Trusted Execution Technology (TEE) in a Secure Execution Environment for IoT

Jawad Ali[1], Ahmad Sharadz Khalid[1], Eiad Ya[1], Shahrulniza Musa[1], Waqas Ahmed[2]
[1]Malaysian Institute of Information Technology
Universiti Kuala Lumpur, Malaysia
{jawad.ali@s.unikl.edu.my} {ahmads@unikl.edu.my} {eiad@unikl.edu.my} {shahrulniza@unikl.edu.my}

[2]UniKL Business School, Universiti Kuala Lumpur, Malaysia
{waqas.ahmed@s.unikl.edu.my}

ABSTRACT: *In the last couple of years, the Internet of Things (IoT) is found to have increasing applications. The IoT components include smart-devices which communicate and exchange the information without the physical intervention of humans. The growth of newer models of IoT and their systems lead the devices more vulnerable and prone to a severe kind of threats. This current study has introduced a new system capturing and verification procedures in Blockchain supported smartIoT systems that can show the trust-level confidence to outside networks. This work has a Behavior Monitor and get implemented on a selected node that can extract the activity of each device and analyzes the behavior using deep machine learning strategy. In addition, we use Trusted Execution Technology (TEE) which can provide a secure execution environment (enclave) for sensitive application code and data on blockchain. To prove the proposed model, we analyze various IoT devices data that is infected by attacks. Experimental findings prove the ability of our proposed method in terms of accuracy and time required for detection.*

Keywords: Security, Privacy, IOT, Blockchain, Trust, Behavior, Neural Network

Received: 24 January 2020, Revised 2 June 2020, Accepted 16 June 2020

DOI: 10.6025/jisr/2020/11/3/75-85

1. Introduction

Currently, in the modern world Internet of Things (IoT) is rapidly increasing and involved in every part of our daily life. According to the industry-leading experts' argument that more than 50 billion devices will be deployed by 2020 [1]. Things in IoT are composed of web-enabled devices that use embedded processors, sensors, micro-controllers and communication hardware (send & receive data from different environments). Such rich communication in IoT devices produces a large volume of data which in turn to use for various dependent services.

Apart from this, IoT allows the advancement in several areas such as home to smart-home, cities to smart-cities, school to smart-school, health-care to smart- health-care, and many more. The main idea behind the IoT ecosystem is the diversity of things that outputs in a massive number of devices. Each device (physical or virtual) connected to the system, should be traceable and the generated information from the device can be retrievable by other users irrespective of their locations [17]. Nevertheless, it is necessary that only authorized users can be able to enter and make use of the system and its resources. Otherwise, it may face several security concerns such as data modification, identity theft and information leakage. Moreover, security and privacy problems remain a demanding challenge in such a massive scale adoption of IoT systems because of the following reasons: (1) Mostly the communications between these IoT devices are wireless which make the system more susceptible to different attacks, i.e. message tampering, eavesdropping and denial-of-service attacks like mirai attack [2] etc. (2) Devices from different company-makers have resource constraints limitation such as processing power, battery and memory capacity that do not allow to deploy advanced security solutions.

Numerous solutions concerning security and privacy in IoT have been proposed that provide the mainstream security requirements i.e. Confidentiality, Integrity, Authentication or simply CIA [23]. However, due to heterogeneous nature and resource-constrained devices, existing solutions cannot fulfill the desired security requirements in the upcoming large-scale IoT system. Even though some security based solutions are efficient and secure but are commonly based on centralized mechanisms. For instance, PKI (Public Key Infrastructure) faces with scalability issues in case of million nodes.

When it comes to decentralization, Block-chain (BC) technology has acquired an enormous attention in regard to tackling security, anonymity, traceability, and centralization. Ethereum [33] a public blockchain was introduced in 2014 that run smart-contracts for BC users in order to write and execute application code in a distributed way. Basically, Blockchain is a distributed ledger technology where each operation such as create, read, update and delete, is recorded in the form of a transaction. Any unauthorized user accessing data or any operations on the previously processed data can, therefore, be detected. Furthermore, smart contracts are used to apply some access control mechanisms on the stored data. A number of researches have shown the integration of BC technology in different IoT use-cases. [15, 32, 10, 8, 31, 19, 12, 7, 14].

Problem Statement and Contribution:
As from various studies, it has been found that blockchain has become a promising technology to meet future IoT security requirements [13]. Several Authors [16, 15, 30, 18, 17] put efforts in decentralized security mechanisms for upcoming large-scale IoT systems. But the limitation to all the approaches is that: there is no device-level trust that can prove any particular zone to external entities in case of supposing the communication to occur between dierent IoT networks.

The contribution of this paper is as follows:

1. Implement our own custom Behavior Monitor in IoT-Blockchain setup that can store & monitor IoT devices data and classify its behavior (normal or malicious) to prevent attacks.

2. Applying a filter on sensor-level that can stabilize output from single/multiple sensors to avoid faulty or malicious sensors in the network.

3. To implement Trusted Execution Environment (TEE)) on a local blockchain of each IoT-Zone that ensure the integrity and confidentiality of sensitive application code and data.

