Performance Evaluation of a Secured Framework for IoT Based on BlockChain

Ali H. Ahmed*, Nagwa M. Omar, and Hosny M. Ibrahim
Information Technology Department, Assiut University, Egypt
Email: {ali.hussein, n_omar, hibrahim} @aun.edu.eg

***Abstract*—**In the last era the number of internet-connected devices surpassed the human population. IoT integration rate into human world equals at least five times the rate of electricity and telephony. Currently in 2020 the number of IoT devices is around 50 billion smart objects. This great invasion to our live requires extensive efforts for controlling and securing those devices. BlockChain (BC) is a distributed write-only ledger that eliminates the need for third party in securing and verifying transactions between peers. BC is considered the most powerful technique for securing transactions between IoT devices. In this work, a robust and scalable blockchain-based security framework for IoT is proposed. This framework comprises clients, device gateways, and administrators. IoT devices access BC through gateways. Ethereum BlockChain is utilized in addition to Ethereum smart contracts for enforcing a set of rules defined by the system administrator. Finally metrics that fulfill both efficiency and effectiveness of the proposed framework are introduced. In the results section, the proposed work provides robust and scalable security framework for the IoT devices under different attack probabilities in addition to satisfying the conditions of lightweight, transparency, and timeliness.

Index Terms—IoT, BlockChain, smart contracts, security, fog computing

I. Introduction

The past years witnessed an unprecedented research focus on Internet of Things (IoT). The broad utilization of IoT in various applications motivates researchers to improve its quality in terms of power, security, and user convenience. IoT can be briefly defined as dynamic global network infrastructure with self-configuring capabilities based on standard and interoperable communication protocols. A set of IoT conceptual designs and applications are exposed in [1]-[4]. Most IoT definitions agree with that its simple embedded sensors connected to the internet and governed through a set of protocols. IoT consists of an enormous number of objects connected to achieve different objectives according to the application context. Objects can include simple home sensors, medical devices, nuclear reactors, and other things. The rapid and wide spread of IoT technology requires the deployment of strong and robust security framework to prevent security violations. Deploying a robust and secure framework for IoT has many challenges, which are concluded in the following points:

a) **Scalability**: IoT devices are increasing rabidly. Security techniques and services must cope with this exponential increase.
b) **Heterogeneity and Resource Limitedness**: IoT devices and communications networks are heterogeneous, which makes the ordinary and legacy security protocols, techniques, and services not suitable to all devices. In addition, resource limitation impedes the implementation of powerful security techniques on top of IoT devices.
c) **Transparency**: Secure framework must hide complex details from users with the capability of having silent deployment and as"plug and play" as possible.

Fig. 1. An example of BlockChain of a genesis block followed by three blocks (Block 1, Block 2, and block 3)

BlockChain (BC) [5], [6] is a persistent timestamped log of records or transactions grouped into blocks. A block is a data structure, which brings together transactions for inclusion in the blockchain. A block contains the number, version, hash of previous block, Merkle root, timestamp, nonce, transaction count, and signed transactions followed by a long list of transactions. A block can be identified in two ways, either by referencing the block hash, or through referencing the block height. The block header consists of three sets of block metadata. Meta-data is data that provides information about other data. A reference to a previous block hash, which connects this block to the previous block in the blockchain. The second set of metadata relates to the mining competition; namely the difficulty, timestamp and nonce. The third piece of metadata is the Merkle Tree root; a data structure used to summarize all the transactions in the block in an efficient manner. A genesis block is the first block of a blockchain. It is always hard-coded into the software of the applications that utilize its BC. Early versions considered it as block number 1, also be counted as block number 0. The genesis block is considered a "special" block as it does not refer to a previous block. Fig. 1 and Fig. 2 depict the concept. The power of BC is in the ability to eliminate the centralized authorities in the network, in addition to its

Manuscript received August 1, 2021; revised December 15, 2021.
doi:10.12720/jcm.17.1.1-10

write-only and tamper-proof nature. BC is maintained by nodes each of which has a copy of the entire blocks in the BC network.

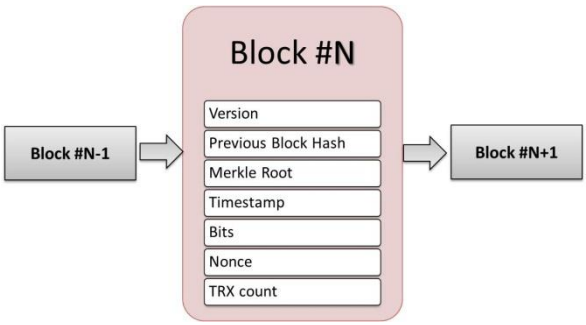

Fig. 2. Block anatomy.

In general, IoT and BC share a set of characteristics [2], which can be categorized into (i) technical and (ii) non-technical. This study sheds the light on the starting year and number of research work on both IoT and BC which realizes that both fields are trending and figures out a new era of highly secured objects.

This paper contributes a secure BlockChain-based framework for monitoring applications in IoT is proposed. The main entities of the framework are a system administrator, user, and IoT devices. A system administrator configures IoT devices and defines a set of access rules for these resources. After the setup phase, a smart contract is deployed into the BC via device gateway to define and manage user access for current resources. An initial set of commands that are commonly associated with environmental and healthcare monitoring applications are introduced. Each command is associated with a smart contract. The commands are:

a) **LOG_CMD:** This command is for informing specific IoT device to transmit the timestamped values associated with its sensors. Usually an IoT device is equipped by various types of sensors.
b) **STOP_LOG_CMD:** This command used to stop the transmission of specific sensor logs.
c) **AUTO_ACT_CMD**: Sets threshold to the sensed values for automatic actuating such as activating fans or A/C if specific temperature value is reached.
d) **ACT_CMD**: An Admin may use this command to actuate on the environment through the IoT node regardless the threshold value.

