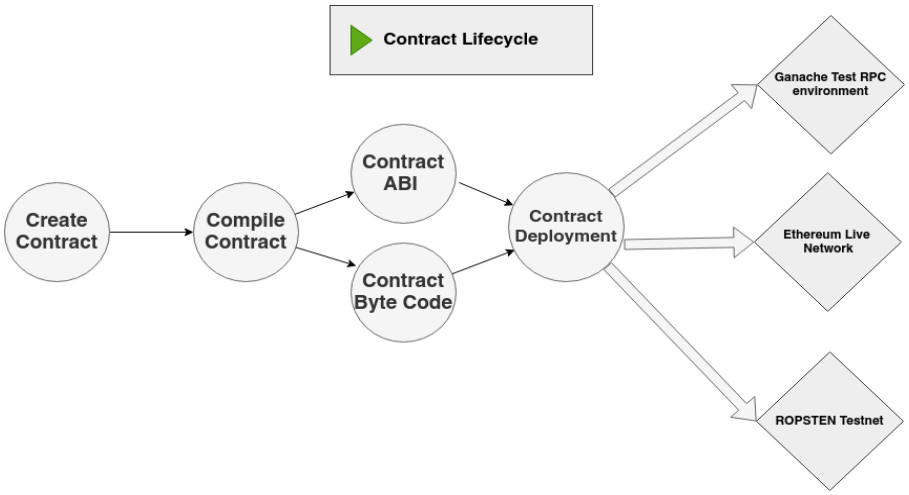

Fig. 4. Smart contract deployment process

In the beginning, one could consider a contract as a class in object-oriented terms, which is a template for objects. A contract may be deployed to a network multiple times, and each instance would have a distinct address, which could be used to interact with that particular instance of the contract at a later point. Therefore, each deployment of a contract could be considered as an object instance in oriented object concepts. Furthermore, each instance is independent and has its state (persistent data). The constructor of the contract is invoked when deploying a contract to the network, and that is the only time it is invoked as depicted in Fig.4.

Creating a smart contract of WSN requires knowing who is interacting with it at run time. In the Ethereum Blockchain, actors (smart contracts or wallets) are identified by their addresses. Storing addresses are used to handle the implementation logic depending on the transaction creator and its whole purpose as well. Destroying a

contract is considered as a separated phase that can be done using a pre-built function mainly called self-destruct. The smart contract life cycle is presented in Fig.5. Based on the desired business process, we will deploy our contract that contains a set of codes that satisfy our needs based on the application requirements.

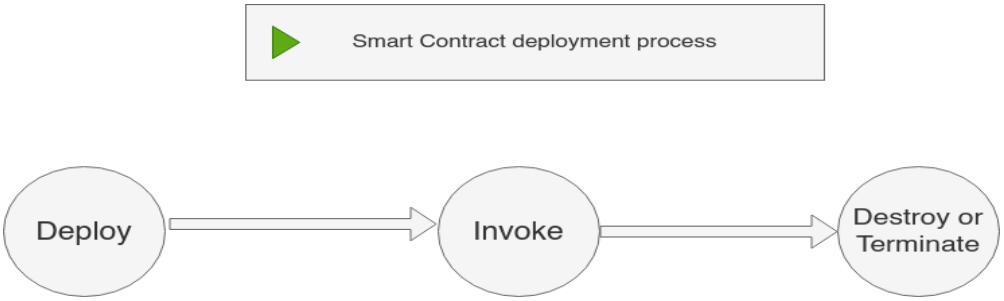

Fig. 5. Smart Contract life cycle

To follow the development process, we should retrieve the code of the smart contract, then compile it and get the application binary interface which will be used to interact with our automated (smart contract). In Table 2, we are exploring our different built functions that perform different actions.

