information

MDPI

Article

BCoT Sentry: A Blockchain-Based Identity Authentication Framework for IoT Devices

Liangqin Gong [1,2,3], Daniyal M. Alghazzawi [4] and Li Cheng [1,2,3,*]

[1] The Xinjiang Technical Institute of Physics & Chemistry, Chinese Academy of Sciences, Urumqi 830011, China; gongliangqin18@mails.ucas.edu.cn
[2] University of Chinese Academy of Sciences, Beijing 100049, China
[3] Xinjiang Laboratory of Minority Speech and Language Information Processing, Urumqi 830011, China
[4] Information Systems Department, Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah 21589, Saudi Arabia; dghazzawi@kau.edu.sa
* Correspondence: chengli@ms.xjb.ac.cn

Abstract: In Internet of Things (IoT) environments, privacy and security are among some of the significant challenges. Recently, several studies have attempted to apply blockchain technology to increase IoT network security. However, the lightweight feature of IoT devices commonly fails to meet computational intensive requirements for blockchain-based security models. In this work, we propose a mechanism to address this issue. We design an IoT blockchain architecture to store device identity information in a distributed ledger. We propose a Blockchain of Things (BCoT) Gateway to facilitate the recording of authentication transactions in a blockchain network without modifying existing device hardware or applications. Furthermore, we introduce a new device recognition model that is suitable for blockchain-based identity authentication, where we employ a novel feature selection method for device traffic flow. Finally, we develop the BCoT Sentry framework as a reference implementation of our proposed method. Experiment results verify the feasibility of our proposed framework.

Keywords: IoT; blockchain; authentication

Citation: Gong, L.; Alghazzawi, D.M.; Cheng, L. BCoT Sentry: A Blockchain-Based Identity Authentication Framework for IoT Devices. *Information* **2021**, *12*, 203. https://doi.org/10.3390/info12050203

Received: 10 April 2021
Accepted: 7 May 2021
Published: 10 May 2021

1. Introduction

Commonly, an IoT device equipped with tags or sensors is attached to "a thing" and collects, stores, and transmits information via an IoT network. The management of the network is typically achieved through a centralized architecture [1,2]. In recent years, the total number of IoT devices has grown exponentially. It was expected that the number of connected devices in use in 2019 was 14.2 billion, and this number is expected to increase to 25 billion by 2025 [3,4].

Meanwhile, cyberattacks against IoT devices and networks have become more frequent. The consequences could be devastating and lead to major threats to society [5]. For instance, the Mirai virus is a typical example of malicious attacks against device authentication. It targets the security vulnerability of IoT devices, turns them into remote-controlled "zombie" devices, and uses them for DDoS attacks. A well-known incident happened in 2016 when Mirai attacked the US DNS service provider Dyn, which nearly took down half of the Internet service in the United States [6].

Existing efficient security solutions are often centralized infrastructure (such as PKI), which relies on trusting third-party service providers. However, this mechanism suffers from single point of failure (SPOF), many-to-one traffic, and reduced scalability. Unlike full functional computing nodes, IoT devices generally have limited security measures for authentication. It is necessary to propose a new authentication system for IoT that has the following characteristics: (1) allows an easy integration of new IoT devices; (2) fully adapted to IoT requirements and needs; and (3) does not depend on the type of device, nor on the use case architecture and design [7].

IoT devices are distributed via connections between different types of physical networks. Devices communicate with IoT applications or other devices through various network protocols, such as ZigBee, Z-Wave, and MQTT. By their nature, IoT devices exist in a heterogeneous distributed network environment, and a huge number of devices are capable of peer-to-peer communication. These features can be directly linked to blockchain architecture, which is also based on a decentralized infrastructure and uses a distributed computational paradigm. It involves three key concepts [8]: (1) encrypted chain-like blocks for data storage; (2) distributed node and consensus algorithms for data generation and updates; (3) smart contracts for data manipulation and operation.