Transactions are considered an execution of specific commands performed by either users or admins of the IoT in small scope. In this work, gateways are treated as minors and can verify transactions in BC. While the BC is public and anyone can access the gateway, the gateway allows only authorized user to access the IoT device as enforced in the smart contract. The BlockChain stores the transactions on blocks and close it through a proof-of-work. The block is then attached to the BlockChain. In order to test the proposed architecture, a private Ethereum testnet is used in addition to system performance measurement.

The paper is organized as follows. Section 2 provides an overview of existing secure frameworks in IoT and discusses their main problems. The details of the proposed protocol building blocks and their interactions are discussed in section 3. Section 4 presents the experimental results performed to test the proposed framework performance. Finally, the paper is concluded in section 5.

II. Review

A. IoT Security Frameworks

The number of IoT devices surpassed earth's population [7]. This huge number of devices motivates current researchers to develop techniques and protocols for managing all aspects related to IoT. Generally, IoT has three main layers architecture [3] consisting of: (i) Perception (sensing device domain), (ii) Network (networking domain), (iii) and Application layers (application domain). Each IoT layer is susceptible to either active or passive threats or attacks. These attacks can originate from external sources or internal network. The main aspect which gained great focus is developing scalable and secured frameworks for IoT applications.

In [8], the authors proposed the SmartOrBAC, an authorizations access model built around a set of security and performance requirements. The model adds enhancements to the current (Organization-Based Access Control) OrBAC model. These enhancements adapt the current OrBAC model to the IoT environment. The authors designed abstraction layers to hide IoT specifications. They also provided a case study for SmartOrBAC in IoT. Although the model works well on an Arduino Mega2560 board which is constrained device, there is no quantitative evaluation provided to examine the model performance.

The authors in [9] introduced an intelligent framework for IoT. This framework is based on cryptographic techniques. An asymmetric end-to-end encryption technique is used to share session key between IoT nodes. The shared session key is used to encrypt further messages between nodes. This framework has many inherited advantages, such as utilizing asymmetric cryptography, eliminating eavesdropping, DoS, and quantum attacks. The framework, however, does not scale to billions of devices that have to share key before communications.

Recently, many secured frameworks were proposed for IoT. These frameworks were introduced to support rapid IoT applications development. Representative examples include: AWS IoT (Amazon), ARM Bed (ARM and other partners), Azure IoT Suite (Microsoft), Brillo/Weave (Google), Calvin (Ericsson), Home Kit (Apple), Kura (Eclipse), and SmartThings (Samsung). We recommend interested readers to read [10]. Interested readers can find more examples of IoT secured architectures in [11].

B. BlockChain-Based Secured Frameworks

IoT technology replaces the centralized structure with a complex network of decentralized devices. Also is has

unique characteristics which hider the development of secured framework to its devices. BlockChain have the solutions to these current obstacles in setting up this secured framework. Quantitatively the following paragraphs show a set of IoT problems and how BC can solve them.

a) Cost and Capacity Constraints: This is how to handle exponential growth of IoT devices with minimum cost. The BC solution for this issue is that no need for powerful and **centralized** entity; devices can communicate directly and securely, exchange value with each other, and execute action automatically through smart contracts.
b) Deficient Architecture: Each block of IoT architecture acts as a bottleneck or point of failure and disrupts the entire network; vulnerability to distributed denial of-service attacks, hacking, data theft, and remote hijacking also exists. This problem can be solved by secure messaging between devices, verify the validity of device's identity, and cryptographically sign and verify transactions to ensure that only message's originator could have sent it.
c) Cloud server downtime and unavailability of services: Cloud servers are sometimes down due to cyber-attacks, software bugs, power, or other **problems**. Using BC no single point of failure and records are cloned to many peer nodes.
d) Susceptibility to manipulation: Information is likely to be manipulated and put to inappropriate uses. The key feature in BC is decentralized access and immutability so, malicious actions can detected and prevented. Devices are interlocked: if one device's BlockChain updates are breached, the system rejects it

Not only BlockChain helps in overcoming the major problems in IoT but also it has many advantages as well:

a) Tamper proof of data.
b) Robust and highly reliable.
c) More private data.
d) Records the historic actions.
e) Elimination of single control authority.
f) Cost reduction in developing huge internet infrastructure.

During the recent years, many frameworks for IoT based on BC are introduced but till now no clear implementation can be utilized in the real world. For example the framework in [12] utilized the BlockChain for smart houses application and introduced four layers architecture:

a) Physical Layer: Smart houses contain many sensory devices that collect and transfer data to the other layers. Many of the smart **devices** (e.g., security cameras) are vulnerable to security attacks as they lack in access control mechanism and encryption.
b) Communicati**o**n Layer: In this layer, smart devices use different communication protocols (e.g., Bluetooth) to exchange data among different devices. To provide security and privacy during data transmission, the blockchain protocol needs to be integrated with the communication layer. This integration is challenge as the requirement can vary from application to another.
c) Database Layer: Block chain is a distributed ledger, which is a type of decentralized database that stores and records received data one after the other. Each record in the ledger has a time constrain and a unique cryptographic signature. The history of the ledger can be verified by a permissible user. There are two different type of distributed ledger in practice: i) permissionless and ii) permissioned. Since the public ledger is prone to anonymous attacks, it is advisable to use permissioned to ensure security, scalability and performance for real time objects.
d) Interface Layer: This layer contains number of devices that communicate with each other and transfer data. For example, controlling a refrigerator or accessing home security cameras through a mobile phone device. The major thing to keep in mind is that the applications or the devices must be integrated carefully such that they do not give access to the intruders.