2. Background

2.1 Internet of Things (IoT)

The Internet of Things is the interconnection of smart-devices, mechanical and digital machines, objects and people that are capable of transferring data over the network without any human intervention. On the broader scale, IoT applications areas are smart-homes, smart-cities, smart-healthcare, etc. The major components [6] in IoT ecosystem includes:

- **Smart-devices & Sensors:** The first layer is the device connectivity layer of IoT network, which constitutes different sensors like temperature sensor and thermostat, humidity sensor and many more.

- **Connectivity:** Devices in IoT are connected to low power wireless networks like LoRAWAN, ZigBee and Wietc.

- **Gateway:** It acts as a middle layer between devices and manages the bi-directional transmission between networks and protocol. One of the key functions of a gateway is to translate different protocols and make them interoperable.

- **Cloud:** This component integrates billion of sensors, smart-devices gateways, data storage and provides different predictive analytics.

- **Analytics:** This is the process of converting the raw data (analog) of billion of devices into useful insights which can be further used for detailed analysis.

2.2 Blockchain - A Decentralized Technology

Blockchain technology was initially introduced and brought in 2008 and used by a remarkable known cryptocurrency, Bitcoin [26]. It is a decentralized ledger technology that builds on a peer-to-peer network. Each node in the BC network holds an updated ledger copy that can hinder from a single point of failure. In the past few years, the blockchain mostly based on crypto currencies such as [26] [9] in order to avoid the double-spending problem. However, recently numerous application areas have been explored where the blockchain can be set up to create and maintain digital transaction records in a secure and distributed fashion.

The ledger in BC is composed of blocks, and each block contains two parts. The first part represents the transaction (that need to be stored in a database), which can be of any kind, such as patient record, network traffic log, goods transaction, etc. The second part includes the header information such as hash of a current transaction, hash of previous hash and timestamp. Thus, storage in this way makes a sequenced block of linked chain as shown in Figure 1. Furthermore, if a new transaction comes, it will first add to certain block. Secondly, miners verify the block contain the transaction according to already defined rules. After the verification process, all miners perform a consensus strategy to validate the transactions. Finally, upon successful validation the verified transaction is ready to append in the BC ledger.

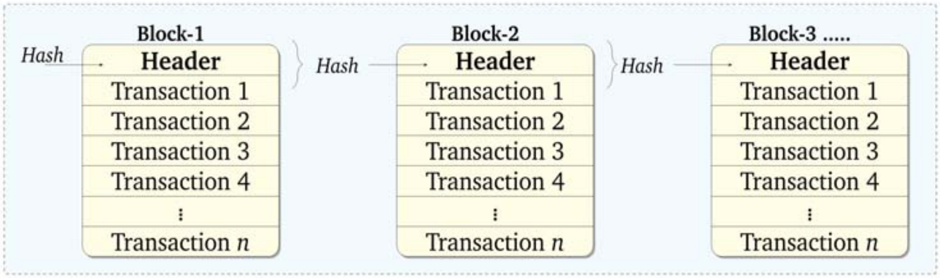

Figure 1. Inter-Linked Blocks in Blockchain

2.3 Blockchain and IoT Systems

IoT devices generate a massive volume of data, that must be appropriately stored and analyzed for useful purposes. For each IoT operation (create, update, delete, read), the data can be registered in the form of transactions in the BC-blocks. Device identity information can be registered in a block such as manufacturer information and live status where the device is in use. Smart-contracts are used to enforce access control policies for IoT devices which means that any unauthorized access to a device can be therefore identified. There is no need for a central authority for storage, such as cloud, for IoT protection. Blockchain provides data authenticity, data integrity, traceability and prevents from unauthorized access. BC can also enable a secure channel of messaging between IoT devices. Exchange of messages from one device to another device can be handled like financial transactions flow in crypto-currencies, e.g. Bitcoin [26].

2.4 Blockchain Security Solutions for IoT

The decentralized and distributed nature of blockchain makes it a promising security solution for IoT use-cases. IoT and blockchain integration enables a higher and sound security level, which otherwise could not be accomplished by any other technology or nearly impossible. Some of recent proposals in regards to IoT security with blockchain are as follows:

In [21], authors proposed a blockchain-based solution for managing IoT devices and configurations using Ethereum. A unique key-pair (Public & Private) is assigned to each device in the network. The private key is kept inside the device, while the public key is registered as a transaction in the blockchain. An IoT device can then be reached and access through ethernet by its public key. Thus, it is concluded that the management and control of IoT devices through blockchain is possible.

A solution proposed in [24], which make use of blockchain for secure firmware updates in IoT devices where traffic directly to the network server is replaced by local peers of the blockchain nodes. The manufacturer is supposed to store the hashes of updated firmware on the blockchain that can be easily accessible to all the IoT nodes.

IoT devices using in medical and healthcare zone are also subjected to the same security and privacy issues. For medical IoT system, it must be attack resistant and reliable enough. User safety and privacy is very critical and must be protected from any malfunction caused by a security incident or imprecise/faulty device. The risk of device malfunction can overcome in blockchain by immutable ledger technology. Nichol et al. [28] proposed the feasibility of BC in order to provide reliability in medical IoT devices. Upon a device is manufactured and installed, a hash of UID (unique identifier) along with the other relevant information such as manufacturer information are stored in BC. Later, this data will be updated with doctor-name, patient-history, and hospital information. The doctors and patients can be automatically informed about the device status like battery expiry, patient health irregularities.

