SECBlock-IIoT: A Secure Blockchain-enabled Edge Computing Framework for Industrial Internet of Things

A. S. M. Sanwar Hosen
Department of Artificial Intelligence and Big Data, Woosong University, Korea
sanwar@wsu.ac.kr

Pradip Kumar Sharma
Department of Computing Science, University of Aberdeen, UK
pradip.sharma@abdn.ac.uk

Deepak Puthal
Department of Electrical Engineering and Computer Science, Khalifa University, UAE
deepak.puthal@ku.ac.ae

In-Ho Ra*
School of software, Kunsan National University, Korea
ihra@kunsan.ac.kr

Gi Hwan Cho
Division of Computer Science and Engineering, Jeonbuk National University, Korea
ghcho@jbnu.ac.kr

ABSTRACT

The IoT is widely used in a number of industries and generates large amounts of data. The data are processed, computed, and stored through distributed computing for analytical purposes. This invokes serious security and privacy concerns, and presents scalability issues. This paper describes a secure P2P and group communication supportive edge computing framework for IIoT systems, a consortium blockchain, and IPFS-based immutable data storage system, and an intelligent threat detection model to protect confidential data and identify cyber-attacks. Secure communications were ensured using a hybrid security scheme that included modified ECC, PUF, and Lagrange interpolation. We utilized a modified PoV consensus algorithm to resolve latency issues due to overhead and point of failure errors during block mining. The threat intelligence model used an autoencoder to transform data into a new format which was then fed into an RNN-DL to identify cyber-attacks. The model detected normal and anomalous activity, and then identified the category of detected malicious activity. We evaluated the framework according to various metrics and compared it with ECC, PoV, and ML-based classifiers. The results showed that the proposed system demonstrated a higher efficiency and improved scalability than conventional frameworks.

CCS CONCEPTS

• **Networking**; • **Distributed computing**; • **Network security**; • **Information storage system**; • **Intrusion/anomaly detection**;

KEYWORDS

Industrial internet of things (IIoT), Edge computing, Security and privacy, Blockchain, Intelligent threat detection (ITD)

ACM Reference Format:
A. S. M. Sanwar Hosen, Pradip Kumar Sharma, Deepak Puthal, In-Ho Ra, and Gi Hwan Cho. 2023. SECBlock-IIoT: A Secure Blockchain-enabled Edge Computing Framework for Industrial Internet of Things. In *Proceedings of the Third International Symposium on Advanced Security on Software and Systems (ASSS '23), July 10–14, 2023, Melbourne, VIC, Australia.* ACM, New York, NY, USA, 14 pages. https://doi.org/10.1145/3591365.3592945

1 INTRODUCTION

Distributed computing consists of various software components operating in geographically distant computers that are connected through a local or wide area network. The computing devices (i.e., Mist, Fog, Cloud) are placed at different physical or virtual layers and provide data processing, storage, access/retrieval, and analytics for individuals or industries [1], [2]. The industrial internet of things (IIoT) utilizes connected IoT devices, such as RFID tags, smart sensors, actuators, and smart meters, in manufacturing, agriculture, healthcare, transportation, power generation, and other fields, to increase productivity.

The data collected by IoT devices are often used to get the services offered with reliability, low latency, moderate service costs, minimum energy-consumption, and robust security and privacy protocols. The concerns can be alleviated using the edge computing which ensures (i) efficient task management, (ii) security and privacy resiliency, (iii) intelligent intrusion detection or prevention system, and (iv) on-chain or off-chain immutable data storage.

Several approaches have been introduced to enable the services and meet the objectives over the last few years. Efficient resource and task allocation, one of the primary objectives of edge computing, can be achieved by applying statistical techniques or artificial intelligence (AI), such as machine learning (ML) and deep learning (DL) [3], [4], [5]. Securing confidential data and user privacy are of paramount importance. To ensure the protection of user data among constrained heterogeneous IIoT devices, several encryption techniques have been studied [6], [7], [8] and utilized, such as ElGamal, physical unclonable function (PUF), elliptic curve cryptography (ECC), and zero trust security.

The significant increase of the number of IIoT devices that rely on the internet to exchange and control large amounts of data has necessitated heightened security measures. Typically, there

*Corresponding author

ASSS '23, July 10–14, 2023, Melbourne, VIC, Australia

ACM ISBN 979-8-4007-0182-5/23/07…$15.00
https://doi.org/10.1145/3591365.3592945

are two types of attacks: physical and cyber. Physical attacks are conducted by tampering with devices, injecting malicious codes, and through falsified node injection. While physical attacks do occur, cyber-attacks are much more prevalent and have the potential to be considerably more damaging and include denial-of-service (DoS), distributed DoS (DDoS), man-in-the-middle (MitM), and ransomware attacks. Intrusion detection systems (IDS) or threat intelligence (TI) improves cybersecurity by examining network traffic for abnormal activity and by accumulating knowledge regarding different types of threats [9].

Data that is stored locally, on-chain, or off-chain needs to facilitate data-oriented services with confirming data unalteration and denying unauthorized access. To meet these requirements, blockchain technology and the InterPlanetary file system (IPFS) can be used concurrently. Blockchain is a decentralized data storage system where data are listed on a verified block and added into a blockchain. The blockchain is recorded using ledgers on the nodes in a blockchain network. In contrast, the IFPS uses a content identifier (CID) to uniquely identify each file in a global namespace that connects devices.

1.1 Motivation and Key Challenges

A unified IIoT-enabled edge computing system requires a fine-tuned uninterrupted flow-based integration and an interoperable framework. The key features of an interoperable framework are secure point-to-point (P2P) and group communications among the devices across the layers, user authentication, data confidentiality, a proactive security mechanism exhibiting a high level of attack detection accuracy, access availability, and finally, the ability to ensure on-chain or off-chain unalterable and authorized data storage. These features can be implemented using state-of-the-art edge computing, AI-based TI, blockchain, and the IPFS.

Data flow of an edge computing network requires P2P and group communications to make the applications more functional and effective. It needs lightweight and supportive cryptography that should be well-suited to minimizing complexity without compromising security. A number of researchers have proposed the use of tightly-coupled security algorithms such as AES, DES, ElGamal, and ECC, with or without modification, where a substantial part of the network is in close-contact with energy, storage, and computationally constrained IIoT-devices. Full-fledged time-complex cryptography is often inapplicable to communications across layers with ensuring efficiency when various evaluation metrics are taken into consideration. This has encouraged researchers to design an interactive and secure communication system featuring a hybrid security scheme, including a private session key derived via PUF and shared using modified ECC at the edge layer, as this layer is more impactful due to its heterogeneous nature.

Most TI cybersecurity models are based on conventional statistical and ML tools. However, these models are ineffective against dynamic threats that are complex and highly non-linear. This may result in lower threat detection accuracy, a higher false alarm rate, and lack of generalization ability. In contrast, DL, a subset of ML, allows a system to learn from unstructured and diverse data sets, and to extract hidden influenced features. Thus, DL can be used to develop adaptive TI models. To protect confidential data and prevent data inference, DL with essential layers reduces data dimensionality to a variable-length through a transformation by training multiple neural networks.

Blockchain and an edge-computing integrated domain can make an entire IIoT network secure and cost-effective by avoiding middlemen-vendors and third-party providers, using cryptographically generated blocks recorded on-chain [10], [11]. It provides immutability, transparency, security, and privacy against various attacks involving false data injection or data poisoning. As IIoT data should be kept confidential in most applications (i.e., healthcare, power grid, federated AI, etc.) and relevant to use by multiple interested beneficiaries, a consortium blockchain can be used. Consortium blockchain is governed by multiple organizations with similar interests through a permission, instead of being controlled by a single organization which lacks the decentralized features of the technology. Data storage limitations, a major impediment of implementing blockchain at the edge layer, can be resolved using IPFS.

1.2 Key Contributions

To accumulate the modules and address the challenges, we designed and implemented a secure blockchain-enabled edge computing framework for IIoT called SECBlock-IIoT. The goal of this study was to develop a sustainable IIoT and edge computing integrated network by combining secure P2P and group communication, an interactive ITD module, and blockchain based on-chain and IPFS based off-chain immutable data storage. The key contributions of this work can be summarized as follows:

- A secure integrated edge computing and IIoT network was designed that implements an arbitrary PUF to generate a session key for P2P communication, and the Lagrange interpolation algorithm to generate a group session key for group communications, and ECC is used to share the secret session keys. In addition, a certificate-based authentication technique was adopted to confirm an authorized participation in the network.
- A two-fold privacy and security data assurance technique was using DL-based TI and blockchain technology. The ITD module uses autoencoder (AE) recurrent neural network (RNN) for attack detection across two levels. In Level-1, the network-flow data are classified as normal or anomalous and anomalous data are then forwarded to Level-2 for further subcategorization of the malicious activity.
- Off-chain IFPS was used for blockchain storage at edge layer to overcome the data storage limitation of the constrained devices.

The rest of the paper is organized as follows. Section 2 discusses the advancement of the IIoT-enabled edge computing and related works in the literature. Section 3 presents the proposed framework. Section 4 includes the security resiliency of the proposed framework. The experimental results and analysis are given in Section 5. Section 6 concludes the work.

2 RELATED WORK

This section discusses the development of edge computing for IIoT networks.