Table 2. Contract functions signification

Functions	**Meaning**
getSinkNode()	Retrieve sink node from the Blockchain
sendData()	Send data from a node to another in the same cluster
sendDataToSinkNode()	Send data from a cluster-header to the sink node
getTransactionsCount()	Get number of transactions stored in the contract for a given node
getSinkNodeTransactionsCount()	Get number of done transactions by the sink node stored in the contract
getGlobalTransactionsCount()	Get number of transactions stored in the contract
getClusterNodesAddresses()	Retrieve nodes of a given cluster
getTransactionsDate()	Retrieve transactions date of a given node
getAllTransactionsDate()	Retrieve all transactions date
getTransactionDetails()	Retrieve a transaction details
getClustersCounters()	Get number of clusters

5. IMPLEMENTATION RESULTS

5.1. *Simulation Settings*

We define a set of dependencies that are needed for development purposes. The contract must be consistent with the business process of the web application to achieve satisfactory results. First, we need to install Python as an interpreter to deploy the smart contract developed by the programming language solidity [20]. As presented above, we described the parameters used to provide clarity on the work of the ecosystem:

- ***Django:*** is a Python-based free and open-source web framework, which follows the model-template-view architectural pattern. It is maintained by the Django Software Foundation. Django's primary goal is to facilitate the creation of complex, database-driven websites.
- ***ABI (Application Binary Interface):*** is necessary to specify which function in the contract to invoke, as well as get a guarantee that the function will return data in the expected format.
- ***Web3:*** is a collection of libraries that allow us to interact with a local or remote Ethereum node, using an HTTP, WebSocket, or IPC connection.
- ***Solc:*** solidity compiler is the special program that processes statements written in solidity programming language and turns them into machine language or "code" that a computer's processor uses.

The architecture as shown in Fig.6 details the different technical and theoretical aspects of the components and the integration between Blockchain networks (Ethereum) and Wireless Sensor Networks for data exchange security.

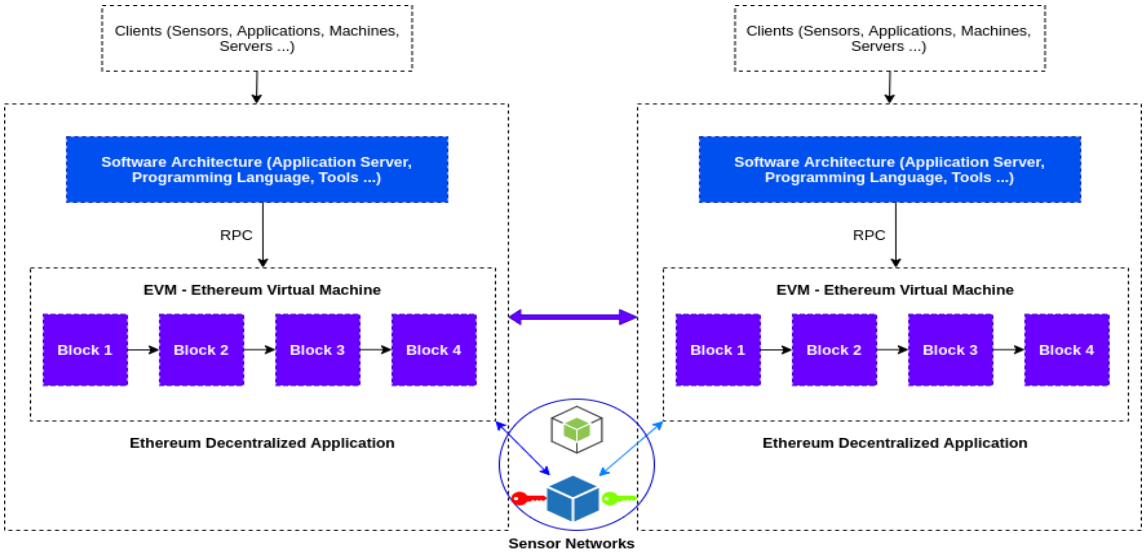

Fig. 6. Implementation Architecture**.**

The parameters values used in our contribution are detailed in Table 3.

Table 3. SCHD framework Parameters.