2.5 Blockchain & Trusted Execution Environment (TEEs)

Trusted Execution Environments (TEEs) [4, 3] have been used to enhance security and efficiency in the blockchain protocol. TEEs provide confidentiality and integrity to the sensitive part of application code in a system, until and unless the CPU is not compromised physically by an attacker. TEEs also support remote attestation [22], that allows remote parties to verify the health of software with genuine TEE.

Intel provided TEEs functionality in Software Guard Extension (SGX) [4]. SGX is a set of CPU instructions inside Intel’s x86 processor design which can allow creating an isolated environment for the execution of selected pieces of code in protected areas called enclaves. These enclaves are designed to run software in a trustworthy environment, even on a system (host) where the operating system and memory are untrusted. There are three main functions of enclaves which are isolation, sealing and attestation. A short description is as follows:

- **Isolation:** Data and code inside the enclave memory are protected and cannot be read or altered by any external process.
- **Sealing:** Data supposed to send to host environment should be encrypted and authenticated with a seal key.
- **Attestation:** Remote parties are allowed to verify an application enclave identity, credentials, and other data.

3. Related Work

Currently, several types of research have been proposed in the integration of blockchain and IoT. Very few of them have shown interest to help IoT security requirements. This section outlines some of the past researcher efforts that intend to realize such integration, mainly for security needs.

Raja et al.[15] demonstrate blockchain-based architecture for smart-home setting. The architecture consists of three different blockchain networks: a local-BC (private), a share BC (private) and overlay BC (public). Although this research solves the issue of identification, still it has some shortcomings such as (1) For each operation, it happened to make at least eight communication links that can flood the network quickly in case of high activity of IoT devices. (2) Local BC’s are centralized and not distributed which is opposite to the main principle of BC - a decentralized technology.

In [29], authors study existing proposed models of access control systems and argue that these systems are not effective in the upcoming large-scale IoT. In order to avoid centralized mechanisms, this proposed research implements capability and access control as a component in a blockchain environment. The other components are data management protocol, messaging service and data storage system. The messaging service deals with the exchange of access control message between two parties with defined roles. The messaging service, then sends a request to the data storage system, where it is stored in the form block. Finally, the receiving party fetches the message from the BC block using the messaging service. Moreover, they defined four roles, i.e. data owner, data source, requester and endorser.

A mechanism named as chainanchor proposed in [18] based on the authorization of IoT devices in the cloud network. It helps device-owner being rewarded upon selling their device data to a service provider and ensure a privacy-preserving communication between owner and service-provider. But this approach is not suitable in most IoT use-cases, because the main scope of this research is full anonymity and IoT devices sometimes need device identification.

Patrick et al. [16] propose a decentralized authentication scheme for IoT devices. In this approach they declare virtual zones like healthcare zone, smart-school zone, smart-school zone for robust identification of smart-devices. Each zone has a group master who is responsible to create a groupID and communicate with blockchain. Each device or follower in a zone gets a ticket signed by their respective zone master. When a device or follower wants to initiate a transaction, an association request signed by private-key is sent to their respective zone master. Upon receiving the request, BC verifies its integrity with the public key of follower. Afterwards, the follower ticket is verified using the master public key. If the ticket found valid, BC stores the association of follower ID with their groupID for further correspondence, otherwise discarded. However, the limitation of this approach is that no mechanism can provide trust-level confidence in each zone to prove it to the outside community.

To summarize, the majority of all these current proposals follow the same security schemes provided by existing BC technologies, i.e. Bitcoin [26], Ethereum [33] etc. However, there is no awareness towards device level trust that means to know the status of running IoT device, whether it is normal or malicious.

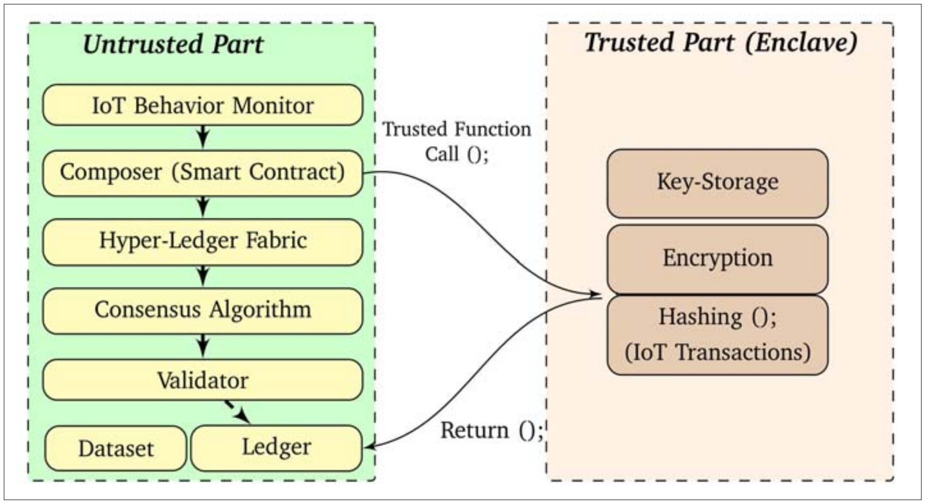

Figure 2. IoT secure behavior capturing and storage environment using TEE

4. Proposed Framework

The main goal of the proposed framework (cf. Figure 4) is to add and implement a security module for behavior monitoring of various IoT-zones in a blockchain setup. As discussed in [16], authors declare zones for different use-cases of IoT. However, they do not consider the devices itself in case of compromised behavior. Furthermore, there is no mechanism that can show the trust-Level confidence of each zone when an external entity needs to know before establishing a connection. In this research, we enhance the said scheme and add a behavior monitor on each zone. A separate local-BC is configured on each zone that is used to store the activity of each zone and provides the trust-level confidence to outside entities.