Integrated edge computing and blockchain systems have several shortcomings, such as decentralized management, security, and scalability [12], [13]. A two-level network flow-based anomalous activity detection technique was been proposed by Ullah et al. to improve the security of IoT networks [14]. This robust anomaly detection system utilizes two levels: Level-1 categorizes the network flow as normal or abnormal, and forwards the results to Level-2 where the detected malicious activity is subcategorized. The subsequent IDS model proposed by Ullah et al. also operated using two levels: Level-1 used a decision tree (DT) and Level-2 used a random forest (RF) classifier on the extracted flow-based features of a IoT-Botnet dataset [15]. This achieved higher anomaly detection accuracy than their previous model. A trustworthy privacy preserving secured framework for smart cities utilizing blockchain and ML was developed by Kumar et al. [16]. Their two-level privacy module implemented a blockchain-based enhanced proof-of-work (ePoW) consensus algorithm, and transformed data using principal component analysis (PCA). The authors applied the XGBoost classifier algorithm for multivariate classification of IoT-Botnet and ToN-IoT datasets [17], and their ITD model achieved a higher threat detection rate while ensuring privacy. A number of studies have examined the applicability of ML and DL techniques to IDS in IIoT networks [18] [19]; however, this study did not evaluate whether the data sources were reliable, which is important for ensuring the quality of data fed to the IDS. The authors in [20], [21] presented an efficient IDS based on ML for IIoT networks. The aforementioned models applied lightGBM and RF algorithms for intrusion detection on the manually extracted features, which requires a high level of expertise to label the threats.

The studies performed by Alkadi et al. [22] and Liang et al. [23] discuss blockchain-based data immutability, data storage, data traceability, and data sharing between participants. Keshk et al. developed a security framework that applied a variational autoencoder and blockchain technology to ensure privacy, and long short-term memory (LSTM) to improve IDS [24]. However, these studies did not analyze block creation or access time of their proposed consensus algorithm regarding various transactions on smart power networks. In another publication, Alkadi et al. emphasized the benefits of integrating blockchain with IDS in a deep blockchain framework, but did not perform blockchain specific implementation and evaluation [25].

Previous studies have largely concentrated on either defensive IDS/TI models with or without data transformation, or a blockchain integrated with existing consensus mechanisms oriented to a public blockchain. These blockchain systems are unsuitable in time-sensitive applications such as smart healthcare, federated learning in edge computing, drone security [26], and energy trading [27]. To remedy these deficiencies, Li et al. proposed a consortium blockchain with a balanced trade-off between performance and security using a proof-of-vote (PoV) consensus algorithm [28]. The drawbacks of the PoV algorithm are that the butler (miner) selection process is not independent, and each broadcast of butler information selected by a commissioner involves a communication overhead and a point of failure caused by internal and external interventions. In addition, network communications are P2P/peer-to-peer and group communication is not allowed among the participants, even though group communication is often required for effective collaboration and decision making. To address these challenges, we developed a blockchain-enabled edge computing for IIoT networks.

3 PROPOSED SECBLOCK-IIOT FRAMEWORK

This section describes the SECBlock-IIoT framework, its components, and functionality.

3.1 System Architecture

The SECBlock-IIoT framework was designed to ensure the privacy and security of IIoT data and tasks in an edge computing network. The system is composed of four functional modules: the IIoT-terminal layer, the edge service management layer (ESML), the blockchain-based data storage module (BDM), and the DL-based intelligent threat detection (ITD) module (Figure 1). The IIoT-terminal layer contains smart devices which monitor, control, and examine different industry 4.0 applications. ESML utilizes user local hosts (ULHs) (i.e., an industrial computer providing local data acquisition and the pre-processing point of the IIoT-terminal layer), and edge computing nodes as data storage and tasks execution hubs. An edge service management hub (ESMH), which manages ESML, receives tasks from users and distributes the tasks to computing nodes [2]. The ESMH facilitates data storage, access/retrieve service management, and task execution. The communication between IIoT-nodes and ULHs, and between ULHs and the edge server/manager, can be grouped or individualized depending on the context of the services. A particular result (and accompanying data) of a task may be required by an interested group of users through ULHs, which requires group communication. On the other hand, confidential data/results should only be available to those who require access or execute by a single user that requires a P2P communication.

The BDM includes a consortium blockchain, a consensus algorithm, and IPFS to process unalterable data record, policy, agreement, and support data storing and sharing system. In the blockchain, transactions are recorded and added into blocks that vary according to the type of service: (i) IIoT-node data are stored, accessed, and retrieved from edge server, and (ii) IIoT-node tasks are sent to the edge server for execution. The classified blocks are then added into the blockchain. Once an IIoT-node task is completed and the task manager receives the result from the assigned computing node(s), a transaction record is generated that contains the *source ID*, *task ID*, *computing node ID*, *task*, *results*, and a *timestamp*. In practice, a blockchain is an immutable distributed ledger of transactions and can be stored on-chain of blockchain or off-chain of IPFS.

The proposed framework incorporates an ITD module using an AE-RNN-based DL technique. AE is an unsupervised artificial neural network that compresses and encodes data into a variable-length latent space and reconstructs data back to the original input as possible. AE reduces data dimensionality by extracting hidden features and preserves data confidentiality by defending against inference attacks.

3.2 Blockchain-enabled Secure Communication

The SECBlock-IIoT consists of three phases: system initialization phase, registration phase, and validation phase.

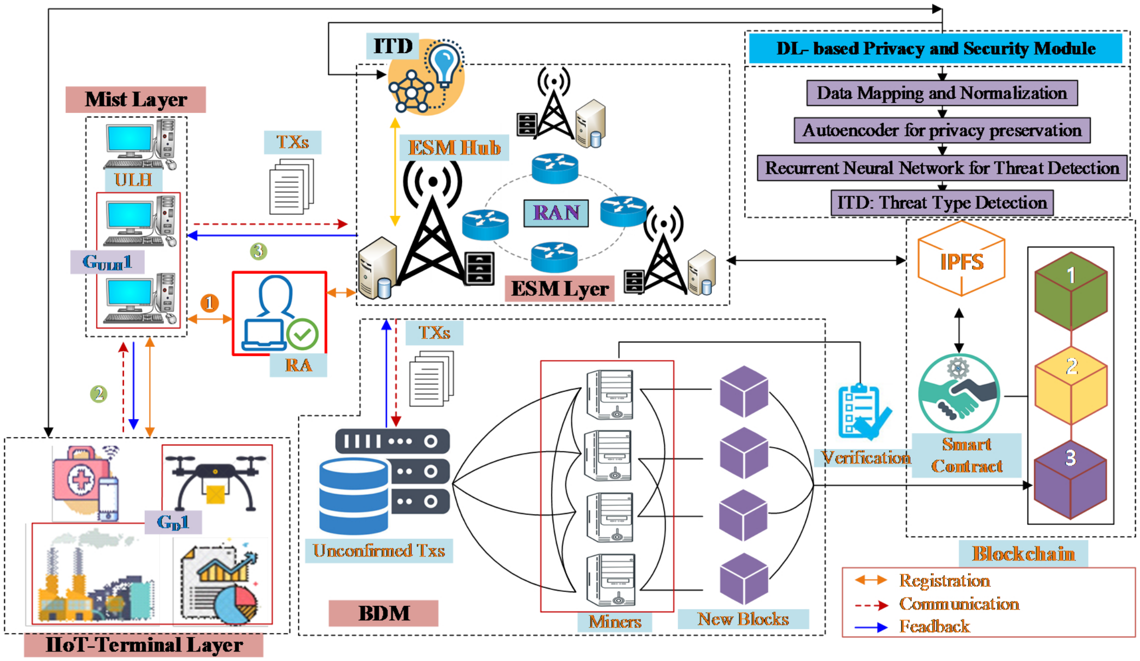


Figure 1: An overview of the architecture of the secure edge computing framework for industrial internet of things (SECBlock-IIoT).

3.2.1 *System Initialization.* In this phase, a trusted registration authority (RA) with a unique identity (ID_{RA}) selects the parameters which support single and group communications among the nodes. To begin, the RA selects a non-singular elliptic curve over a finite field $\mathbb{F}_{p\neq 2,3}$ that satisfies the equation $E(a,b) = x^3+ax+b\ (mod\ p)$, where p is a large prime number, $(a,b) \in \mathbb{F}_p^2$ are the coefficients, and $4a^3+27b^2 \neq 0$ is the discriminant. The RA also selects a generator $(G \leq p)$, a private key $(Pr_k^{RA} \in \mathbb{F}_p^*=\{1,2,3,\ldots,p-1\})$, and computes a public key $(Pb_k^{RA}=Pr_k^{RA}G)$. RA then publishes the parameters $(\{E(a,b),\ G,\ Pb_k^{RA}, h(.)\})$ on the public directory. Suppose that $A(x_A, y_A) \neq B(x_B, y_B)$ are two points on given elliptic curve $E(a,b)$, such that the line (L) through (A,B) is not tangent to E at either A or B, and that L intersects E at a third point $R(x_R, y_R) \neq A, B$. The proposition of points addition $R = A + B$ is evaluated as $x_R = m^2 - x_A - x_B\ (mod\ p)$ and $y_R = m(x_A - x_R) - y_A\ (mod\ p)$, where the slope $(m) = (\frac{3x^2+a}{2y})\ (mod\ p)$ if $A = B$, otherwise $m = \frac{y_B - y_A}{x_B - x_A}\ (mod\ p)$. The scalar point multiplication is calculated using the repetition of point doubling and addition operations as, $n \times (A,B) = (A,B)+(A,B)+\ldots+\ (A,B)_n$.