The concept of BCoT is therefore proposed to merge IoT with blockchain [9]. However, IoT device security is still an open research field in BCoT research and practices, especially device identity authentication, which remains an active research direction in both academia and industry [10].

Most of the IoT devices are enabled with IP-connected network functionality yet limited resources for computational intensive security models [11]. Specifically, the following questions need to be addressed: (a) How to deploy blockchain in IoT scenarios, i.e., how to manage IoT data through blockchain? (b) How to store device identity information in a blockchain network where participant nodes have limited computational power? (c) How to utilize the smart contract mechanism to enhance device identity authentication?

Goals and Contributions. This paper responds to the above questions by proposing BCoT Sentry—a framework that integrates blockchain with an IoT network and enhances network security by analyzing device traffic flow patterns obtained from data storage in blockchain.

The main contributions of this study are listed as follows:

1. We design an IoT blockchain architecture to store device identity information in a distributed ledger.
2. We propose a BCoT Gateway to facilitate the recording of authentication transactions in a blockchain network without modifying existing device hardware or applications.
3. We propose a new device recognition model that is suitable for blockchain-based identity authentication, where a novel device traffic flow feature selection method is proposed.
4. We develop a BCoT Sentry framework as a reference implementation of our proposed method.

This paper is organized as follows: First, in Section 2, we describe the motivation and related works, and then in Section 3, we lay out the framework design and propose our device recognition model. In Section 4, we introduce the reference implementation of our model and framework. In Section 5, we explain the experiments and evaluation metrics. Finally, we summarize our conclusion and the potential future directions.

2. Motivation and Related Work

2.1. IoT Network Security

IoT integrates sensors, transmitters, and controllers through various communication networks. Powered by advanced data analysis and other technologies, IoT greatly improves manufacturing efficiency and product quality, and meanwhile, reduces product costs and resource consumption.

In a typical industrial IoT scenario, a gateway device is commonly applied to isolate terminal sensors and controllers from the upper-layer network. Data collected by sensors are transmitted to centralized IoT applications that may remotely control executable units in order to achieve certain business logic requirements. However, this type of setting has known vulnerability. For instance, Stuxnet damaged the property of a number of parties outside Iran, which sustained only 60% of the Stuxnet infections [12]. In the local industrial infrastructure, the programmable logic controllers (PLCs) from Siemens were attacked.

Moreover, industrial robots exposed directly to the Internet could also be attacked via FTP services or industrial routers [13,14]. Among the total 83,673 robots surveyed in

their studies, 5105 devices do not have an authentication mechanism at all; 59 devices have known embedded vulnerabilities, and 6 devices identified with new security holes.

Another widely adopted IoT scenario is an intelligent warehouse management system (WMS). It involves electronic labels, RFID scanners, and various warehouse supporting facilities. Different types of environmental sensors and safeguard devices need to be properly identified and inter-communicated in a stable and robust network environment. If the WMS is equipped with less-secure sensors or robots, attackers can tamper with raw sensor data and execute malicious operations through the robots, which might cause significant loss.

Gope et al. [15] propose a computationally efficient lightweight and privacy-preserving mutual user authentication scheme. In the proposed scheme, physical security of devices as well as the sensor nodes deployed in the open hostile environment are protected. These devices and sensor nodes are not required to store any sensitive information, such as secret credentials on the sensing devices. However, this research uses a centralized architecture, which has limited scalability and is vulnerable to SPOF.

The concept of 'Smart City' is referred to as the safe, secure, environmental, and efficient urban center of the future with advanced infrastructures, such as sensors, electronic devices, and networks, to stimulate sustainable economic growth and a high quality of life [16].

For example, transportation is the artery of a city and an important part of smart city construction. Intelligent traffic management applies IoT technologies, such as wireless communication, cloud computing, perception technology, video vehicle surveillance, and GPS. Intelligent transportation employs various IoT devices, such as microcontrollers for connected cars, RFID devices, microchips, video camera equipment, GPS receivers, and navigation systems. By analyzing the real-time traffic information of people, cars, and traffic in the entire area from various perceptions, the platform controls traffic through traffic signals, ramp flow control, and dynamic traffic information signs.