All kinds of communications between devices are considered as transactions and must be passed through the blockchain for validation. For example, if node A needs to send a message to node B, then A must first send the message to blockchain. If BC validates and authenticates the message from A, then B is finally allowed to read the message.

4.1 Initialization & System Functioning

In the first phase of deployment, one device from each zone is designated as a Main or Master node, which can be considered as a certification authority (CA). Any node can be declared as a master, but in this case, we assigned to the node that is more resource capable and powerful. All the other nodes in each zone are known as follower. Every Master node creates a groupID and send a signed ticket to each follower for identification. For the first transaction of any follower, it must require authentication. After that, an association of the follower and master are stored in the *BC* for future correspondence.

Hardware Model of IoT The hardware architecture we use in our proposed framework for prototyping consists of multiple

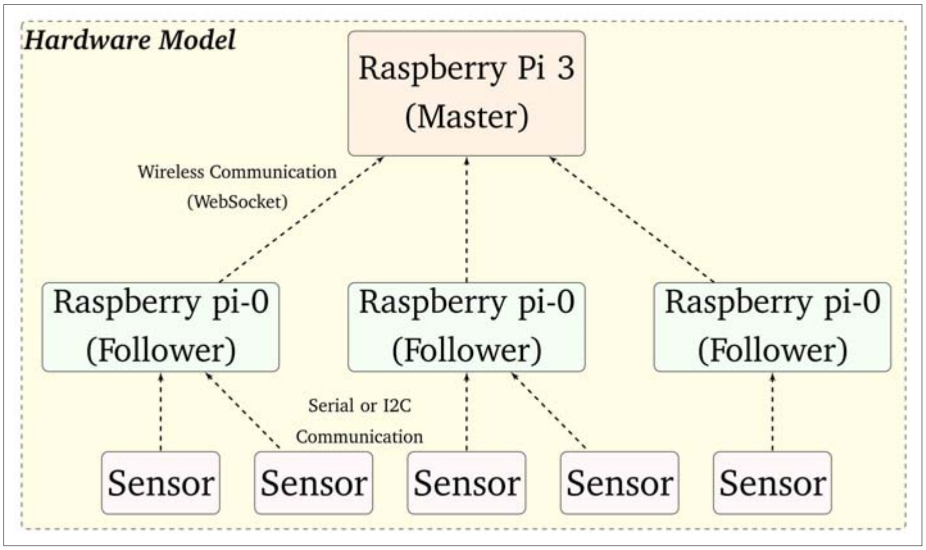
Hardware Model
Raspberry Pi 3 (Master)
Wireless Communication (WebSocket)
Raspberry pi-0 (Follower)
Raspberry pi-0 (Follower)
Raspberry pi-0 (Follower)
Serial or I2C Communication
Sensor
Sensor
Sensor
Sensor
Sensor

Figure 3. Hardware Model for IoT Zone

raspberry *pi's*. The main/master node is configured on raspberry *pi*-3 for the sake of more resources. Followers or clients node work on raspberry *pi*-0 with a direct connection to sensors and other digital devices. Wifi is used for communication between master nodes, and follower communicates to their sensors using serial or I2*C* communication protocol as shown in Figure 3.

Every device is assigned by a key pair that consists of a public and private key. The private key is stored in follower (*pi*-0), while the corresponding public key is stored in their respective master node (*pi*-3). The connection between the follower and master node is established through WebSocket. Upon a connection request from follower to master, the follower must be required to send a digital signature. Afterwards, master node should validate the digital signature in the blockchain before a secure WebSocket authorization.

Improving Sensor Level Data Accuracy In order to improve sensor level security, the data acquisition procedure will use Kalman filter to make a data model based on single/multiple sensor readings and covariance. For example, the position of a