3.2.2 *Registration Process.* The registration of an ESMH is processed by the RA through a secure channel, and registration of ESML-nodes (i.e., Fog, Mist, and other edge service providers) is processed by an associate ESMH. The registration process varies according to P2P and group communications (discussed below) and the data encryption and decryption process are provided in Algorithm 1 and 2, respectively.

Edge Service Management Hub (ESMH) Registration. ESMHs include ULHs and edge service managers. The registration of an ESMH proceeds according to the following steps:

P2P Communication: Prior to registration, each ESMH, MH_j chooses a private and public key $(Pr_k^{MH_j} \in \mathbb{F}_p^*, Pb_k^{MH_j} = Pr_k^{MH_j}G)$ and sends the information $\{SID_{MH_j}, Pb_k^{MH_j}, SID_{G\{.\}}, t_{stamp}\}$, encrypted using Pb_k^{RA}, decrypt support vector kG, and a digital signature sig_{MH_j}, to the RA, where $SID_{G\{.\}}$ refers to the interested nodes pseudo-IDs of a group and $k \in \mathbb{F}_p^*$ is a secret random integer. The pseudo-IDs set of a group is null as a single entity and is recorded during the registration. The RA receives the encrypted information and decrypts it using the private key Pr_k^{RA}. After the RA verifies the credentials, it generates a certificate Crt_{MH_j} corresponding to the node. The RA then publishes the information $\{SID_{MH_j}, Pb_k^{MH_j}, Crt_{MH_j}, E(a,b),\ G,\ h(.)\}$ on the public directory. The certificate of each MH_j is generated as defined in Equation 1).

$$Crt_{MH_j} = Pr_k^{MH_j} + h\left(SID_{MH_j}\left\|Pb_k^{MH_j}\right.||ID_{RA}||\,Pb_k^{RA}\left\|t_{stamp}\right.\right) \times Pr_k^{RA}\ \ (mod\ p) \tag{1}$$

Where $h(dot)$ is the collision free hash function (i.e., sha512) and t_{stamp} is the registration time.

Group Communication: For this, assume nodes n with the same interest are grouped to participate in group communication. For registration as a group, RA provides a pseudo group identification $SGID_{MH_{j:n}}$ of the nodes. To generate a shared group private key, the Lagrange interpolation $\mathcal{L}_n(x)$ algorithm is utilized. The RA begins by mappings n points $\mathcal{P}_n = \{\mathcal{P}_0(x_0, y_0)\ldots\mathcal{P}_n(x_n, y_n)\} \in$

$E(a, b)$ $(mod\ p)$ corresponding to the interested nodes and applies the interpolation formula defined in Equation 2).

$$\mathcal{L}_n(x) = \sum_{i=0}^{n} \mathcal{L}_i(x) f(x_i) \quad (mod\ p) \tag{2}$$

Where $(x_i) = y_i$ and $\mathcal{L}_i(x) = \prod_{\substack{j=0 \\ j \neq i}}^{n} \frac{x - x_j}{x_i - x_j}$ $(mod\ p)$.

The RA then generates the private key associated with each node as $Prs_k^{MH_j} = \mathcal{L}_i(x = r)$ $(mod\ p)$, where $r \in \mathbb{F}_p^*$ is a chosen random number based on the agreement of a group. RA also generates a group public key $GPb_k^{MH_{j:n}} = G \sum_{j=1}^{n} Prs_k^{MH_j}$ and a group certificate defined in Equation 3).

$$GCrt_{MH_{j:n}} = \sum_{j=1}^{n} Pr_k^{MH_j} \\ +h\left(\left\{SID_{MH_j} \,||\ldots||\, SID_{MH_n}\right\} ||ID_{RA}||\, Pb_k^{RA} \left\|GPb_k^{MH_{j:n}}\right\| t_{stamp}\right) \\ \times Pr_k^{RA} \ \ (mod\ p) \tag{3}$$

The generated private keys are assigned to each node in the group and the keys are deleted from the RA database for security reasons. The RA then shares the information $\{SGID_{MH_{j:n}}$, $GPb_k^{MH_{j:n}}, ID_{RA}$, $Pb_k^{RA}, GCrt_{MH_{j:n}}, E(a, b), h(.), G\}$ on the public directory before deployment of the ESMHs.

Edge Service Management Layer (ESML) Registration. Each ESML-node $ESML_i$ is registered via their associate ESMH as a single or group entity, as previously described. During single entity registration, each ESML-node sends the $SID_{G\{.\}}$ and derived public key $Pb_k^{ESML_i}$ to the ESMH. The ESMH generates the certificate on to the ESMH. The ESMH generates the certificate once the provided information is verified. The generation of certificate of interested m nodes in a group is based on the Lagrange interpolation, where the group public key is expressed as $GPb_k^{ESML_{i:m}}$=$G \sum_{i=1}^{m} Prs_k^{ESMLH_i}$. Once the single or group certificate (Crt_{ESML_i}/ $GCrt_{ESML_{i:m}}$) generation is completed, MH_j publishes the credentials $\{SID_{ESML_i}/SGID_{ESML_{i:m'}}, Pb_k^{ESML_i}/Pb_k^{ESML_i}$, SID_{MH_j}, Pb_k^{MHj}, $Crt_{ESML_i}/GCrt_{ESML_i}$, $E(a, b), G, h(\cdot)\}$ on the public directory. The certificates of a single or group entity are generated as defined in Equation 4) and 5).

$$Crt_{ESML_i} = Pr_k^{ESML_i} \\ +h\left(SID_{ESML_i} \left\|Pb_k^{ESML_i}\right\|SID_{MH_j}\right\| Pb_k^{MH_j} \Big\| t_{stamp}\Big) \\ \times Pr_k^{MH_j} \ \ (mod\ p) \tag{4}$$

$$GCrt_{ESMLm} = \sum_{i=1}^{m} Prs_k^{ESML_i} \\ +h(\{SID_{ESML_i} \,||\ldots||\, SID_{ESML_m}\} \left\|SID_{MH_j}\right\| GPb_k^{ESML_{i:m}} ||t_{stamp}) \\ \times Pr_k^{MH_j} \ \ (mod\ p) \tag{5}$$

IIoT-Terminal Layer Registration. This layer comprises IIoT-devices that collect and generates data in industry 4.0 systems. The registration of IIoT-devices is processed by an associate ULH via a secure channel. Due to constraints of IIoT devices, particularly energy and computational limitations, we applied a hybrid security scheme that utilizes an ECC-based asymmetric public key to share a unique symmetric key to encrypt and decrypt data during a session. The shared session key $S_k^{D_i}$ is generated on the devices PUF and is shared to the ULH using the ULH's public key. This approach addresses the time-complexity and energy-consumption issues of edge layer devices that use conventional security algorithms.

Algorithm 1 Encryption and decryption of IIoT-devices and ULH(s)

```
Input: $SID_{ULH}$,$Crt_{ULH}/GCrt_{ULH}$, $Pb_k^{ULH}/GPb_k^{ULH}$,$S_k^D$, $GS_k^D$
/* P2P communication
for each $D_i$ do
    $\{C_1 = Pm + k_{D_i}Pb_k^{ULH}, C_2 = k_{D_i}G)\} \longrightarrow SID_{ULH_j}$ /*$S_n$ is reference to the session number
    $|Pm = (SID_{D_i}||S_k^{D_i}||S_n||t_{stamp}||sig_{D_i})p(x_i, y_i) \neq G \in E(a, b)$
    $D : Pm + k_{D_i}Pb_k^{ULH_j} - k_{ULH_j}C_2$
    $\{E : (S_k^{D_i}, (PT)), sig_{D_i}\} \longrightarrow SID_{ULH_j} \mid PT = (SID_{D_i}||data||t_{stamp})$
    $ID_{ULH_j} : D : (S_k^{D_i}, E(S_k^{D_i}, (data),)$ & verifies $sig_{D_i}$
end
/* Group communication
for each $D_i \in SGID_{ULH_{j:m}}$ do
    $\{C_1 = Pm{+}k_{D_i}GPb_k^{ULH_{j:m}}, C_2 = k_{D_i}G\} \longrightarrow SID_{ULH_{j:m}}$ /*multicasts
    $| Pm = (SID_{D_i}||GS_k^{D_i}||S_n||t_{stamp}||sig_{D_i})$ $p(x_i, y_i) \neq G \in E(a, b)$
        for each $\{ULH_1, \ldots, ULH_m\} \in GULH_{i:m}$ do
            $\{C_1' = Pm' + k_{ULH_i}GPb_k^{ULH_j}, C_2' = k_{ULH_i}G\} \longrightarrow ID_{ULH_{j \neq i}}$
            | $Pm' = (SID_{ULH_i}||k_{ULH_i}C_2||S_n||t_{stamp}||sig_{ULH_i})$
                $p'(x_i, y_i) \neq G \in E(a, b)$
            for each $ULH_j$ do
                $D : Pm = C_1' - (k_{ULH_i}GPb_k^{ULH_j} = \sum_i^m k_{ULH_i}C_2')$
            end
        end
    $ID_{D_i} : \{CT = E : (GS_k^{D_i}, (PT)), sig_{D_i}\} \longrightarrow SID_{ULH_{i:m}}$
    $ID_{ULH_i} : D : (GS_k^{D_i}, (CT))$ & verifies $sig_{D_i}$
end
Output: $(S_k^{D_i}, PT)$
```

Several PUF-based key generation methods have been introduced in the last few years [29]. The proposed SECBlock-IIoT framework utilizes the Arbiter PUF which is considered more efficient for IIoT-devices [30]. To implement the Arbiter PUF, we assumed that the devices are field-programmable gate array (FPGA) enabled. FPGAs are semiconductor devices that are composed of a matrix of configurable logic-blocks connected via programmable interconnects. The shared secret keys are changed periodically.