Parameters	**Values**
Total number of nodes	270
Size of the monitoring area	200m*200m
Number of Clusters	6
Balance	100 (Initial Value)
Block Hash	Ethash
Gas price	20000000000
Gas limit of blocks	6721975
Gas limit of transactions	90000

5.2.*Balancing Energy*

In this part, several tests of scenarios are investigated to ensure the proper functioning of the proposed model. We consider that the transactions are not directly stored in the contract but they construct a supply-chain called Blockchain. More precisely, the transaction goes through a series of steps before being completed. First of all, we need to check the current Cluster-Header (CH) by verifying its energy if it satisfies the minimum required one as specified in **Algorithm 1**. Knowing in advance the full network nodes has the advantage to proceed on a fully recognized chain by injecting necessary data in the contract itself without the need to recognize the network before each transaction. In the meantime, less time is consumed during any transaction thanks to the non-complexity activities. Not only had that but also to the number of clusters determined at the beginning. Typically, when we access a multi-nodes system, we access either a master node or a gateway node (default one). The CH is configured to be the start point for the jobs running on the network. When a user desires to log in or access the system, it is automatically prompted to log on to the primary node. In the Blockchain, a transaction is performed by the CH which is changed when its energy reaches a minimum balance leading to the end of the data transmission when all the nodes have a balance lower than the minimal one.

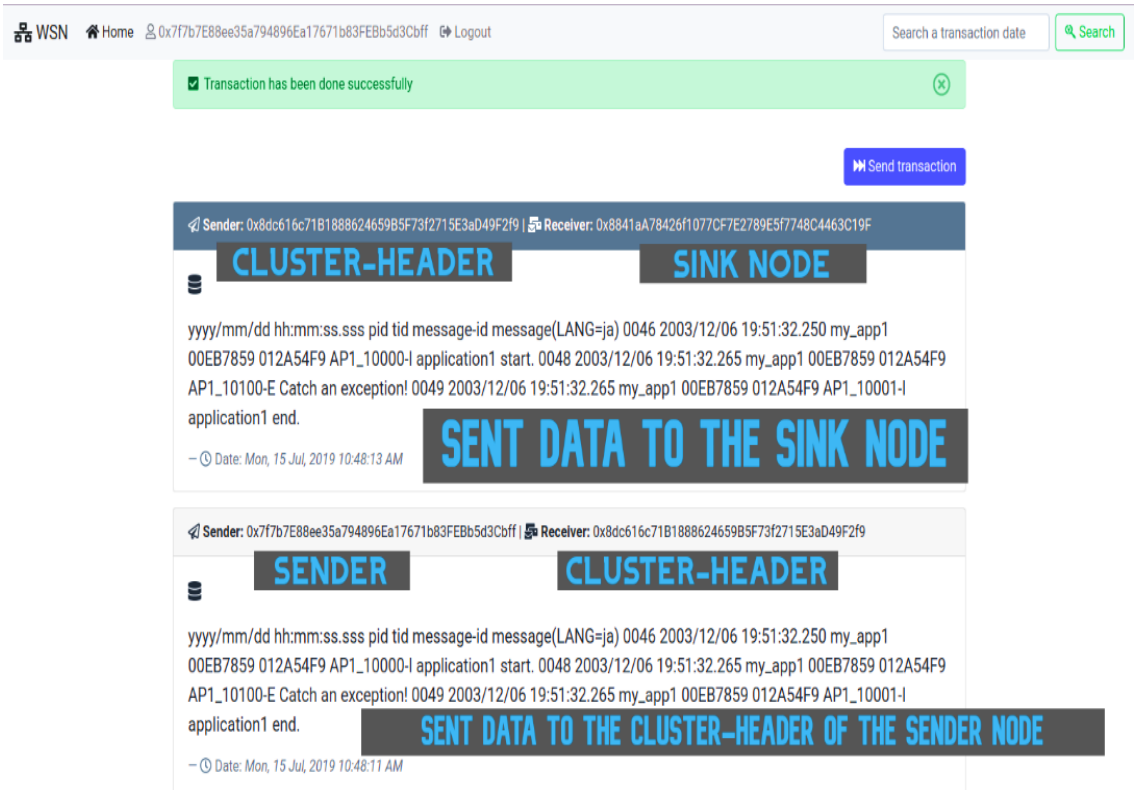

Fig. 7. Successful Data Transmission Scenarios.