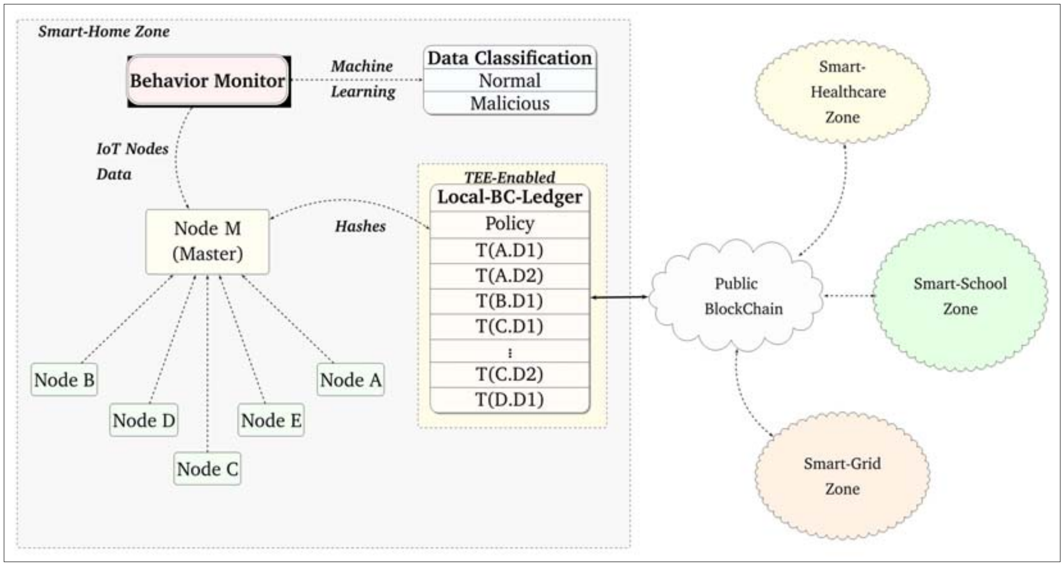

Figure 4. Proposed IoT Blockchain Framework

drone can be estimated in 3-axis based on GPS, but GPS alone cannot guarantee accurate altitude. Similarly, a Barometer data can drift based on different weather conditions at the same altitude. Radar or Lidar will output the altitude value from the ground, but if an obstacle supposed to happen between the ground and radar the readings might become inaccurate. To avoid such discrepancies, Kalman filter uses data from all the 3 sensors GPS, barometer and radar/lidar, to predict the correct value (3D location) based on the covariance. This way, if a faulty or malicious sensor found, the Kalman filter will automatically filter out the data from that sensor.

4.2 Configuring Local Blockchain

A local private blockchain is deployed on a master node (Raspberry pi-3) of each zone and populated with the hashes of transactions generated from smart- devices. Hyperledger Fabric [9] a permissioned-BC is implemented as a local BC, we discussed the workflow of fabric with IoT in our previous research [8]. For prototype implementation, we use the dataset [5] of IoT traffic that has been collected from various sensor communication. For each communication between nodes or smart-devices, a transaction is created and stored in the local BC. Note that in the majority of the current BC technologies, actual data of IoT devices are not stored in the BC due to overheads (i.e. processing & network).

In each zone, a single device having more computational power than others, acts as a master or main node. Likewise in our model, we use raspberry *pi* - 3 which is computationally and energy-efficient act as a master/main node. Once the number of transactions reaches a pre-define blocksize, the master node creates a new block and append it to local BC. Afterwards, we realize Intel SGX [4] as a root-of-trust on top of BC to ensure that the execution of sensitive code and applications are in trusted mode. As shown in Figure 2, the TEE-enabled application is composed of a trusted and untrusted part. For sensitive operations like encryption and hashing, a trusted-function is called. The function returns, and the data inside the trusted part (enclave) remains in trusted memory and is not accessible to external entities. Moreover, implementing SGX technology on blockchain allows the proposed scheme to:

- Protect the applications running on BC and data protection that cannot be accessed by the execution host.
- Make sure that the application/data on BC is expected and correct.
- Protect end-to-end privacy of application result, which cannot allow others to inspect but the user.
- Provide a BC-based validation by verifying the applications inside enclave is neither tampered nor interrupted by any node in BC.
- Make sure the application and execution results are valid, and not tampered or fabricated by any malicious node.

4.3 Behavior Monitor

The main goal of this research is to integrate our custom behavior monitor that can classify the behavior of every device and compute a level-of-trust on each zone. As mentioned earlier, all the nodes (followers) in a specific zone do their operations (read, write) via the master/main node. The scheme in Fig. 4 depicts our proposed approach with all the entities in detail. Data or transactions from nodes is considered as a behavior of that particular node. The master node is a device that centrally processes all the incoming and outgoing transactions to and from a zone.

Whenever a data is received by the master node from the follower node, the master node stores the data in the behavior monitor and appends the corresponding hash to the ledger in blockchain. A sequence-ID (SEQ-ID) is assigned to each transaction while storing in behavior monitor, and a Hash-ID (H-ID) is attached to the corresponding hash in BC, for reference. Finally, a machine learning strategy is used to actively monitor the incoming data and classify them as normal or malicious.

For analysis and detection of behavior, we rely on deep Auto-encoders (AE) [20, 27] for IoT devices, which is trained from statistical correlation features extracted from benign data. The process of behavior detection and monitoring consists of the following stages. (1) Data collection (2) Feature extraction (3) Training model (4) Continuous Behavior Monitoring.

Data Collection At this point, we refer to the dataset [5] that has been collected from various sensors in the IoT network. In real-time, to ensure that the training data is clean and not malicious, normal traffic from IoT devices are collected immediately after its joining to the IoT network.

Feature Extraction Whenever data from IoT devices arrives, a behavioral snapshot of the protocols and host related to data are stored in our behavior monitor. The snapshot contains different parameters, i.e. source IP, destination IP, MAC-address and port number, etc. We use the same set of features mentioned in the dataset for real-time detection of malicious activities in IoT devices. For example, when a compromised node in a zone spoof an IP, then the features aggregated from the source-IP, destination-IP and MAC-Address will immediately mark as malicious because of unseen activity from the respective spoofs IP.