During their registration, each IIoT-device generates a unique pseudo-ID using PUF, and produces a key pair ($Pr_k^{D_i} \in \mathbb{F}_p^*$, $Pb_k^{D_i}$ =

Algorithm 2 Encryption and decryption of ULH/ESML and ESMH

```
Input:
$SID_{MH}/SGID_{MH_{j:n}}, Pb_k^{MH}/GPb_k^{MH_{j:n}}, Crt_{MH}/GCrt_{MH_{j:n}}$
/* P2P communication
for each $SID_{ESMLi}$ do
    $\{C_1 = Pm + k_{ESML_i} Pb_k^{MH_j}, C_2 = k_{ESML_i} G)\} \longrightarrow SID_{MH_j}$
    $| Pm = (SID_{ESML_i} || S_n || t_{stamp} || data || sig_{ESML_i})$
            $p(x_i, y_i) \neq G \in E(a, b)$
    $ID_{MH\,j} : D : Pm = (Pm + k_{ESML_i} Pb_k^{MH}) - k_{MH_j} C_2$
end
/* Group communication
for each $ESML_i \in SGID_{MH_{j:n}}$ do
  $\{C_1 = Pm + k_{ESML_i} GPb_k^{MH_{j:n}}, C_2 = k_{ESML_i} G\} \longrightarrow SID_{MH_{j:n}}$
    for each $SID_{MH_j}$ do
        $\{C_1^{'} = Pm^{'} + k_{MH_j} Pb_k^{MH_i}, C_2^{'} = k_{MH_j} G\} \longrightarrow SID_{MH_{i \neq j}}$
$| Pm^{'} = (SID_{MH_j} || k_{MH_j} C_2 || S_n || t_{stamp} || sig_{ESML_i})$
                $p'(x_j, y_j) \neq G \in E(a, b)$
        for each $SID_{MH_i}$ do
            $D : Pm^{'} = (Pm^{'} + k_{MH_j} Pb_k^{MH_i}) - k_{MH_i} C_2^{'}$
        end
    end

$ID_{MH_j} : D : Pm = (Pm + k_{ESML_i} GPb_k^{MH_{j:n}}) - k_{ESML_i} GPb_k^{MH_{j:n}}$
$| k_{ESML_i} GPb_k^{MH_{j:n}} = \sum_i^n k_{MH_i} C_2^{'})$
end
Output: $PT$
```

$Pr_k^{D_i} G)$ using the credentials of the associate ULH, as published by the RA. Each device then provides the pseudo group-ID ($SID_{G\{.\}}$) and generated public key to the ULH. According to the information provided by the single or group entity, the ULH generates the certificates Crt_{D_i} and $GCrt_{D_i}$, where the $GPb_k^{D_{i:w}} = G \sum_{i=1}^{w} Pr_k^{D_i}$ of w nodes is derived using Lagrange interpolation and ECC addition and doubling operations. Once the registration is completed, the ULH shares the information $\{SID_i$, $Pb_k^{D_i}/GPb_k^{D_{i:w}}$,$SID_{ULH}$, $Crt_{ESML_i}/GCrt_{ESML_i}$ $E(a, b)$, $G\}$ on the public directory. The devices are authorized through challenges and responses when starting a new communication session [10]. The single and group certificates are generated as defined in Equation 6) and 7).

$$
\begin{aligned}
&Crt_{D_i} = Pr_k^{D_i} \\
&+h\left(SID_{D_i} \left\| Pb_k^{D_i} \right\| SID_{MH_{j/ULH}} \left\| Pb_k^{ULH} \right\| t_{stamp}\right) \\
&\times Pr_k^{ULH} \ (mod\ p)
\end{aligned}
\tag{6}
$$

$$
\begin{aligned}
&GCrt_{D_{i:m}} = \sum_{i=1}^{m} Prs_k^{D_i} \\
&+h(SID_{D_i} || \ldots || SID_{D_m} || SID_{ULH} || GPb_k^{D_{i:m}} || Pb_k^{ULH} || t_{stamp}) \\
&\times Pr_k^{ULH} \ (mod\ p)
\end{aligned}
\tag{7}
$$

3.3 Validation and Block Generation Module

The IIoT-nodes through the ESML-nodes interested to join the blockchain network starts after the successful registration. The verification is processed by the ESMHs. After the nodes are successfully verified using the available credentials (i.e., public key, certificate, and timestamp), they are permitted to store data on the blockchain and is confirmed via a successful/unsuccessful message. The data/transactions of an ESML-node are stored in a $block_i$, verified through the federated nodes $FN = \{fn_1, fn_2, \ldots, fn_n\}$ joined the consortia network applied modified-PoV (mPoV) consensus algorithm. In the proposed mPoV consensus algorithm, a consortium committee is formed among the ESMH and ESML-nodes. A commissioner, C_i, has the right to recommend, vote for, and evaluate the butlers $B = \{B_1, B_2, \ldots, B_N\}$. Butlers are miners that specialized in generating blocks.

Unlike the butler selection in conventional PoV algorithm, in mPoV, they are independently and randomly self-appointed based on a function that generates a random number R^n in the interval between 0 and 1, and compares to the threshold number Th^{fn} given in Equation 8). A butler whose random number is less than or equal to the threshold ($R^n \leq Th^{fn}$) is elected independently as a duty butler to gather transactions, pack into a block, and sign it by the consensus rules. The butlers are also permitted to verify blocks, and forward both blocks and transactions. A block must collect at least $\frac{C_N}{2} + 1$ signatures from different commissioners to become a valid block within a time bound (T_b). $B_N + 1$ rounds and $B_N + 1$ blocks are generated in a tenure. A $(B_N + 1)^{\text{th}}$ block is a special block used to record the information of the B_N self-elected candidates. Once a special block is generated, the current butlers are officially retired for the next round as duty nodes.

$$
Th^{fn} = \begin{cases} \dfrac{P^d}{1-P^d\left[mod\left(r, \frac{1}{P^d}\right)\right]} & if\ \ fn \in\ G' \\ 0 \qquad Otherwise \end{cases}
\tag{8}
$$

Where P^d is the desired percentage of duty butlers depending on the uncertainty factors of the network, r is the current round, and G' is the set of FN that have not been duty butlers in the last $\frac{1}{P^d}$ rounds in a tenure.

Through ULH, and ESMH, or other ESML-nodes (as ordinary users), create transactions with their signature and forward the valid transactions to commissioners and butlers to store into their local transaction pool. An elected butler B_N selects transactions from the pool, add them into a pre-block with a timestamp, and sends the pre-block to the commissioners for verification. A commissioner receives a pre-block(s), verifies the earliest one according to the piggybacked timestamp and sends pre-block back with a signature to the selected one. The pre-block with the earliest one is selected. Once the duty butler receives the required number of signatures, it orders them based on the timestamps of the signatures, and attaches them to the pre-header of the block. After generating a valid block, the butler calculates the $R - value$ ($R = getPrevious$-$BlockRandomNum$()) and the block time to form the final-header that is sent to the commissioners and other nodes. When more than 50% commissioners confirm the receipt of the valid block, the block receives final confirmation. After received a valid block, the added transactions are deleted from the transaction pool. Nodes with limited storage are allowed to record the blocks on the IPFS. This addresses the storage limitation of edge nodes by recording the CID instead of storing the blockchain in conventional manner. This method solves network scalability problems. The details of how mPoV-based general block are generated and added into the blockchain are provided in Algorithm 3.

Algorithm 3 Generating a valid block using mPoV

```
Input: $ID_{D_i}/GID_{D_i}$, transaction, index, timestamp, previous has, nonce
for $transaction \in \{SID_{D_i}, SGID_{D_i} \rightarrow ULH, SID_{ESMLi} \rightarrow ESMH\}$ do
    $verifies: Crt_{D_i}/\ GCrt_{D_i} \& \ sig_{D_i}$
    $status: (sucessfull/unsucessfull) \rightarrow (SID_{D_i}/SID_{ESMLi})$
    if $status ==$ 'successful' then
        $transaction \rightarrow FN$
    end
end
for $B_i \in G$ do
    $pre_{block\ i} \leftarrow \{index, timestamp, previous\,hash, nonce, transactions\}$
    $pre\_block_i \rightarrow C$
    $count\_sig = 0$
    for $pre\_block_i \in C$ do
        $pre\_block_i$(varified)$||sig_{C_i}||timestamp \rightarrow B_i$
        $count += 1$
        if $count \geq \frac{C_N}{2} + 1$ then
            $block_i \leftarrow pre\_block_i ||\ sort(sigs)$
            $block_i \rightarrow FN$
            $IPFS \leftarrow hash(block_i)$
        else
            continue
        end
    end
end
```

3.4 Intelligent Threat Detection (ITD) Module

After the successful execution of the mPoV algorithm, and the network is in operation, the DL module (Figure 2) converts the network flow data into a new format. The encoded data is used by the ITD-module in two-steps: first, the network traffic information is encoded using an AE and is then transferred to the ITD-module for further threat detection. The AE compress the higher dimensional data into a latent space $D = \{\vartheta_i\}_{i=1}^{S}$, includes S records with ϑ features, and decompresses the encoded data into the original one with minimum loss $L = |x - \hat{x}|$, where x and $\hat{x}$ are the ground truth and predicted output, respectively. The encoded data is protected from inference attacks and data manipulation.