Fig.7 shows several running test scenarios such as the selection of cluster header nodes, the sink node, and the transaction between nodes. The simulation was performed to ensure the dynamically functioning of the platform to execute a full scenario. As soon as the user (node) is connected with his Blockchain address, he will be able to perform a transaction in the deployment network by sending any kind of data. In the main context, sending data always takes the path to the CH which is chosen dynamically, then the CH itself will perform the remaining transaction to the sink node which receives data only from CHs of different clusters. The main goal consists of performing any transaction by the powerful and robust node of the network. This node in reality is chosen based on its energy level which must exceed the whole nodes which are placed in the same cluster.

The methodology of our model is very flexible on the variety of taking a node as a CH one. The reason for taking the powerful node from the network is based on the insurance and quality criteria, that's why in cluster1, cluster2, and cluster3 is shown in Table 4, we are taking the node with the largest amount of the residual energy (as shown in the last account address will be selected) to be responsible for transferring data from the cluster to its target. Checking these criteria is allowing us to take the advantage of system stability over time from any future changes, security between nodes, and availability even in critical situations.

Table 4. Cluster-Header Node Flexibility in Term of Energy.

Account address	Energy	TX Count	Index
Cluste1			
0x8841aA78426f1077CF7E2789E5f7748C4463C19F	96.90	76	0
0xE2AE330Ba3B07C4b8882d3d1FCD449BB0055cf49	99.48	68	1
0xC39394c9898380a367E0870831169D6ba0Fd6951	99.86	18	2
0x3474fA62123D4497257223e9169De2AA98862Ae9	99.98	3	3
0xA3747842C6e5fC5903C12b7098d87DfEc3bc293f	99.94	13	4
0x5288a3ccE3F45be0C374072dccAD829B9f8DFe91	99.99	1	5
Cluste2			
0x180524eD52a43b5d3e1E88dae42d486c122bd3BB	99.93	97	0
0x1C10dE035539a5cdb39C5BEcaF767f78adeF6844	99.94	94	1
0x0304E8A643bd5a8401dc0250db472a00706d5939	99.95	105	2
0x00426aE247e572Ab18318219E6c09a300F4bF31B	99.96	96	3
0xBE14431B101bE86987283834929ceFA03b862c16	99.98	121	4
0x4bd3CD24c12F9dc197582d82D87c4439cfC8733d	100	124	5
Cluster3			
0xf5cAF46A61Ef892Cabb7804bd3AC42b3e30e80E3	97.80	17	0
0xB78e22f1896bcA33c1003E8eCd73a958Ae85af50	98.95	43	1
0x802A15465a54f854O66b3D301503d6A0C0e4ACD4	99.90	20	2
0x182f7Cc7dAa77bc036cfB22a59D84089f0fG51D3	99.94	6	3
0x25327a33Ec5ce2067921085436Bc0a45539Oc9f8f	99.95	34	4
0x6B0a718F9c72b19c8Ec89B858c4C0a283e6F54cd	99.96	50	5

Fig.8 shows the energy-consumption state of the entire wireless sensor network. The comparative results of the energy consumption status, the LEACH was used *0.0347J*, the DDACM was used *0.0265J*, the RSSI cluster routing method was used *0.0248J*, and our proposed scheme was *0.0237J*. Thus, it were could be obtained result of energy consumption reduced compared with existing approaches of cluster head selection in the WSN network.

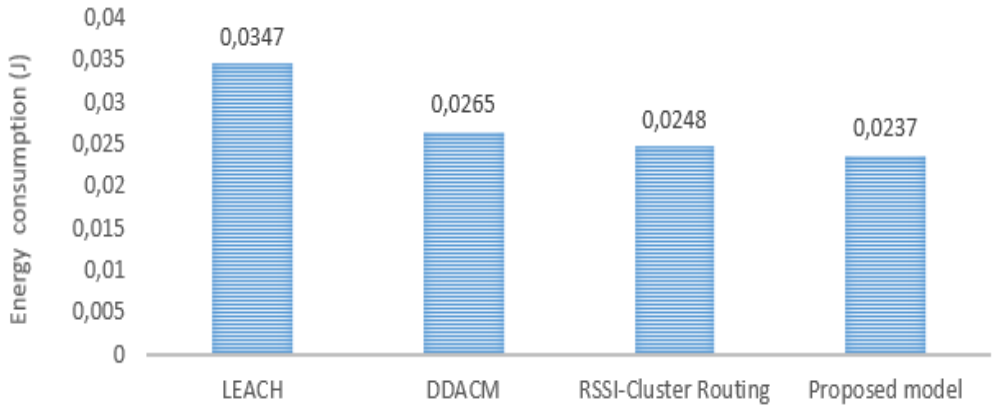

Fig. 8. Energy-consumption state of nodes for each approach.