Training Model As our baseline model for behavior detection, we use deep auto-encoders that can build and maintain a learning model on each zone of IoT use-cases. An auto-encoder is a type of artificial neural network (*ANN*), which is trained to re-structure the data after some compression. The compression ensures that the model would be able to learn meaningful concepts and the correlation between different sets of features. For training purposes, we use two sets of data which consists of only benign (normal) data. The first dataset is a *training dataset* (T_{DS}) which is used to train the auto-encoder by declaring input parameters such as learning rate (lr_n, size of gradient descent step), and epochs (number of iterations through T_{DS}). The second dataset Opt_{DS} (*Optimization Dataset*) is used to optimize the above hyper-parameters (lr_n *& epochs*) iteratively until the mean square error (*MSE*) function between the input and output stop decreasing. This stopping prevents overfitting in TDS and help out better detection results with future data. Later on, (Opt_{DS}) is used to identify normal and malicious activities and false positive rate (*FPR*).

After the model training and optimization is completed, the threshold value (th^v) is set by which an instance of data is considered malicious. Empirically, it is calculated by the sum of the sample mean along with the standard deviation of *MSE* on Opt_{DS} (see Equation 1).

$$th^v = \overline{MSE}_O\ pt_{DS} + s\ (MSE_O\ pt_{DS}) \qquad (1)$$

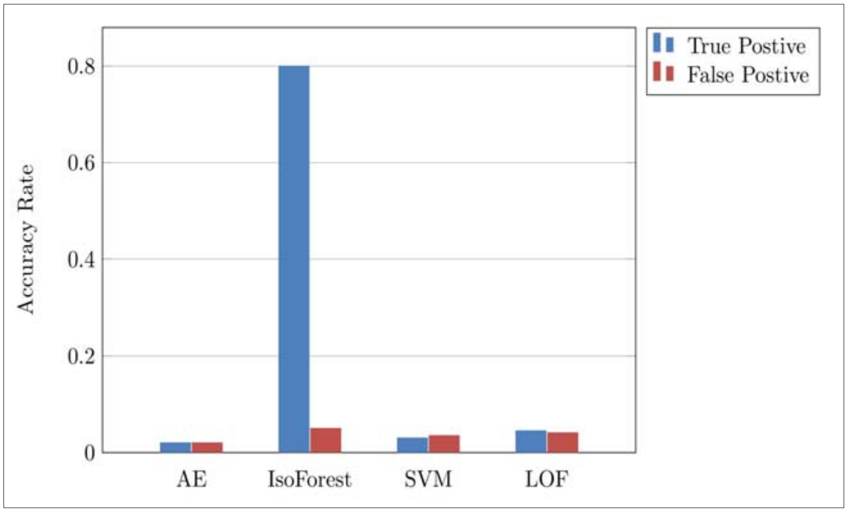

Figure 5. Detection Accuracy comparison with other Algorithms

Continuous Behavior Monitoring Finally, the model is applied to continuously observe the data and to label each instance as normal or malicious. Consequently, an alert against abnormal behavior can be issued to indicate the IoT device is malicious. Afterwards, for each IoT zone, the behavior monitor calculates a trust-level measurement and a threshold must be defined for every use-case. Whenever a user or node from outside needs to accessed data from any specific zone, our model is capable of disclosing the health of zone before establishing a connection. This way a trusted environment can be built and informed the user about the state of any particular zone before actual communication.

5. Experimental Analysis

In our experiments, we use a real-time large dataset available in [5], for realizing the framework. The dataset contains both

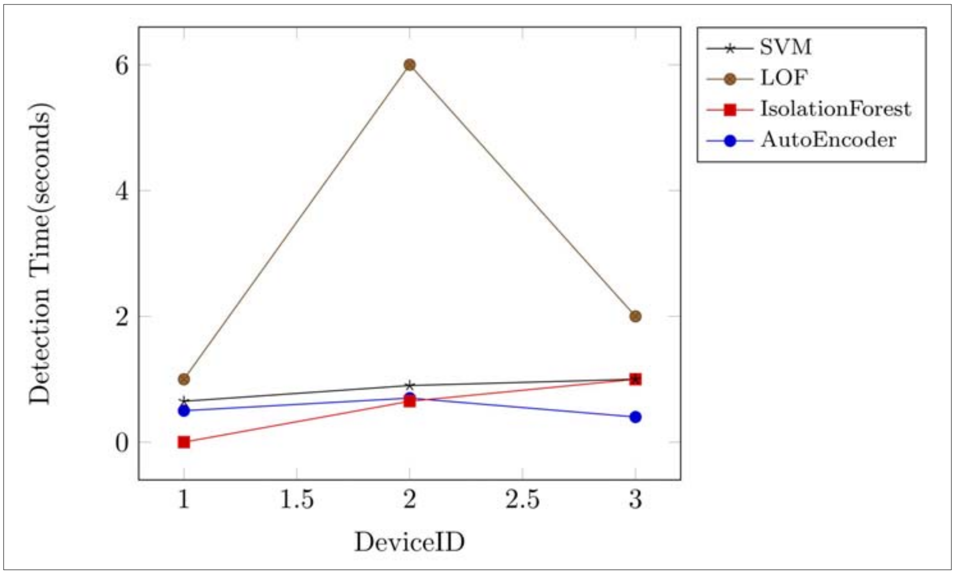

Figure 6. Detection time comparison with other Algorithms

benign and malicious (attacked) data. The data we choose from the dataset belongs to three different devices which are Ecobee-thermostat, Webcam, and Security-camera. For training and optimization, we use tensorflow and keras libraries in python language. An auto-encoder makes an input layer whose dimension is the same as the number of features in the dataset, i.e. 115.