The obtained encoded data is then fed into the RNN model. The RNN is highly accurate at predicting output of sequential and timeseries data. The proposed AE-RNN consists of input layers (latent space of the AE), hidden layers, and output layers. The RNN maps the ϑ_i values to a corresponding sequence of output O values. L measures the actual output (y) and the predicted output (O). It has also input to hidden layer parameterized by a weight matrix U, hidden to hidden layers parameterized by a weight matrix W, and hidden to output layers parameterized by a weight matrix V. Then time step $t = 1$ to $t = n$, the following Equation 9), 10), 11), and 12) are applied.

$$a^{(t)} = b + Wh^{(t-1)} + Ux^{(t)} \tag{9}$$

$$h^{(t)} = \tanh\left(a^{(t)}\right) \tag{10}$$

$$O^{(t)} = c + Vh^{(t)} \tag{11}$$

$$\hat{y}^{(t)} = sigmoid\left(O^{(t)}\right) \tag{12}$$

4 SECURITY ANALYSIS

SECBlock-IIoT resiliency against various attacks is discussed in this section.

4.1 Authentication

The registration is conducted using two processes: registration of ESMH through RA, and registration of IIoT-nodes and ESML-nodes through ESMHs. In both methods, the authentications are executed using their credentials (i.e., certificates, signatures, and timestamps) which are encrypted using corresponding public keys. Afterwards, the IIoT-nodes are authenticated via challenges and responses using their shared PUF-derived secret keys $S_k^{D_i}$ which are sent confidentially (encrypted using the public key Pb_k^{ULH} during every session). These processes ensure that the certificates cannot be forged, that the shared secret keys of ULH and IIoT-nodes remain hidden, and that an adversary cannot receive messages from IIoT and ESML-nodes within the time frame.

4.2 Preserving Privacy

Messaging between participants is encrypted using public keys of the authorities in the upper layers and PUF-based unique symmetric keys in the edge layer. The keys are updated frequently. The unclonable pseudo-IDs and their single or group certificates published on the directory do not reveal the actual identity of the nodes, thereby preserving their privacy.

4.3 Replay Attack

A data generator node or a computing node generates a message within a certain time-interval prescribed by the timestamp piggybacked in a message along with a certificate. The recipient nodes (ULHs, ESMHs) verify the message by confirming the certificate, digital signature, and timestamp. This process prevents sending messages from any unauthorized nodes which thwarts replay attacks.

4.4 Man-in-the-Middle Attack (MitM)

An attacker may intercept an encoded message transmitted through an insecure channel in order to inject malicious data. To prevent this, the public keys and secret session keys used must be revealed which is computationally infeasible in a limited amount of time.

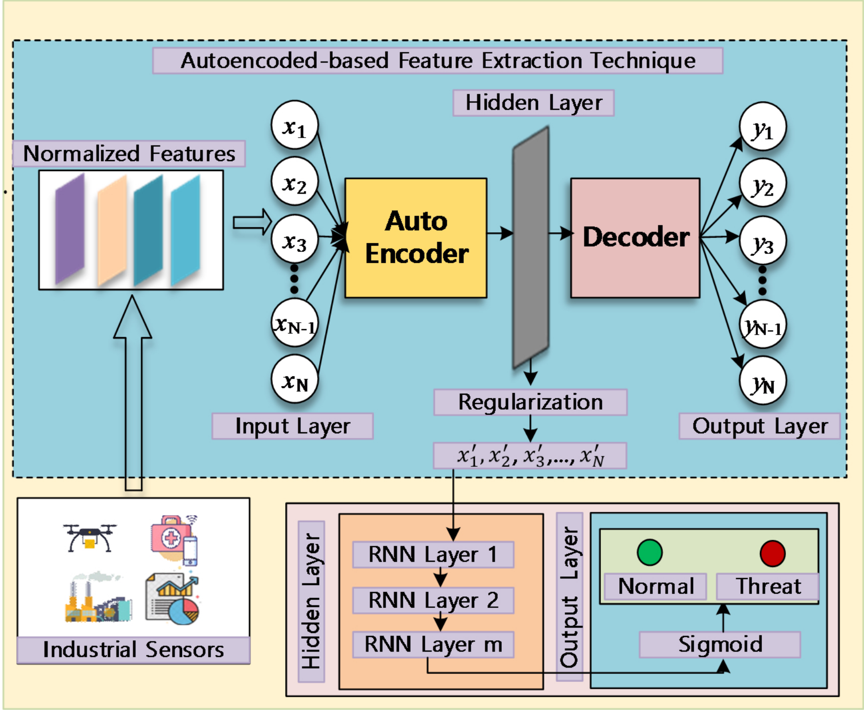


Figure 2: Deep learning architecture of the proposed intelligent threat detection (ITD) model.

Table 1: Parameters and values used in the simulation

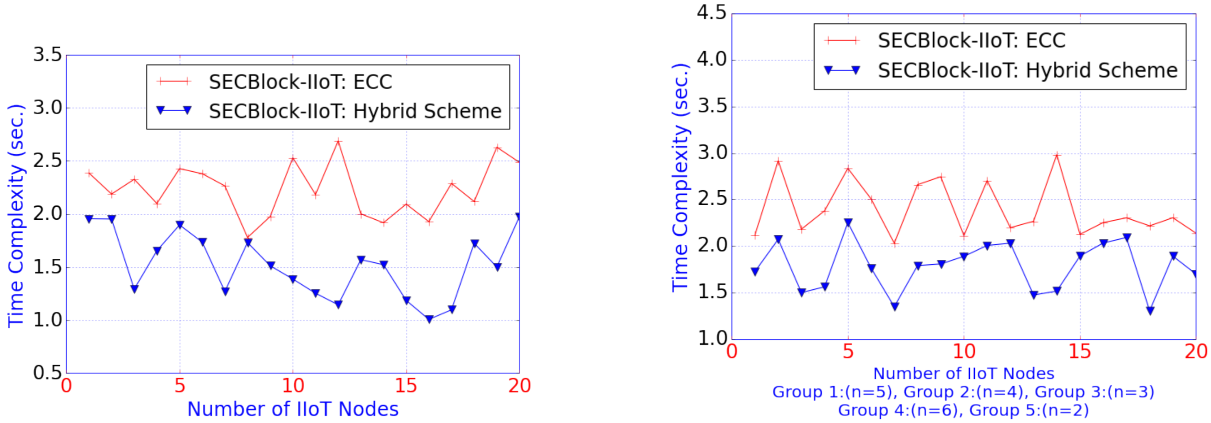
Parameter	Value
No. of IIoT, ULH, ESMH and ESML-nodes	20, 2, 1, 4
No. of groups and members (IIoT-nodes)	(5, random (1,20))
No. of Packets/tasks per IIoT-nodes	Random (1, 50), 512 bytes
Bandwidth of IIoT, ULH, and ESMH	(2, 5), (2, 10), (4, 12) Mbps
Hash Function (.)	SHA512

5 EXPERIMENTAL RESULTS

We evaluated the SECBlock-IIoT framework by testing its time complexity, energy consumption of the IIoT-nodes, average delay of the mPoV consensus algorithm, and attack detection performance of the ITD module. We used the open-source TensorFlow and Keras libraries to develop the ITD model. Ethereum and Solidity version 6.0 were used to build a consortium blockchain network. The IPFS (version 0.4.19) was used to store the blockchain data off-chain, and it was configured using AMD Phenom™ || X2 555, 3.20 GHz, 12.0 GB installed memory, and a Windows 10 (64-bit) operating system. The ITD model was evaluated by measuring its accuracy, precision, recall and F-Score, using original and transformed data. The results were compared with logistic regression (LR), support vector machine (SVM), and Gaussian naïve bayes (Gaussian NB) classifiers. The ITD model used the IoT-Botnet dataset as a model dataset. The other parameters used in the evaluation are shown in Table 1.