5.3.*Cost Evaluation*

In this study, we estimate the cost of the creation of the smart contract as well as the execution of each function used in WSN. We carried out the experiment values when the gas price was set to *1Gwei*, where $1Gwei = 10^{9}wei = 10^{-9}$, and 1ether was equal to *172, 53USB* [22].

TABLE 5 illustrates the low cost of the running process of different functions in terms of some indicators such as the gas used Gas and the Fees used to construct our contracts. We observe that the highest-paid amount is corresponding to the deployment function only with 6.23569 USD. Nonetheless, it is only executed while setting up the system collaboration. As well as, the remaining functions use approximately an amount between 0.02 and 0.3 USD as a maximum paid amount.

Table 5. WSN Creation and Low cost contract functions.

Function	**Gas used**	**fee(ETH)**	**fee(USD)**
constructor	6700000	0.02747	$6.23569
getSinkNode()	21943	0.00009	$0.02043
sendData()	334048	0.0013696	$0.3109
sendDataToSinkNode()	332819	0.0013646	$0.30976
getTransactionsCount()	50213	0.0002059	$0.04674
getSinkNodeTransactionsCount()	57927	0.0002375	$0.05391
getGlobalTransactionsCount()	27421	0.0001124	$0.02551
getClusterNodesAddresses()	207357	0.0008502	$0.193

getTransactionsDate()	84317	0.0003457	$0.07847
getAllTransactionsDate()	43146	0.0001769	$0.04016
getTransactionDetails()	31384	0.0001287	$0.02921
getClustersCounters()	21902	0.0000898	$0.02038

6. CONCLUSION

In this study, we proposed a secured approach using the Ethereum (ETH) smart contract tool to choose the Cluster-Header (CH) for IoT application as in WSN. This approach can be integrated with the existing methods to designate a CH from a set based-Blockchain cluster in order to guarantee the security of data transmission between sensing nodes and the sink through cluster heads while ensuring low energy consumption compared with the existing approaches, in addition, a low-cost for deploying the smart contract is performed. Therefore, ensuring the system availability and the safety of database decentralization. Therefore, the implementation of the framework is deployed on the Ganache environment using solidity as a programming language to write smart contracts to reach and resolve both the flexibility and efficiency of proof of work for each node. Consequently, the implementation results of the proposed scheme with the existing approaches confirm that this study achieves integrity, agreement, validity, availability, and termination which make the whole system a promising solution by decreasing energy consumption and building low cost so that improve the architecture effectiveness and trustworthiness. As future work, we plan to consider the integration of our framework with others' processes of safety and apply it in a decentralized architecture for different networks than WSN as IoT, to empower and ensure the security and the privacy of the whole framework. In addition, we will set the necessary performance indicator of this approach based on the comparison with other cryptocurrencies like Bitcoin, Litecoin.

REFERENCES

[1] Sayadi, S., Rejeb, S. B., and Choukair, Z., 2019,"Anomaly detection model over blockchain electronic transactions," Proc.15th International Wireless Communications & Mobile Computing Conference (IWCMC), pages 895–900. IEEE.

[2] Al-Karaki, J. N., Gawanmeh, A., Ayache, M., and Mashaleh, A., 2019, "Dass-care: a decentralized, accessible, scalable, and secure healthcare framework using blockchain," Proc. 15th International Wireless Communications & Mobile Computing Conference (IWCMC), pages 330–335. IEEE.

[3] Cheikhrouhou, O. and Koubaa, A., 2019,"Blockloc: Secure localization in the internet-ofthings using blockchain," CoRR, abs/1904.13138.