After training, we apply a famous DDOS attack known as (mirai) to calculate the detection time and accuracy of our model in comparison with other algorithms commonly used for anomaly detection. The same benign dataset is used to train three other algorithms: SVM (support vector machine), Isolation forest and LOF (Local Outlier Factor). Our method shows 99% results in terms of TPR (True Positive Rate) and fewer FPR (False Positive Rate). Furthermore, as evident in Fig. 5 SVM and LOF have almost similar TPR values and found much better than the isolation forest.

Next, we evaluate the average detection time for each algorithm as depicted in Fig. 6. The detection time of all the three devices in our case is lower than the others. The deep auto-encoders outer-perform on all the selected devices in terms of False-positive, True-positive and detection time. This is because of the ability in auto-encoders to learn approximate complex functions and non-linear structure mapping [25]. Moreover, as shown in Fig. 6, our technique required much less time than the other algorithms which is approximately 175230ms (milliseconds) to detect the attacks. This means that the launch attack could be detected or alerted in less than a second and thus considers as a substantial reduction in a typical time required for DDOS attacks [11].

6. Conclusions and Future Work

In this research, we analyze device level trust in IoT-Blockchain Infrastructure. A smart-home setting is used as a use-case for realizing the proposed idea. For prototype implementation a Local Blockchain on each zone is deployed on a master (raspberry pi-3) node that can store every traffic coming from their follower (Raspberry pi-0)) in the form of transactions. Behavior Monitor is defined and configured on the Main/Master node of each zone, which is capable of capturing and analyzing the runtime activity of IoT devices. We apply a deep learning strategy (auto-encoders) for realization on the behavior monitor to classify the device and make a level-of-trust. Furthermore, we incorporate Trusted Execution technology (TEE) as a root-of-trust over the blockchain to provide security for sensitive code and applications. Finally, the proposed framework could meet the current security problems in IoT-Blockchain environment. And the evaluation of our study shows its ability to mitigate the mainstream security requirements and resilience to attacks.

This research work is our first step towards classification of devices in IoT- Blockchain framework by means of deep learning.

Our future plan is to investigate a comparative study of other machine learning approaches for better results in terms of performances and accuracy. Another goal would be to realize the framework in other use-cases of IoT domain and analyze the outcomes. Finally, in the near future we will provide a full implementation on various IoT devices datasets along with full verification mechanism of zones in a trusted way and make the source online to the research community.

References

[1] Gartner Says By 2020, More Than Half of Major New Business Processes and Systems Will Incorporate Some Element of the Internet of Things. Technical report, Gartner, Inc, 2017. urlhttps://www.gartner.com/newsroom/id/3185623 (2017), [Online; accessed 08-April-2017]

[2] Mirai attack. urlhttps://www.corero.com/resources/ddos-attack-types/miraibotnet-ddos-attack.html (2017).

[3] ARM Trust Zone. urlhttps://www.arm.com/ products/security-on-arm/trustzone. (2018), [Online; accessed 25-Dec-2018]

[4] Intel SGX. urlhttps://software.intel.com/en-us/sgx (2018), [Online; accessed 25- Dec-2018]

[5] UCI Machine Learning Repository. urlhttps://archive.ics.uci.edu/ml/machinelearning-databases/00442/ (2018), [Online; accessed 03-Nov-2018]

[6] IOT Components. urlhttps://www.rfpage.com/what-are-the-major-componentsof-internet-of-things/ (2019), [Online; accessed 02-April-2019]

[7] Adnan, M., Eiad, D. Decentralizing Privacy Implementation at Cloud Storage Using Blockchain-Based Hybrid Algorithm p. 1-13

[8] Ali, J., Ali, T., Musa, S., Zahrani, A. (2018). Towards Secure IoT Communication with Smart Contracts in a Blockchain Infrastructure. *International Journal of Advanced Computer Science and Applications* 9 (10) 584 - 591. https://doi.org/10.14569/IJACSA.2018.091070, http://thesai.org/Publications/ViewPaper?Volume=9{&}Issue=10{&}Code=ijacsa{&}SerialNo=70

[9] Androulaki, E., Barger, A., Bortnikov, V., Cachin, C., Christidis, K., De Caro, A., Enyeart, D., Ferris, C., Laventman, G., Manevich, Y., Muralidharan, S., Murthy, C., Nguyen, B., Sethi, M., Singh, G., Smith, K., Sorniotti, A., Stathakopoulou, C., Vukoli_c, M., Cocco, S.W., Yellick, J. (2018). Hyperledger Fabric: A Distributed Operating System for Permissioned Blockchains. https://doi.org/10.1145/3190508.3190538, http://arxiv.org/abs/1801. 10228{%}0Ahttp://dx.doi.org/10.1145/3190508.3190538

[10] Banafa, A. (2017). Iot and blockchain convergence: Bene_ts and challenges. IEEE Internet of Things.