The main disadvantages of this framework are that it does not produce sufficient details regarding minors locations in the framework. Also the literature does not provide any information about how the access roles to smart devices are enforced. Finally, the provided architecture framework is theoretical with no actual realization and validation against any security model.

In [13], a PKI for IoT devices is deployed based on an Ethereum framework [14]–[16]. This technique used a smart phone and three raspberry PIs as a proof-of-concept. A smart phone is used to setup the policy via a smart contract. Then, the configurations are deployed on Ethereum. The policy contains a threshold value which is used to trigger power saving mode on. The Raspberry PIs are considered as IoT devices (light bulb, air conditioning, and meter) such devices also have an Ethereum account. Another contract is used for updating meter values into the BlockChain. The light bulb needs to retrieve values from both meter contract and policy contract. Once values are received from meter contract, the validity of them checked using a public key and a signature. RSA algorithm is used with a smart phone and a meter to keep their secret keys. In the case of having electricity surpasses policy while periodically retrieving values, devices simply switch to the energy saving mode.

Although the technique have many advantages, it overused encryption. Authors attempt to include extra RSA-based PKI in smart contracts during either publishing the reading meter values or setting up the policy. Ethereum accounts have an implicit implementation of public key cryptography. The only contribution is the utilization of BlockChain as an access control for devices.

A Smart Home System (SHS) was proposed in [9]. This system is based on a private Ethereum BlockChain. The

SHS extended the work in [13]. The framework includes a home minor and a temperature sensor. Transactions used in the SHS can be divided into three categories, which are: (i) monitor, (ii) store, and (iii) access. A set of policies, which are defined by home owner, can be enforced by deploying smart contracts into BC. The frameworks' main shortage is that it handles only three types of transactions (i.e., monitor, store, and access). Another weakness is that the details of smart contract and how policies are enforced were not clearly mentioned.

The main shortages in recent secured frameworks which is based on BC is that no clear and detailed plan for implementing the framework in real world. Also these frameworks didn't provide any quantitative and performance analysis. During this paper a secured framework based on BC for IoT is proposed and its implementation details are introduced. The literature contains a PoC for the proposed work using a set of hardware and software components. A mathematical analysis for the framework is presented for both the frameworks' throughout and delay.

III. The Proposed Secure Framework

A. The Framework Architecture

For the best of our knowledge there is now standardized protocol stack for IoT. Accordingly, we include the most commonly existing layers in IoT architectures. The BC secured ledger is combined to the proposed architecture to secure messages exchanged between IoT devices and entities and to keep historical record from the environmental value updates. the proposed framework architecture is depicted in Fig. 3. The framework is composed of five layers (Layers 1-5). The purpose of each layer can be described as follows:

a) **Layer 5 (Application Layer):** This layer contains the applications themselves built on top of the IoT in different domains such as health, agriculture, industry, and utilities.

b) **Layer 4 (BlockChain Layer):** This layer implements the logic for the use of the BC technology for secure transactions and accessing history for monitoring application. It is responsible for receiving, storing, and displaying the transmitted transactions. It also performs PoW calculations for every created block. Many BC frameworks available today such as Ripple, Ethereum, Quorum, and R3.

c) **Layer 3 (Communication Layer):** This layer includes the communication modules to transmit the data to the Internet. The technologies used in this layer includes WiFi, LoRa, Zigbee, and Cellular (3G or 4G).

d) **Layer 2 (Sensor Management Layer):** This layer includes the processing node which receives data from the sensors and performs pre-processing steps. These preprocessing steps can include filtering, analog to digital conversion, data aggregation, and compression. These steps are performed before transmitting the data over the Internet. The node usually has appropriate processing capabilities—for example, it may be a microcontroller, Arduino, Raspberry Pi or a smart phone.

e) **Layer 1 (Sensors Layer):** This layer includes the devices (i.e., "things") to be connected to the Internet. It is expected to be the largest and the most heterogeneous one. This layer can include several devices, such as sensors and actuators. It is responsible for extracting the necessary information needed to be transmitted over the Internet or performing the required action received form the controlling entity.

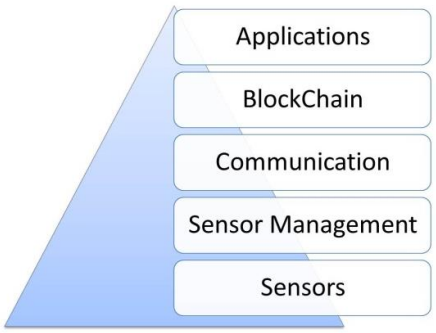

Fig. 3. An overview of the proposed framework architecture.