[4] Misic, J., Misic, V. B., Chang, X., Motlagh, S. G., and Ali, M. Z., 2019, "Block delivery time in bitcoin distribution network," Proc. the International Conference on Communications (ICC). IEEE.

[5] Hong, Z., Wang, Z., Cai, W., and Leung, V., 2017, "Blockchain-empowered fair computational resource sharing system," Proc. the d2d network. Future Internet, 9(4):85.

[6] Nakamoto, S., 2019,"bitcoin: A peer-to-peer electronic cash system Technical report," Manubot.

[7] Riabi, I., Ayed, H. K. B., and Saidane, L. A., 2019, "A survey on blockchain based access control for internet of things," Proc. 15th International Wireless Communications & Mobile Computing Conference (IWCMC), pages 502–507. IEEE.

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

[9] Kulkarni, K., Sanyal, S., Al-Qaheri, H., and Sanyal, S., 2009, "Dynamic reconfiguration of wireless sensor networks," IJCSA, 6(4):16–42.

[10] Abdullah, K. M., Houssein, E. H., and Zayed, H. H., 2018, "New security protocol using hybrid cryptography algorithm for wsn," Proc. 1st International Conference on Computer Applications & Information Security (ICCAIS), pages 1–6. IEEE.

[11] Bloch, M., Barros, J., Rodrigues, M. R. D., and McLaughlin, S. W., 2008, "Wireless information-theoretic security," IEEE Transactions on Information Theory, 54(6):2515– 2534.

[12] Bai, Q., Zhou, X., Wang, X., Xu, Y., Wang, X., and Kong, Q., 2019, "A deep dive into blockchain selfish mining," Proc. International Conference on Communications (ICC), pages 1–6. IEEE.

[13] Makhdoom, I., Abolhasan, M., Abbas, H., and Ni, W., 2019, "Blockchain's adoption in iot: The challenges, and a way forward," Journal of Network and Computer Applications, 125:251–279.

[14] Noshad, Z., Javaid, A., Zahid, M., Ali, I., Javaid, N., et al., 2019, "Node recovery in wireless sensor networks via blockchain," Proc. International Conference on P2P, Parallel, Grid, Cloud and Internet Computing, pages 94–105. Springer.

[15] Kushch, S. and Prieto-Castrillo, F., 2018, "A rolling blockchain for a dynamic wsns in a smart city," arXiv preprint arXiv:1806.11399.

[16] Guerrero-Sanchez, A. E., Rivas-Araiza, E. A., Gonzalez-Cordoba, J. L., Toledano-Ayala, M., and Takacs, A., 2020, "Blockchain mechanism and symmetric encryption in a wireless sensor network Sensors," 20(10):2798.

[17] Gupta, K., Goyal, A., and Tripathi, A. K., 2015, "A novel approach for cluster

head selection in wireless sensor network," Proc. International Journal of Computer Applications, 113(16).

[18] Azad, P. and Sharma, V., 2013,"Cluster head selection in wireless sensor networks under fuzzy environment," ISRN Sensor Networks, 2013.

[19] Islam, N., 2018, "Towards a secure and energy efficient wireless sensor network using blockchain and a novel clustering approach".

[20] Spathoulas, G., Collen, A., Pandey, P., Nijdam, N. A., Katsikas, S., Kouzinopoulos, C. S., Moussa, M. B., Giannoutakis, K. M., Votis, K., and Tzovaras, D., 2018, "Towards reliable integrity in blacklisting: Facing malicious ips in ghost smart contracts," In Innovations in Intelligent Systems and Applications (INISTA), pages 1–8. IEEE.

[21] Buterin, V., Reijsbergen, D., Leonardos, S., and Piliouras, G., 2019, "Incentives in ethereum's hybrid casper protocol," Proc. International conference on blockchain and cryptocurrency (ICBC), pages 236–244. IEEE.

[22] Abou El Houda, Z., Hafid, A., and Khoukhi, L., 2019, "Co-iot: A collaborative ddos mitigation scheme in iot environment based on blockchain using sdn," Proc. Global Communications Conference (GLOBECOM), pages 1–6. IEEE.