[11] Blenn, N., Ghinette, V., Doerr, C. (2017). Quantifying the spectrum of denial-of-service attacks through internet backscatter. *In*: Proceedings of the 12th International Conference on Availability, Reliability and Security. p. 21. ACM.

[12] Bocek, T., Rodrigues, B.B., Strasser, T., Stiller, B. (2017). Blockchains everywhere - a use-case of blockchains in the pharma supply-chain. 2017 IFIP/IEEE Symposium on Integrated Network and Service Management (IM) p. 772-777. https://doi.org/10.23919/INM.2017.7987376, http://ieeexplore.ieee.org/document/7987376/

[13] Christidis, K., Devetsikiotis, M. (2016). Blockchains and Smart Contracts for the Internet of Things. IEEE Access 4, 2292-2303. https://doi.org/10.1109/ACCESS.2016.2566339

[14] Darwish, M. A., Yafi, E., Almasri, A. H., Zuhairi, M. F. (2018). Privacy and Security of Cloud Computing : A Comprehensive Review of Techniques and Challenges 7, 239-246.

[15] Dorri, A., Kanhere, S. S., Jurdak, R. (2017). Towards an optimized blockchain for IoT. *In*: Proceedings of the Second International Conference on Internet-of-Things Design and Implementation. p. 173-178. ACM.

[16] Hammi, M. T., Hammi, B., Bellot, P., Serhrouchni, A. (2018). Bubbles of Trust : a decentralized Blockchain-based authentication system for. Computers & Security (July) (2018). https://doi.org/10.1016/j.cose.2018.06.004, https://doi.org/10.1016/j. cose.2018.06.004

[17] Hammi, M. T., Livolant, E., Bellot, P., Serhrouchni, A., Minet, P. (2017). A lightweight mutual authentication protocol for the IoT. *In*: International Conference on Mobile and Wireless Technology. p. 3-12. Springer.

[18] Hardjono, T., Smith, N. (2016). Cloud-based commissioning of constrained devices using permissioned blockchains.

In: Proceedings of the 2nd ACM International Workshop on IoT Privacy, Trust, and Security. p. 29-36. ACM.

[19] Hashemi, S. H., Faghri, F., Rausch, P., Campbell, R. H. (2016). World of empowered iot users. *In*: Internet-of-Things Design and Implementation (IoTDI), 2016 IEEE First International Conference on. p. 13-24. IEEE.

[20] Hinton, G. E., Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. Science 313 (5786), 504-507.

[21] Huh, S., Cho, S., Kim, S. (2017). Managing IoT devices using blockchain platform. In: Advanced Communication Technology (ICACT), 2017 19th International Conference on. p. 464-467. IEEE.

[22] Johnson, S., Scarlata, V., Rozas, C., Brickell, E., Mckeen, F. (2016). Intel R software guard extensions: Epid provisioning and attestation services. White Paper 1, 1-10 (2016)

[23] Komninos, N., Philippou, E., Pitsillides, A. (2014). Survey in smart grid and smart home security: Issues, challenges and countermeasures. *IEEE Communications Surveys & Tutorials* 16 (4) 1933-1954.

[24] Lee, B., Lee, J. H. (2017). Blockchain-based secure firmware update for embedded devices in an internet of things environment. *The Journal of Supercomputing* 73 (3) 1152-1167.

[25] Li, Y., Ma, R., Jiao, R. (2015). A hybrid malicious code detection method based on deep learning. methods 9 (5).

[26] Nakamoto, S. (2008). Bitcoin: A Peer-to-Peer Electronic Cash System. Www.Bitcoin.Org p. 9. https://doi.org/10.1007/s10838-008-9062-0, https://bitcoin.org/bitcoin.pdf

[27] Nauman, M., Tanveer, T. A., Khan, S., Syed, T. A. (2017). Deep neural architectures for large scale android malware analysis. *Cluster Computing*, p. 1-20.

[28] Nichol, P. B., Brandt, J. (2016). Co-creation of trust for healthcare: The cryptocitizen framework for interoperability with blockchain. Research Proposal. *ResearchGate* (2016)

[29] Ouaddah, A., Abou Elkalam, A., Ait Ouahman, A. Fairaccess: a new blockchain based access control framework for the internet of things. *Security and Communication Networks* 9 (18) 5943-5964.

[30] Ouaddah, A., Elkalam, A. A., Ouahman, A. A. (2017). Towards a novel privacy-preserving access control model based on blockchain technology in IoT. *In*: Europe and MENA Cooperation Advances in Information and Communication Technologies, p. 523-533. Springer.

[31] Roulin, C., Dorri, A. On the Activity Privacy of Blockchain for IoT

[32] Walker, M. A., Dubey, A., Laszka, A., Schmidt, D. C. (2017). Platibart: a platform for transactive iot blockchain applications with repeatable testing. *In*: Proceedings of the 4th Workshop on Middleware and Applications for the Internet of Things. p. 17-22. ACM.

[33] Wood, G. (2014). Ethereum: A secure decentralised generalised transaction ledger. Ethereum project yellow paper 151, 1-32.