5.1 Analysis of the Security Scheme

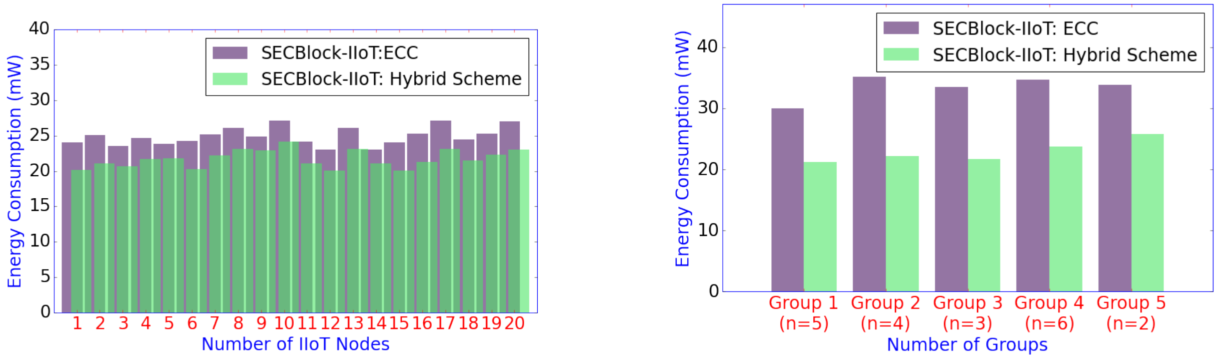
We analyzed the time-complexity of the proposed hybrid security scheme including registration, verification, encryption, and decryption for each active IIoT node and compared the results with ECC public key cryptography. The evaluated performance of the system using two different network scenarios: P2P and group communication. In the group communication scenario, five groups with a random number of IIoT nodes generated packets and communicated with their corresponding ULHs. Figure 3(a) and (b) show the average time-complexity of the system based on a varied number of IIoT nodes in the network. The results show that the proposed system generated PUF-derived secret session keys at the edge layer around 47.32% faster than the SECBlock-IIoT: ECC. On the other hand, when the nodes in a group communicated, the proposed system performed approximately 34.61% faster than the SECBlock-IIoT: ECC. These results were due to the conventional group key distribution, using P2P communication, to share the secret key,



(a)

(b)

Figure 3: Average time-complexity of each IIoT node using the proposed hybrid security scheme vs. ECC with varied IIoT nodes in (a) P2P communication and (b) group communication.



(a)

(b)

Figure 4: Average amount of energy consumption of each IIoT node using the proposed hybrid security scheme vs. ECC in (a) P2P communication and (b) group communication.

which added an additional communication overhead in contrast to the proposed scheme.

Energy consumption of IIoT nodes is important as the nodes are energy-constrained. The nodes deployed in this network often lack an external power supply, unlike the other types of nodes (i.e., ULH, ESMH). Therefore, energy-consumption of IIoT nodes due to communication was the only form of energy consumption that was considered. Figure 4(a) and (b) show the average energy consumption of the nodes in P2P and group communication scenarios, and reveal that the proposed system can significantly reduce the amount of energy dissipation of the nodes. We observed that the nodes of the proposed system used around 14.63% and 45.93% less energy than that of SECBlock: ECC in the P2P and group communication scenarios, respectively. Reduced energy consumption can enhance network stability.

5.2 Analysis of the Blockchain Module

We evaluated the performance of the proposed consortium blockchain featuring mPoV against a blockchain that utilized the PoV consensus algorithm. The blocks contained data/transactions received from IIoT nodes or data that was related to computational tasks generated by ESMHL-nodes. Blocks were generated and verified by at least 50% of elected commissioners among the FNs. To avoid network disruption or point of failure errors, P^d = 10% of FNs were self-elected as duty butlers in each round in the proposed blockchain network. The duty butlers generated blocks and added them to the blockchain. This significantly reduced broadcasting latency of the selected butler-name and rebroadcasted a replacement butler-name when a butler failed to generate a block in time when PoV was in use. The IPFS allowed storage of CID updated blocks on the blockchain which solved the network scalability problem, that is caused due to memory scarcity of edge nodes. Figure 5

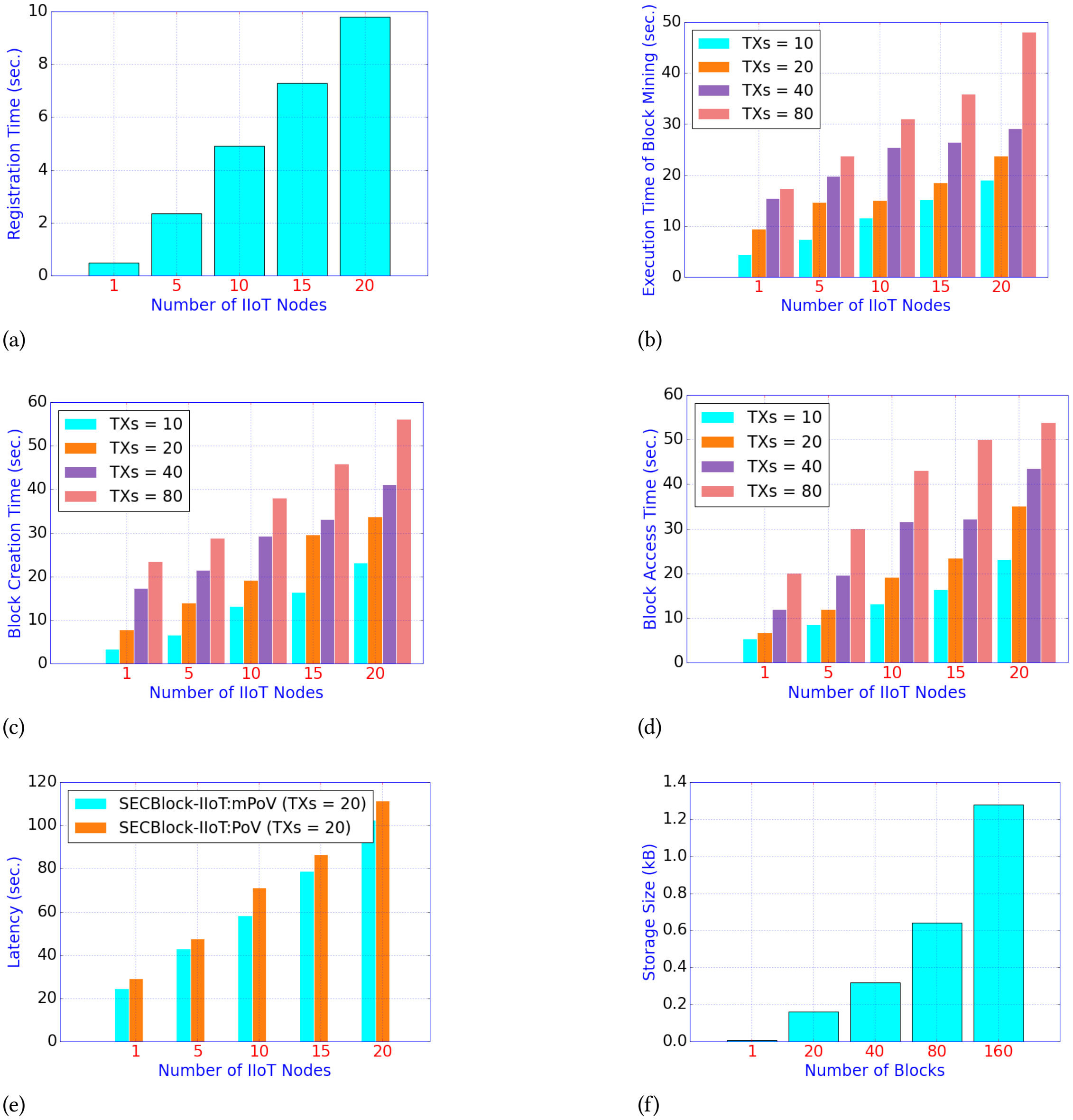


Figure 5: Performance analysis of the consortium blockchain system: (a) Registration time for various IIoT nodes, (b) Execution time of block mining with various TXs, (c) Block creation time for various TXs, (d) Block access time for various TXs, (e) Latency of the blockchain with mPoV vs. PoV consensus, including registration, block mining time, block creation time and block access time for TXs = 20, and (f) Storage size of CID of blocks (TXs = 20) using an IPFS-enabled node.

shows the results of the registration time of IIoT nodes, execution time of block mining, block creation and access time, and off-chain storage consumption of varying blocks using the IPFS. We also compared the consortium blockchain using both mPoV and PoV. The outcomes show that the proposed blockchain outperformed the PoV-applied blockchain in terms registration time, block mining time, block creation time, and block access time, which required approximately 12.78% less time than the blockchain.

5.3 Analysis of the ITD Module

There are 1498334 anomalous and 79053 normal instances in the IoT-Botnet dataset [20]. The labeling feature identified the network traffic as normal or anomalous, the categorical labeling feature classified the network traffic as Normal, DDoS, DoS, Reconnaissance, or Theft while the subcategorical labeling feature classified the network traffic as Normal, DDoS-HTTP, DDoS-TCP, DDoS-UDP,

Table 2: Selected network flow features of the IoT Botnet dataset

Feature Name			
Src IP	Flow Byts/s	Flow IAT Min	Subflow Bwd Pkts
Dst IP	Flow Pkts/s	Fwd IAT Tot	Subflow Bwd Byts
Dst Port	Flow IATMean	Fwd IAT Mean	Label
Protocol	Flow IATStd	Subflow Fwd Pkts	Cat
Flow Duration	Flow IAT Max	Subflow Fwd Byts	Sub Cat

Table 3: Autoencoder (AE) parameters

Settings	Hyperparameters
Input Layer	17 features
Encoder	Layer 1: 10 hidden nodes, activation = 'relu'
Decoder	Layer 1: 17 hidden nodes, activation = 'relu'
AE model	Loss = 'mean_squared_error, optimizer = 'sgd', epochs = 20, and batch_size = 50

Table 4: RNN parameters

Settings	Hyperparameters
Input Layer	18 features
Hidden Layers	Layer 1: 100 hidden nodes, activation 'relu', and dropout rate = 0.2
	Layer 2: 100 hidden nodes, activation 'relu', and dropout rate = 0.2
Output Layer (Binary class)	1 unit, (classified anomaly and normal), activation = 'sigmoid'
Compiler	loss = 'binary_crossentroppy', optimizer = 'Adam', with learning rate = 0.001, epochs = 20, and batch size = 50

DoS-HTTP, DoS-TCP, DoS-UDP, OS-Fingerprint, Service-Scan, Keylogging or Data-Exfiltration. The dataset was split into train (70%) and test (30%). Before training the DL model, the non-numeric and categorical features were normalized using column normalization (MinMax normalization). We used 17 network-flow features from the dataset as input features, as shown in Table 2.