B. Blockchain Utilization

Nodes in the BlockChain paradigm keep the network running by participating in the relay of information. This can be achieved by downloading the suitable framework for the target application. Special set of nodes called minors groups transactions into blocks and add them to the BlockChain by solving a complex mathematical puzzle. The puzzle is finding a nonce that, when combined with a specific data in the block and passed through a hash function, produces a result in a certain range. The nonce is found by guessing at random. The hash function makes it impossible to predict what the output will be. So, minors guess the mystery number and apply the hash function to the combination of that number and the data in the block. The resulting hash has to start with a pre-established number of zeroes. Let the currently guessed nonce by specific minor is, *N*, and , *T*, is a timestamp, *P* is the target block hash value, and *prvblk* is the hash of the previously mined block, then the mining problem is described by Equation 1.

$$P \geq h(prvblk|N|T) \quad (1)$$

The first minor who gets a resulting hash within the desired range announces its victory to the rest of the network. All the other minors immediately stop work on that block and start trying to figure out the mystery number for the next one. As a reward, the victorious minor gets some sort of reward. The difficulty of the calculation (the required number of zeroes at the beginning of the hash string) is adjusted frequently, to cope with the networks' nodes processing power.

Currently there is no a feasible and low-cost technique to hack blockchain. Let an attack is held to generate an

alternate chain of blocks faster than the honest chain. Even if this is accomplished, it does not make the system easy to be hacked, such as logging events which may not actually sensed by the IoT devices. Minors will not accept this invalid updates. An attacker can only try to change one of his own transactions to inject environmental values that don't reflect actually the environment.

C. The Proposed Framework Security Analysis

Let the original transmitted message is, *M*, where, *M*, contains nodeID, timestamp, and value as in equation 2. In this case the home minor appends and transmits extra transaction to The to the Ethereum BC. Let an Intruder introduces, *M'*, where *M'* contains fully or partially modified information as in Equation 3:

$$M = NodeID|Timestamp|Value \quad (2)$$

$$M' = NodeID'|Timestamp'|Value' \quad (3)$$

Both the messages will be broadcasted to the BC network, *M* and then *M'*. The Home minor in addition to and other minors after reaching specific number of transactions introduce a Block *Blk* which contains:

$$Blk = ver|MerkleRoot|Timestamp|PrevHash|(M_0|M_1|..\,|M_n) \quad (4)$$

The fake block, *Blk'*, which contains *M'* will have the same sequence of operations as *Blk*:

$$Blk' = ver|MerkleRoot|Timestamp|PrevHash|(M_0'|M_1|..\,|M_n) \quad (5)$$

In PoW, minors aim to find nonce value which satisfy Equation 1. The BC uses this consensus mechanism to identify the forged block Blk0 or path of blocks and it is finally discarded. In order such attack to be succeed, intruder must gain not less that 51% of the BC nodes which means controlling the majority users and taking over most of the mining power which can't happen.

In The replay attach intruder attempts to resend valid packets to obtain the same effect twice, such as triggering alarms by resend packets contains values greater than preset threshold. In this case attacker has to modify the packet time, this goes the same as interception attack discussed earlier.

Binomial Random Walk mathematical model can be used in characterizing the race between the honest chain and an attacker chain. The success event is described as the case which the honest chain being extended by one block, and the failure event is the attacker's chain being extended by one block. The probability of an attacker catching up from a given deficit is analogous to a Gambler's Ruin problem. Suppose a gambler with unlimited credit starts at a deficit and plays potentially an infinite number of trials to try to reach breakeven. We can calculate the probability he ever reaches breakeven, this is similar to the probability that an attacker ever catches up with the honest chain. Let, p_h, is probability an honest node finds the next block, p_a, is probability the attacker finds the next block, and, p_{az} = probability the attacker will ever catch up from z blocks behind, p_{az}, can expressed as:

$$p_{az} = \begin{cases} 1 & p_h \le p_a \\ (p_a|p_h)^z & p_h > p_a \end{cases} \quad (6)$$

Given our assumption that $p_h > p_a$, the probability is reduced exponentially as the number of blocks the attacker has to catch up with increases. Let an IoT node reported specific value through the LOG_CMD and a transaction has been added to a block and, *z*, blocks have been linked after it. Attacker doesn't know the exact amount of progress the attacker has made, but assuming the honest blocks took the average expected time per block, the attacker's potential progress will be a Poisson distribution with expected value:

$$\lambda = z.\frac{p_a}{p_h} \quad (7)$$

To get the probability of an attack is done if invalid block is appended over, *z*, block, p_{az}, we multiply the Poisson density for each amount of progress he could have made by the probability he could catch up from that point:

$$p_{az} = \sum_{k=0}^{\infty} \frac{\lambda^k e^{-\lambda}}{k!} \begin{cases} 1 & k > z \\ (p_a|p_h)^{z-k} & k \le z \end{cases} \quad (8)$$

By trimming insignificant and in finite values from equation 8, the equation turns into an implementable version:

$$p_{az} = 1 - \sum_{k=0}^{z} \frac{\lambda^k e^{-\lambda}}{k!}(1 - (p_a|p_h)^{z-k}) \quad (9)$$

The results section shows the value of, p_{az}, after programming equation 9. Curve is depicted to show that, p_{az}, is reducing with increasing the number of blocks needed to be caught up by an attacker, z, for different values of, p_a. In this proposed work, Ethereum [17] testnet BlockChain framework is utilized for proof-of-concept scenarios [18]. A contract Ethereum account is created to be used for deploying smart contracts. A smart contract is used to define a set of access policies and deployed by an administrator. Remix tool [19] is used in coding and debugging smart contracts in Solidity programming [20]. The following subsections describe the detailed operation of the proposed framework.