Mohit et al. [17] propose an authentication protocol based on a user ID and password for a vehicular system in WSN to tackle the problem of vehicles running on the road, such as avoidance of traffic jams and other related problems. All of the vehicle sensors are registered through a registration authority. However, there is no additional measure taken to verify the identity of the device.

Despite the advantages IoT offers in a smart city, new security threats are also introduced, especially in transportation, where cyberattacks (such as device hijacking) could lead to devastating consequences.

The issue we are trying to address here is to enhance the device authentication without introducing extra computational burden on the end devices, yet take advantage of distributed reliability from blockchain.

2.2. Related Work

2.2.1. Blockchain and Smart Contract

Blockchain is a distributed shared ledger. In 2008, Satoshi Nakamoto proposed tBitcoin [18], explaining the architectural concept of an electronic cash system based on P2P network, encryption, time stamp, and Merkel tree, etc. As the underlying technology of digital cryptocurrencies such as Bitcoin, blockchain technology was originally designed to solve the long-term double payment problem [19] and the Byzantine generals problem [20].

In 2015, Ethereum [21] and Hyperledger [22] were proposed as a representative of a new generation of blockchain. They provide a decentralized computing platform, which allows a smart contract to be deployed as a manager so that the transaction can be executed with the contractual terms of an agreement [23]. A smart contract can encode any set of rules represented in its programming language. For instance, a contract can execute transfers when certain events happen (e.g., payment of security deposits in an escrow system). Accordingly, smart contracts can be applied to a wide range of applications, including financial instruments (e.g., sub-currencies, financial derivatives, savings wallets, wills)

and self-enforcing or autonomous governance applications (e.g., outsourced computation, decentralized gambling) [24].

Since 2017, recent research, for instance, the cross-chain technology [25], sharding [26], and redesigned blockchain structure (e.g., directed acyclic graph (DAG)) [27], has improved the throughput, reduced the delay of transaction confirmation, and expanded the application scenarios of blockchain. These technologies allow the blockchain to be widely used in various fields, indicating a new era of blockchain.

2.2.2. Security Challenges in IoT

Generally, IoT security should address issues such as data authentication, access control, and user privacy. Meanwhile, the lightweight feature and limited computing power of IoT devices should be well considered when designing security models [28,29].

Several representative related studies are listed as follows:

- Mnif et al. [30] propose a new method adapted to resource-constrained wireless sensor networks, where only legitimate users can access node resources, and unauthorized users are denied access.
- Markus et al. [31] propose a system capable of automatically identifying the types of devices being connected to an IoT network and enabling enforcement of rules for constraining the communications of vulnerable devices to minimize damage resulting from their compromise.
- There are some research and development works in the fields of wireless sensor networks and RFID [32,33].

Exploration and implementation of security technologies in IoT is still an open challenge, and the issue of the security architecture of IoT still has room for improvement [34].

In the PKI framework, the single CA model is a commonly used model in an enterprise environment, and a CA is used to issue and manage certificates for all end users in the network. We list the advantages and drawbacks of blockchain and single CA model in Table 1 to show the improvements brought by the blockchain [35,36]:

Table 1. Comparison of blockchain-based model and PKI.

Comparison Item	Single CA Model	Blockchain-Based Model
How to Build Trust?	Based on users subjective trust	Based on mathematics
Trust Anchor	Public key of the CA	Cryptography method and Consensus mechanism
Vulnerable to SPOF	Yes	Naturally immune
Vulnerable to Replay Attack?	Additional applications need to be deployed	Each of transactions is verified by timestamp, nonce, transaction ID, etc.

In the existing PKI method, the CA periodically updates and releases Certificate Revocation Lists (CRL). One drawback of this method is that the time granularity of revocation is limited to the CRL release period. During this period, the revoked certificate is still trusted, and malicious attackers can illegally obtain data through revoking delay attacks. In addition, the existing revocation certificate inspection scheme is centralized, which will cause security bottlenecks.