Once the mPoV was implemented, the AE transformed the network-flow data into an encoded format. Encoding the data prevents the inference attacks that is used in the training phase of the model. The transformed data was used to train the RNN model. The hyper-parameters used in AE and RNN are shown in Table 3. and Table 4.

Table 5 and 6 present the subcategorical and categorical class threat detection performance of the ITD model with original and transformed data. The results show that the ITD model achieved a high level of threat detection accuracy = (99.36, 99.09)%, (99.99, 97.80)%, precision = (99.63, 99.27)%, (99.19, 98)%, recall = (99.54, 99.63)%, (99.60, 99.60)%, and F-Score = (99.54, 99.45)%, (99.39, 98.60)% for subcategorical and categorical class threat detection with original and transformed data, respectively. The results obtained from transformed data are considered acceptable when compared to the performance of the ITD model with the original data.

We compared the ITD model with higher performance classifiers such as LR, SVM and Gaussian NB, with original and transformed data, and the results are shown in Figure 6 and Figure 7. The ITD model achieved an average score of (2.20, 0.82, 8.27)% accuracy, (5.0, 2.28, 7.28)%, (5.36, 0.73, 7.73)%, (5.09, 2.73, 9.73)% precision, (9.09, 2.54, 2.45)%, (0.0, 0.45, 1.54)% recall, and F-Score (9.36, 1.91, 7.45)%, (3.0, 1.73, 7.6)% with original and transformed data for subcategorical class threat detection higher compared to the LR, SVM and Gaussian NB. The ITD model also achieved an averages of accuracy (4.99, 0.79, 12.5)%, (3.20, 3.0, 9.0)%, precision (6.19, 0.39, 13.9)%, (15.20, 7.0, 12.0)%, recall (19.2, 0.2, 0.0)%, (19.4, 18.80, 1.0)% and F-Score (18.60, 0.20, 11.0)%, (18.80, 17.40, 10.0)% with the original and transformed data for categorical class-wise threat detection higher compared to the other classifiers.

6 CONCLUSION

In this study, we developed a secure blockchain-enabled edge computing framework for IIoT networks. To ensure secure P2P and group communications, a hybrid security scheme was implemented which featured robust registration, verification, and authentication processes. PUF and Lagrange interpolation were adopted to generate and share secret keys, which reduced the communication overhead without compromising the security strength of the network. The framework included an ITD module to detect dynamic threats and was capable of using transformed data to prevent possible inference attacks. Moreover, the framework incorporated a consortium blockchain technique to prevent data alteration, as the data of IIoT systems are often confidential. IPFS was used to

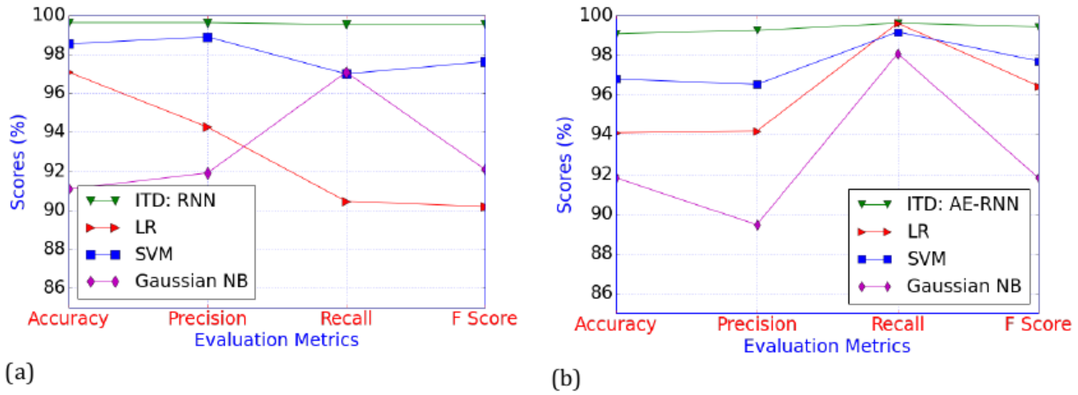


Figure 6: Comparison of the subcategory threat detection of the classifiers using IoT Botnet dataset with (a) original data and (b) transformed data.

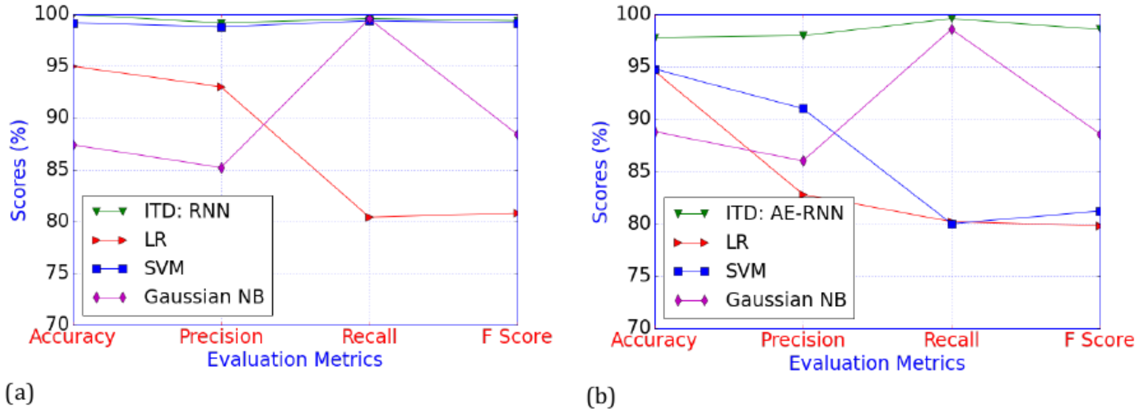


Figure 7: Comparison of the category threat detection of the classifiers using IoT Botnet dataset with (a) original data and (b) transformed data.

Table 5: Subcategory classification (%) of RNN and AE-RNN using IoT-Botnet dataset

Data	Parameters	Normal	DDoS-HTTP	DDoS-TCP	DDoS-UDP	DoS-HTTP	DoS-TCP	DoS-UDP	OS-Fingerprint	Service _Scan	Keylogging	Data-Exfiltration
Original	Accuracy	100.00	100.00	100.00	100.00	100.00	100.00	100.00	97.00	100.00	99.00	97.00
	Precision	100.00	100.00	100.00	100.00	100.00	100.00	100.00	98.00	100.00	99.00	99.00
	Recall	100.00	100.00	100.00	100.00	100.00	100.00	100.00	99.00	100.00	99.00	97.00
	F-Score	100.00	100.00	100.00	100.00	100.00	100.00	100.00	98.00	100.00	99.00	98.00
Transformed	Accuracy	100.00	100.00	100.00	100.00	100.00	100.00	100.00	96.00	100.00	99.00	95.00
	Precision	100.00	100.00	100.00	100.00	100.00	100.00	100.00	99.00	100.00	100.00	93.00
	Recall	100.00	100.00	100.00	100.00	100.00	100.00	100.00	97.00	100.00	99.00	100.00
	F-Score	100.00	100.00	100.00	100.00	100.00	100.00	100.00	98.00	100.00	100.00	96.00

store the blockchain at the edge node which alleviated storage and network scalability problems.

We evaluated and analyzed the performance of the proposed system in terms of security, energy-consumption of IIoT-nodes, latency of the hybrid security scheme, execution time and latency of block creation and access, and attack detection rate. We compared the developed SECBlock-IIoT with the baseline security scheme ECC, the blockchain consensus algorithm PoV, and ML classifiers

Table 6: Category-wise classification (%) based on RNN and AE-RNN using IoT-Botnet dataset

Data	Parameters	Normal	DDoS	DoS	Reconnaissance	Theft
Original	Accuracy	100.00	100.00	100.00	99.99	99.99
	Precision	100.00	100.00	100.00	99.99	96.00
	Recall	100.00	100.00	100.00	100.00	98.00
	F-Score	100.00	100.00	100.00	99.99	97.00
Transformed	Accuracy	100.00	100.00	100.00	98.00	91.00
	Precision	100.00	100.00	100.00	99.00	91.00
	Recall	100.00	100.00	100.00	100.00	98.00
	F-Score	100.00	100.00	100.00	99.00	94.00

LR, SVM and Gaussian NB. The results show that the proposed framework outperformed existing and conventional systems.

ACKNOWLEDGMENTS

This work was supported by a National Research Foundation of Korea (NRF) grant funded by Korean Government (MSIT) (No. 2021R1A2C2014333).