D. Proposed Framework PoC

The proposed framework utilizes Ethereum BlockChain in enforcing smart contracts deployed by system admin. Each set of IoT devices belongs to either a specific enterprise or user are attached to a single minor. The minor is a full power PC and acts as a starting point for the smart contracts deployment process. Each IoT device has a connection to the minor. This connections is used as the controlling connection for the IoT device. Users can access IoT devices by sending requests to execute specific

smart contract created and deployed by an admin. The flowchart in Fig. 4 shows the sequence of transactions for the LOG_CMD contract. Setting a threshold and unconditional actuation ACT_CMD sequence of transactions are the same as the shown transactions in Fig. 4, but with different smart contracts.

Automatic actuation requires an IoT device to invoke AUTO_ACT_CMD smart contract, which calls the Threshold contract to check whether the updates of an IoT node are greater than the preset threshold value. The sequence of transactions associated with this command is shown in Fig. 5. In order to examine the proposed work performance, we designed a proof-of-concept for the entire framework. The following few subsections describes the components and communications in the proposed proof of concept.

1) Framework PoC components

The main components and tools of this proof-of-concept design for the proposed framework are:

- Minor: HP PRODESK PC Core i7, 16GB RAM, and 1 TB Hard Disk.
- IoT Devices: 6 Arduino Mega 2560 Rev3 [21]. Arduino Mega is based on the ATmega2560, which is a powerful 8-bit AVR RISC-based microcontroller [22]. Arduino platform was chosen, because it presents an easy-to-use introductory programming environment that suits basic educational purposes. The specific Mega 2560 board was chosen, because its ATmega2560 controller is rich in resources. It has multiple USART communication modules that are used to connect with the wireless communication modules. Figure 6 shows the Arduino ATmega2560 Shield. The componets of the proposed shield used as an IoT node in out PoC are:
    1. Arduino mega2560 Board
    2. IoT Arduino Shield
    3. Xbee-S2 Zigbee Module
    4. ESP-12E WIFI Module
    5. Logical Converter
    6. DHT-11 Sensor
- Ethereum: The utilized BC framework is Ethereum which provides rich capabilities for implementing BC-based applications, such as testnet, smart contracts deployment, and enforcement.
- Remix: An online tool [19] for writing and debugging smart contracts using solidity programming language.

E. Communication

Three IoT devices out of six act as parent nodes, they directly deliver their data to the minor through WIFI module in addition to performing some sort of aggregation to the other child nodes. The other three child nodes use their Xbee-S2 Zigbee module to deliver their data to specific nearby parent IoT node. Fig. 7 shows the communication topology between the six IoT Nodes and the minor. IoT nodes either parent or child is sending updates to the BC repository. The updating message contains nodeID, timestamp, and the value of either temperature or humidity captured by the DHT sensor in the IoT proposed shield. The messages contains NODEID, T IMESTAMP, and the V ALUE sensed by an IoT node. The three fields are delimited and encoded in binary.

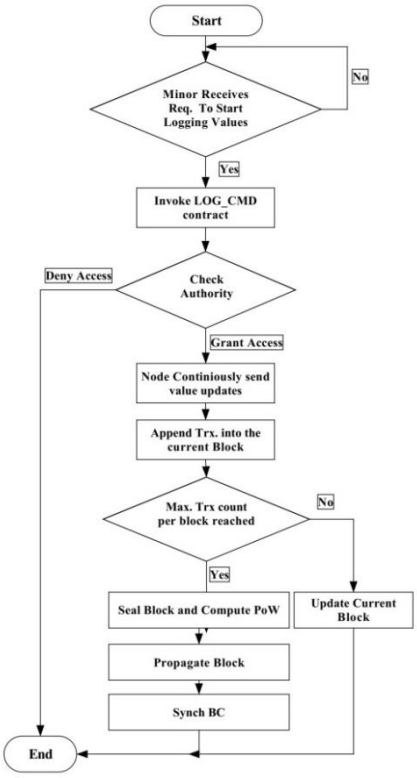

Fig. 4. Sequence of transactions for the LOG_CMD Command

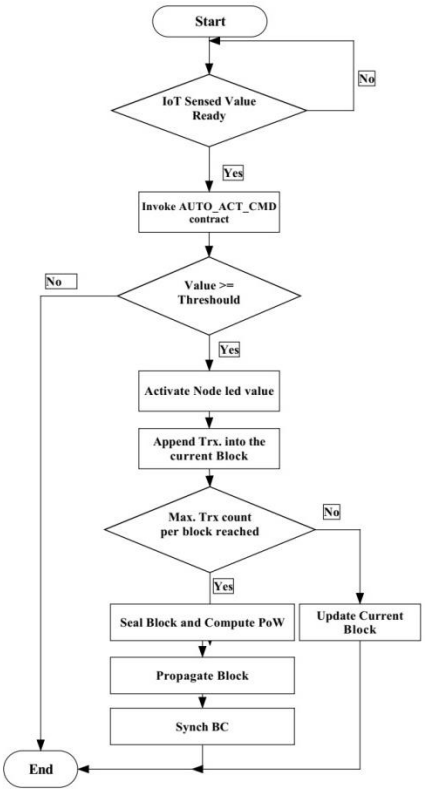

Fig. 5. Sequence of transactions for the automatic actuation Command

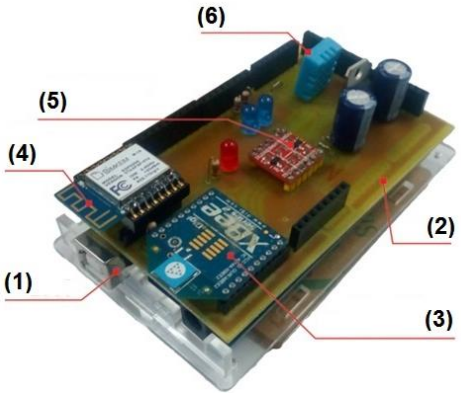

Fig. 6. An arduino mega 2560 Rev3 proposed shield components.