If blockchain is used to manage the operation of certificates, the security bottleneck caused by the existing centralized solution can be effectively eliminated. In addition, the smart contract can make the operation and revocation verification of certificates effective and rapid response.

2.2.3. Convergence of Blockchain and IoT

Blockchain has the following characteristics that meet the needs of IoT [37]:

(1) Decentralization. Distributed nodes maintain data consistency on the blockchain network through a consensus algorithm without third parties.
(2) Persistency. In blockchain, invalid transactions will not be identified by miners, so transactions that have been confirmed cannot be deleted.
(3) Auditability. Each transaction can be easily verified and tracked for every packaged transaction on the blockchain and can point to the transaction packaged in the previous block.

The main goal of the convergence includes: (1) to introduce trust and secure data exchange between IoT devices (systems) by taking advantages of blockchain; (2) to record, identify, and verify IoT transactions using cryptographic mechanisms provided by blockchain technology while balancing the network overhead and device computing capability; (3) to enable the secure P2P interactions between IoT devices without centralized third-party intervention by using blockchain nodes and smart contracts.

In BCoT, IoT data are synchronized to all nodes after reaching a consensus. A consensus mechanism is used to ensure the consistency of the system in Blockchain. There are several common consensus algorithms, such as Proof of Work (PoW) [38] states that generating a piece of data must satisfy certain requirements, which is difficult to produce but easy to verify. Proof of Stake (PoS) [39] states that miners can mine or validate block transactions based on the amount of cryptocurrency coins the miner holds. Practical Byzantine Fault Tolerance (PBFT) [40] is a method to solve the Byzantine Generals Problem that can be used in a real production environment.

In order to optimize the resource consumption of the blockchain and make it suitable for IoT devices, Karlsson et al. [27] propose a permissioned, DAG structured blockchain suitable for power-constrained environments with limited network connections. Liu et al. [41] propose LightChain, which has the characteristic of resource-efficient without affecting the traceability and nonrepudiation of blockchain, and propose a novel consensus mechanism to reduce the consumption of computing power. Prescilla et al. [42] propose a sliding window mechanism that stores only a limited part of the blockchain and maintains the whole blockchain in the private cloud to make the blockchain suitable for IoT devices. Ellul et al. [43] describe a split virtual machine that allows devices to interact with the blockchain system. These studies target blockchain structure optimization in order to incorporate IoT devices as direct blockchain nodes. However, device identity authentication is not fully covered in this research.

Gochhayat et al. [44] design a multi-user model composed of cloud storage servers and group users. Users encrypt files and store them in the district. On the blockchain, the cloud storage of files is done after the data are on the chain. Yakubov et al. [35] and Louise et al. [45] propose a feasible PKI identity authentication scheme in the blockchain. Cruz et al. [46] used blockchain to solve the cross-organizational access control problem in role base access control (RSAC) and realized the cross-organizational authentication of user roles. Bouras et al. [47] propose IoT-CCAC, a decentralized capability-based access control architecture designed for IoT consortium networks where a blockchain-based database is utilized. Cui et al. [48] propose a data management model based on the blockchain platform, where multiple IoT devices are controlled by a management center and the management center obtains access rights through a third party. Bouras et al. [49] propose a lightweight architecture and the associated protocols for consortium blockchain-based identity management to address privacy, security, and scalability issues in a centralized system for IoT. These studies improve the existing methods from the perspectives of cloud, PKI system, and access control. However, the work of identity authentication for IoT devices has room for improvement.

In order to solve the aforementioned identity authentication problem of IoT devices: Omar et al. [50] use function-based tokens based on the ERC721 standard to provide secure identity verification and authorization for IoT devices. Ujjwal et al. [51] propose a verification mechanism based on physical unclonable functions (PUFs), which generates a unique device ID for IoT devices. The registered manufacturer uploads each device ID

to a blockchain network. When registering a new device, the end user verifies whether the hash value exists in the blockchain. Alblooshi et al. [52] proposed a traceable medical IoT device management solution to solve the problem of counterfeit devices through two smart contracts.