REFERENCES

[1] Tie Qiu, Jiancheng Chi, Xiaobo Zhou, Zhaolong Ning, Mohammed Atiquzzaman, and Dapeng Oliver Wu. 2020. Edge computing in industrial internet of things: architecture, advances, and challenges. IEEE Communication Survey and Tutorials 22, 4 (2020), 2462–2488. DOI: 10.1109/COMST.2020.3009103

[2] Abhishek Hazra, Mainak Adhikari, Tarachand Amgoth, and Satish Narayana Srirama. 2023. A comprehensive survey on interoperability for IIoT: taxonomy, standards, and future directions. ACM Computing Surveys 55, 1 (2023), 1–35. https://doi.org/10.1145/3485130

[3] A. S. M. Sanwar Hosen, Pradip Kumar Sharma, and Gi Hwan Cho. 2022. MSRM-IoT: a reliable resource management for cloud, fog, and mist-assistant IoT networks. IEEE Internet of Things 9, 4 (2022), 2527–2537. DOI: 10.1109/JIOT.2021.3090779

[4] Sihua Wang, Mingzhe Chen, Xuanlin Liu, Changchuan Yin, Shuguang Cui, and H. Vincent Poor. 2021. A machine learning approach for task and resource allocation. IEEE Internet of Things 8, 3 (2021), 1358–1372. DOI: 10.1109/JIOT.2020.3011286

[5] Sha Zhu, Kaoru Ota, and Mianxiong Dong. 2022. Green AI for IIoT: energy efficient intelligent edge computing for industrial internet of things. IEEE Transactions on Green Communications and Networking 6, 1 (2022), 79–88. DOI: 10.1109/TGCN.2021.3100622

[6] A. S. M. Sanwar Hosen, Pradip Kumar Sharma, In-Ho. Ra, and Gi Hwan Cho. 2022. SPTM-EC: a security and privacy preserving task management in edge computing for IIoT. IEEE Transactions on Industrial Informatics 18, 9 (2022), 6330–6339. DOI: 10.1109/TII.2021.3123260

[7] S. Velliangiri, Rajesh Mhnoharn, Sitharthan Ramachandran, Krishnasamy Venkatesan, Vani Rajasekar, P. Karthikeyan, Pradeep Kumar, Abhishek Kumar, and Shanmuga Sundar Dhanabalan. 2022. An efficient lightweight privacy-preserving mechanism for industry 4.0 based on elliptic curve cryptography. IEEE Transactions on Industrial Informatics 18, 9 (2022), 6494–6502, DOI: 10.1109/TII.2021.3139609.

[8] Dawei Li, Enzhun Zhang, Ming Lei, and Chunxiao Song. 2022. Zero trust in edge computing environment: a blockchain based practical scheme. Mathematical Biosciences and Engineering 19, 4 (2022), 4196–4216. doi: 10.3934/mbe.2022194

[9] Prabhat Kumar, Randhir Kumar, Govind P. Gupta, Rakesh Tripathi, and Gautam Srivastave. 2022. P2TIF: a blockchain and deep learning-based framework for privacy-preserving threat intelligence in industrial IoT. IEEE Transactions on Industrial Informatics 18, 9 (2022), 6358–6367. DOI: 10.1109/TII.2022.3142030

[10] Weizheng Wang, Hao Xu, Mamoun Alazab, Thippa Reddy Gadekallu, Zhaoyang Han, and Chunhua Su. 2022. Blockchain-based reliable and efficient certificateless signature for IIoT devices. IEEE Transactions on Industrial Informatics 18, 10 (2022), 7059–7067. DOI: 10.1109/TII.2021.3084753

[11] 0A. S. M. Sanwar Hosen, Saurabh Singh, Pradip Kumar Sharma, Uttam Ghosh, Jin Wang, In-Ho Ra, and Gi Hwan Cho. 2020. Blockchain-based transaction validation protocol for a secure distributed IoT network. IEEE Access 8, 117266–117277. DOI: 10.1109/ACCESS.2020.3004486

[12] Saurabh Singh, A. S. M. Sanwar Hosen, and Byungun Yoon. 2021. Blockchain security attacks, challenges, and solutions for the future distributed IoT network. IEEE Access 9 (2021), 13938–13959. DOI: 10.1109/ACCESS.2021.3051602

[13] Ruizhe Yang, F. Richard Yu, Pengbo Si, Zhaoxin Yang, and Yanhua Zhang. 2019. Integrated blockchain and edge computing systems: a survey, some research issues and challenges. IEEE Communication Surveys and Tutorials 21, 2 (2019), 1508–1532. DOI: 10.1109/COMST.2019.2894727

[14] Imtiaz Ullah and Qusay H. Mahmoud. 2020. A tow-level flow-based anomalous activity detection system for IoT networks. Electronics 9, 3 (2020), 1–18. https://doi.org/10.3390/electronics9030530

[15] Imtiaz Ullah and Qusay H. Mahmoud. 2021. IoT-Botnet Dataset. Available online: https://sites.google.com/view/iotbotnetdataset (accessed on 1 April 2021).

[16] Prabhat Kumar, Govind P. Gupta, and Rakesh Tripathi. 2021. TP2SF: a trustworthy privacy-preserving secured framework for sustainable smart cities by leveraging blockchain and machine learning. Journal of System Architecture 115 (2021), 1–15. https://doi.org/10.1016/j.sysarc.2020.101954

[17] Nour Moustafa. 2019. ToN_IoT. IEEE Dataport. doi: https://dx.doi.org/10.21227/fesz-dm97.

[18] Mohammad Mehedi Hassan, Shamsul Huda, Shaila Sharmeen, Jemal Abawajy, and Giancarlo Fortino. 2021. An adaptive trustboundary protection for IIoT networks using deep-learning feature extraction based semisupervised model. IEEE Transactions on Industrial Informatics 17, 4 (2021), 2860–2870. DOI: 10.1109/TII.2020.3015026

[19] Imtiaz Ullah and Qusay H. Mahmoud. 2021. Design and development of a deep learning-based model for anomaly detection in IoT network. IEEE Access 9 (2021), 103906–103926. DOI: 10.1109/ACCESS.2021.3094024

[20] Haipeng Yao, Pengcheng Gao, Peiying Zhang, Jingjing Wang, Chunxiao Jinag, and Lijun Lu. 2019. Hybrid intrusion detection system for edge-based IIoT network relaying on machine-learning-aided detection. IEEE Network 33, 5 (2019), 75–81. DOI: 10.1109/MNET.001.1800479

[21] Maede Zolanvari, Marcio A. Teixeira, Lav Gupta, Khaled M. Khan, and Raj Jain. 2019. Machine learning-based network vulnerability analysis of industrial internet of things. IEEE Internet of Things Journal 6, 4 (2019), 6822–6834. DOI: 10.1109/JIOT.2019.2912022

[22] Osama Alkadi, Nour Moustafa, and Benjamin Turnbull. 2020. A collaborative intrusion detection system using deep blockchain framework for securing cloud networks. In Proceedings of the SAI International System Conference. Vol. 1250. Springer-Verlog, New Work, NY, 553–565.

[23] Wei Liang, Lijun Xiao, Ke Zhang, Mingdong Tang, Dacheng He, and Kuan-Ching Li. 2022. Data fusion approach for collaborative anomaly intrusion detection in blockchain-based system. IEEE Internet of Things Journal 9, 16 (2022), 14741-14751. DOI: 10.1109/JIOT.2021.3053842

[24] Marwa Keshk, Benjamin Turnbull, Nour Moustafa, Dinusha Vatsalan, and Kim-Kwang Raymond Choo. 2020. A privacy-preserving framework-based blockchain and deep learning for smart power networks. IEEE Transactions on Industrial Informatics 16, 8 (2020), 5110–5118. DOI: 10.1109/TII.2019.2957140

[25] Osama Alkadi, Nour Moustafa, Benjamin Turnbull, and Kim-Kwang Raymond Choo. 2021. A deep blockchain framework-enabled collaborative intrusion detection for protecting IoT and cloud network. IEEE Internet of Things Journal 8, 12 (2021), 9463–9472. DOI: 10.1109/JIOT.2020.2996590

[26] Maninderpal Singh, Gagangeet Singh Aujla, and Rasmeet Singh Bali. 2021. A deep learning based blockchain mechanism for secure internet of drones environment. IEEE Transactions on Network Science and Engineering 22, 17 (2021), 4404–4413. DOI: 10.1109/TITS.2020.2997469

[27] Zhetao Li, Jiawen Kang, Rong Yu, Dongdong Ye, Qingyong Deng, and Yan Zhang. 2018. Consortium blockchain for secure energy trading in industrial internet of things. IEEE Transactions on Industrial Informatics 14, 8 (2018), 3690–3700. DOI: 10.1109/TII.2017.2786307

[28] Kejiao Li, Hui Li, Han Wang, Huiyao An, Ping Lu, Peng Yi, and Fusheng Zhu. 2020. PoV: an efficient voting-based consensus algorithm for consortium blockchains. Frontiers in Blockchain 3, 11 (2020), 1–16. https://doi.org/10.3389/fbloc.2020.00011

[29] Alireza Shamsoshoara, Ashwija Korenda, Fatemeh Afghah, and Sherali Zeadally. 2020. A survey on physical unclonable function (PUF)-based security solutions

for internet of thing. Computer Networks 183 (2020), 1–21. https://doi.org/10.1016/j.comnet.2020.107593

[30] Mahabub Hasan Mahalat, Suraj Mandal, Anindan Mondal, and Bibhash Sen. 2019. An efficient implementation of arbiter PUF on FPGA for IoT applications. In Proceedings of the 32nd IEEE International System-on-Chip Conference (SOCC). IEEE, Singapore, 334–329. DOI: 10.1109/SOCC46988.2019.1570548268