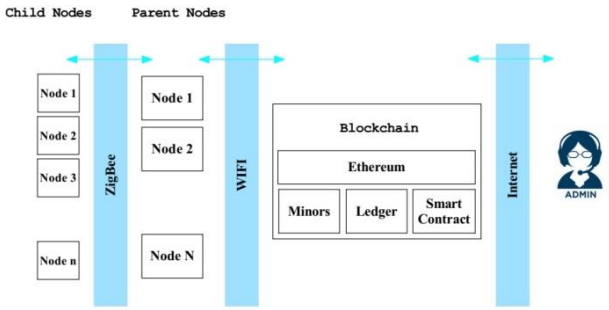

Fig. 7. Communication topology between nodes and different components in the proposed framework.

Algorithm 1: The LOG_CMD Contract, value, timestamp, and nodeID are associated with the capturing event.

```
1 Method Update(_NodeId,_Timestamp,_value):
2   if  Node.Ready() then
    // clone the values as a trx. data
3       value = _value
4       timestamp = _Timestamp
5       nodeID = _NodeId

6 Method getValue():
7    return value
```

Algorithm 2: The STOP_LOG_CMD Contract, parameter Mode is set = 1 to enable capturing

```
1 Method Update (Mode):
// clone the value
2   status = _Mode
3
4 Method getStatus():
5 return status
```

Algorithm 3: Algorithm for the Threshould Contract, parameter Limit is set to specific value by an admin.

```
1 Method Set_Threshould(_Limit):
2   Limit = _Limit
3
4 Method getLimit():
5   return _Limit
```

Algorithm 4: Algorithm for the ACT_CMD Contract, parameter LedVal is set to specific value by an admin.

```
1 Method Actuate(LedVal):
2   LedVal =_LedVal
```

Algorithm 5: Algorithm for the AUTO_ACT_CMD Contract, parameter LedVal is set to specific value by an admin.

```
1  Method Auto_Actuate(value):
2    if value ≥ Threshould.getLimit() then
3      Actuate(_LedVal)
```

F. The Proposed Framework Delay Analysis

In this work a delay analysis is introduced. The first step is to find an expression for the frameworks' data rate, Rm. The data rate is identified for every communication channel where the captured IoT updates propagate through. Let R_z, R_w, and R_i refer to zigbee, Wifi, and Internet data rates consequently. During this work, IoT nodes continuously capture and send environmental values. The values captured from parent nodes are transmitted directly to the minor and Blockchain network through Wifi and Internet consequently whereas values captured from child nodes are propagated to the parent node via zigbee channel and relayed to minors then the BC. Zigbee data rate R_z in the proposed framework topology is considered the minimum data; hence it acts as a bottleneck for the entire framework. Generally the framework operating data rate is expressed as in equation 10:

$$R_m = \min(R_Z, R_w, R_i) \quad (10)$$

Table I shows the data rates for the hardware modules included in our IoT PoC.

TABLE I: THE DATA RATES FOR THE MODULES INCLUDED IN OUR IOT POC FOR THE PROPOSED FRAMEWORK

Module	Protocol	Wireless Transmission Range
Xbee	IEEE 802.15.4	35 Kbit/sec
ESP8266	IEEE 802.11	11 Mbit/sec

Delay is the time which a packet encounters from a source to destination. The average delay is the value of the elapsed time from transmission of the first chunk, i, of the message from the source to reception of the last chunk of the message at the destination. The transmission delay is encountered by every packet in the proposed framework. However the processing delay, which is the average block generation rate in blockchain, is encountered for every environmental value captured by IoT node. The other sources of delay such as the propagation and the queuing delays are neglected because of their small values. Every packet is transmitted by data rate Rm through either two nodes (Parent→Minor→BC) or three (Child→Parent→Minor→BC). The Average delay is the sum of every packets' delay passing through 2.5 node on average in addition to BC delay time to the entire event. Equation 11 shows an expression for the average events' delay in seconds:

$$D \leq \frac{5}{2N_p}\sum_{i=1}^{N_p}\left(\frac{N_p L}{R_m}\right) + D_{BC} + \epsilon \qquad (11)$$

where N_p is the number of packets or events propagates from the nodes to the minor and ϵ is the time to propagate the last packet in the last event. Blockchain delay D_{BC} can be expressed as in equation 12:

$$D_{BC} = T_{BC} + T_{proc} \qquad (12)$$

where T_{BC} The time elapsed to build a block which equal to the time required to accumulate group of transactions each is related to an event, this group length is N_{Trx}, and T_{proc} is the Blockchain's average processing time. The next step is to characterize T_{BC}. The values captured and sent from X parent node and Y child are following Poisson distribution and M/D/with λ event propagates per second. In this case T_{BC} can be expressed as in equation 13:

$$T_{BC} = \frac{N_{trx}}{\lambda(X+Y)} \qquad (13)$$

Equation 15 shows the theoretical delay encountered in the proposed framework.

$$D \leq \frac{5}{2N_p}\sum_{i=1}^{N_p}\left(\frac{N_p L}{R_m}\right) + \frac{N_{trx}}{\lambda(X+Y)} + \epsilon \qquad (14)$$

IV. Results

The goal of the proposed framework is to secure all of IoT transactions by utilizing BC with taking into consideration a set of requirements. In this paper three common types of attacks is analyzed and their ability to compromise proposed system is tested. Message interception or modification occurs if messages' integrity are violated. In the IoT malicious user may change the value captured by an IoT sensor and reported via the *Update_Value* contract. To prove that an attacker probability to propagate incorrect values is decreasing with prolonging the chain of blocks; we implement equation 9 in C# programing language to obtain range of values for this probability with different values of p_a and z, see Fig. 8.

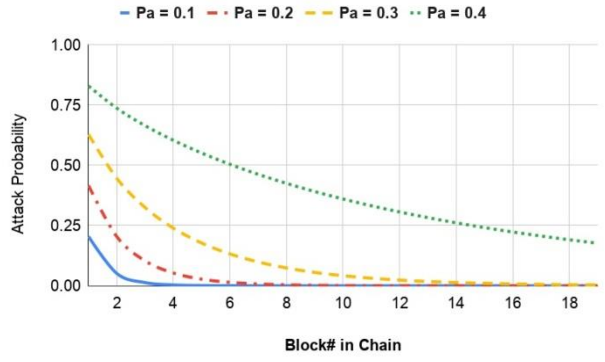

Fig. 8. System attack probability plotted against different block count and malicious block generation probability of p_a

Fig. 8 shows the system probability to be compromised with different block count in chain and different values of p_a. The probability of system compromising is reduced exponentially by increasing the block length. In the proposed framework we define the term Stable and Secure (*SS*) as the state where $p_{az} < SS_{val}$ for any number of blocks z and pa. By solving equation 8 programmatically we got the following result set in Table II.

Table II also shows that SS system state have the following characteristics:

a) With increasing p_{az} and z: SS ranges are stretched and extra blocks are included.
b) SS ranges shrinks by increasing pa, if the value reaches 0:51 experiments shows that no z ranges are considered save for any value of p_{az}.

TABLE II: SECURE AND STABLE (SS) CASES FOR DIFFERENT VALUES OF BLOCKCHAIN LENGTH, Z , AND PROBABILITY OF APPENDING INVALID BLOCK TO THE CHAIN P_A.

SS_{val}	p_a	Insecure Chain length z	SS z Ranges
0.001	0.1	$z <= 4$	$4 < z < 1$
	0.2	$z <= 10$	$10 < z < 1$
	0.3	$z <= 23$	$23 < z < 1$
	0.4	$z <= 88$	$23 < z < 1$
	0.51	Compromised	N/A
0.01	0.1	$z <= 3$	$3 < z < 1$
	0.2	$z <= 6$	$6 < z < 1$
	0.3	$z <= 15$	$15 < z < 1$
	0.4	$z <= 57$	$57 < z < 1$
	0.51	Compromised	N/A
0.1	0.1	$z <= 1$	$1 < z < 1$
	0.2	$z <= 3$	$3 < z < 1$
	0.3	$z <= 6$	$6 < z < 1$
	0.4	$z <= 26$	$26 < z < 1$
	0.51	Compromised	N/A

TABLE III: LIST OF COMMON ATTACKS AND THEIR ABILITY TO COMPROMISE THE PROPOSED FRAMEWORK.

Attack	Ability to Compromise System	Justification
Interception and Modification	No	BlockChain Consensus mechanism exclude transactions will malformed
Replay	No	BC transactions are timestamped and cannot be replayed
DoS/DDoS	No	Ethereum BlockChain Can't be down causative the distributed nature and powerful enough to prevent this type of attacks

Finally The DoS attach is the process of setting overall system down. For DDoS The same attacker target is achieved from different points. This type of attack requires

very expensive computing power to control the 51% BC peer Nodes, so this attack is infeasible. Table III lists common types of attacks and their ability to compromise the system.

TABLE IV: ADDRESSING THE MAJOR IOT PROBLEMS IN THE PROPOSED FRAMEWORK.

Issue	How the issue is fixed
Scalability	BlockChain is scalable framework by nature. The proposed secured framework inherits its core security from BC. So, there is no problem with scaling IoT as the BC layer will handle all the security issues with no chance for failures.
Heterogeneity and Resource Limitedness	The core security of the proposed frameworks is performed in the BC Logic itself and does not depend on the IoT devices. As a result, their heterogeneity does not matter.
Transparency	The proposed framework allows devices to join without a lot of work. Every IoT device should have an ID to flag its updates and attached to the network minor. Finally, an administrator must define the access roles for the device by means of utilizing smart contracts.

In Table IV, a list of major IoT problems facing security and how the proposed framework can handle them.

V. CONCLUSION

In this work, a secured IoT framework based on blockchain is proposed which satisfy a set of requirements such as robustness, transparency, security, and lightweight. A mathematical analysis for both security and delay between different components are introduced. Finally the numerical results ensure the frameworks' security and robustness against different attacks.

ACKNOWLEDGMENT

This paper is an extended work for conference paper presented in ICICIS 2019 entitled "Secured Framework for IoT Using Blockchain" [2].

CONFLICT OF INTEREST

The authors declare that there is no conflict of interest regarding the publication of this article.

AUTHOR CONTRIBUTIONS

In Performance Evaluation of a Secured Framework for IoT based on BlockChain

Ali H. Ahmed

1. Review existing secured frameworks for IoT
2. Suggesting the PoC to the proposed framework.
3. Programing the Ethereum framework and the solidity-based E-contracts.
4. Deriving the mathematical model.
5. Programming and implementing the mathematical model
6. Write the paper.

Nagwa M. Omar and Hosny M. Ibrahim

1. Revise the paper and ensure the mathematics
2. Write the paper.
3. Guidance and validation of the whole work

REFERENCES

[1] M. Zorzi, A. Gluhak, S. Lange, and A. Bassi, "From todays's INTRAnet of things to a future INTERnet of things: A wireless- and mobility related view," *IEEE Wireless Commun.*, vol. 17, no. 6, pp. 44–51, 2010.

[2] H. Ahmed, N. M. Omar, and H. M. Ibrahim, "Secured Framework for IoT Using Blockchain," in *Proc. Ninth International Conference on Intelligent Computing and Information Systems (ICICIS)*, Cairo, Egypt, 2019, pp. 270-277.

[3] A. H. Ahmed, N. M. Omar, and H. M. Ibrahim, "Modern IoT architectures review: A security perspective," in *Proc. International Conference on ICT: Big Data, Cloud and Security (ICTBDCS)*, 2017.

[4] P. Datta and B. Sharma, "A survey on IoT architectures, protocols, security and smart city based applications," in *Proc. International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, 2017.

[5] Satoshi Nakamoto, *Bitcoin: A Peer-to-Peer Electronic Cash System*, academia.edu, 2009.

[6] J. Leng, G. Ruan, P. Jiang, K. Xu, Q. Liu, X. Zhou, and C. Liu, "Blockchain-empowered sustainable manufacturing and product lifecycle management in industry 4.0: A survey," *Renewable and Sustainable Energy Reviews*, vol. 132, 2020.

[7] IDC, *Connecting the IoT*. http://www.idc.com/infographics/IoT/ATTACHMENTS/IoT.pdf

[8] I. BouiJ-Pasquier, A. A. El Kalam, A. A. Ouahman, and M. D. Montfort, "A security framework for internet of things," *Cryptology and Network Security*, 2015.

[9] S. Sridhar and S. Smys, "Intelligent security framework for IoT devices," in *Proc. International Conference on Inventive Systems and Control (ICISC)*, 2017.

[10] M. Ammar, G. Russello, and B. Crispo, "Internet of things: A survey on the security of IoT frameworks," *Journal of Information Security and Applications*, vol. 38, pp. 8–27, 2017.

[11] X. Liu, M. Zhao, S. Li, F. Zhang, and W. Trappe, "A security framework for the internet of things in the future internet architecture," *Journal of Future Internet*, vol. 9, no. 3, pp. 1–28, 2017.

[12] H. Gupta and G. Varshny, "A security framework for IoT devices against wireless threats," in *Proc. International Conference on Telecommunication and Networks (TEL-NET)*, 2017.

[13] S. Huh, S. Cho, and S. Kim, "Managing IoT devices using blockchain platform," in *Proc. International Conference on Advanced Communication Technology (ICACT)*, 2017.

[14] Ethereum White Paper. [Online]. Available: https://github.com/ethereum/wiki/wiki/WhitePaper
[15] Ethereum. *Writing a Contract*. [Online]. Available: https://github.com/ethereum/goethereum/wiki
[16] Wood, Ethereum: *A Secure Decentralised Generalised Transaction Ledger*. [Online]. Available: http://gavwood.com/paper.pdf
[17] Ethereum. [Online]. Available: https://www.ethereum.org/
[18] [Online]. Available: https://drive.google.com/open?id=1rzJXVtE8NbSHhSHI899mN2cs1iA-CJXf
[19] Remix. [Online]. Available: https://remix.ethereum.org/
[20] SOLIDITY. [Online]. Available: https://solidity.readthedocs.io/en/v0.5.3/types.html
[21] STORE. [Online]. Available: https://store.arduino.cc/usa/arduino-mega-2560-rev3
[22] MICROCHIP. [Online]. Available: https://www.microchip.com/wwwproducts/en/ATmega2560

Ali H. Ahmed received the B.Sc in Information Technology in 2008 from Faculty of Computers and Information, Assiut University, Egypt. He completed master degree in 2013 in wireless sensor networks. Besides the research work in his PhD, he is a teaching assistant in information technology department, and he had more that 12 years in the academic work. His research interests include Security, IoT, WSN, image processing, and computer vision.

Nagwa M. Omar received the B.Sc., M.Sc., and PhD degrees in Computer Engineering from the Faculty of Engineering, Assiut University, Assiut, Egypt, in 1999, 2002, and 2008 respectively. She worked as Assistant Professor at the Information Technology Department, Faculty of Computers and Information, Assiut University, Assiut, Egypt, from 2009 to 2016. She is working as Associate Professor at the Information Technology Department, Faculty of Computers and Information, Assiut University, Assiut, Egypt from 2016 to now.

Hosny M. Ibrahim received the B.Sc., and M.Sc. degrees in Electrical Engineering from the Faculty of Engineering, Assiut University, Assiut, Egypt, in 1973, and 1977 respectively. He received the Ph.D. degree in Electrical Engineering from Iowa State University, Ames, Iowa, U.S.A. in 1982. He was the Dean of the Faculty of Computers and Information, Assiut University, Assiut, Egypt from September 2002 to August 2011. He was the head of the Information Technology Department, Faculty of Computers and Information, Assiut University, Assiut, Egypt from July 2010 to May 2015. He is currently Professor at the Information Technology Department, Faculty of Computers and Information, Assiut University, Assiut, Egypt.