In the above-surveyed literatures, the authors propose new methodologies and methods for the integration of IoT and blockchain. A few studies focus on identity authentication through global registration on the public chain. These approaches lay out a theoretically feasible solution; however, it is challenging for IoT manufacturers to adopt the idea due to foreseeable cost trade-off. In this research, we intend to explore a practically feasible consortium blockchain solution for IoT device authentication.

3. The BCoT Sentry Methodology

Due to the cost-performance factor, the limited resources of most IoT devices could hardly support complex security models or algorithms. Practically, a security mechanism is implemented in different IoT applications in order to realize various business logic requirements. The cost of modifying existing applications could be extremely high; therefore, our end goal is to propose a new mechanism that could enhance security through a more complex blockchain-based security model without introducing a practically unfeasible cost increase due to the modification of end-device hardware design or the reconstruction of IoT applications.

We propose BCoT Sentry, a system that integrates blockchain with an IoT network and enhances network security by analyzing device traffic flow patterns. In BCoT Sentry, BCoT Gateways are blockchain nodes where an IoT device security module is employed through a smart contract.

Kanhere et al. [53] propose a lightweight blockchain-based architecture for IoT that virtually eliminates the overheads of classic blockchain while maintaining most of its security and privacy benefits. The constituent nodes in a P2P network are grouped in clusters, each cluster selects a Cluster Head (CH), and then CHs maintain a public blockchain. They verify the effectiveness of the proposed architecture against DOS, modification attack, dropping attack, and appending attack. Finally, they evaluate the traffic overhead and processing overhead of the architecture.

Ours work stores device fingerprints in the consortium blockchain through a specially designed BCoT Gateway, which facilitates the recording of authentication transactions in a blockchain network.

3.1. BCoT Sentry Architecture

The BCoT Sentry architecture is depicted in Figure 1, which includes the following components.

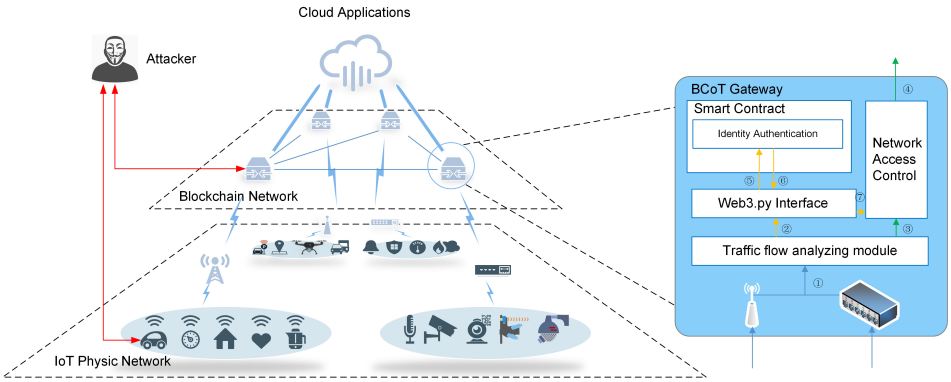

Figure 1. BCoT Sentry system design.

(1) IoT Physic Network: An IoT physic network is a communication network composed of numerous tiny devices with limited capabilities. The IoT physic network can

operate in an independent environment, or it can be connected to the Internet through a gateway.

In our proposed framework, IoT devices join the blockchain network through special gateways, and therefore, existing hardware and software applications can be easily integrated without additional cost.

(2) Blockchain Network: In our framework, the blockchain network is a consortium chain. Nodes communicate with the blockchain through a reserved interface. Transaction logs and device records are maintained on the blockchain by each node and are decentralized and cannot be tampered with.
(3) Cloud Applications: In a smart city scenario, IoT devices are typically utilized by cloud-based applications, such as smart transportation, smart home, and telemedicine. Our framework should also support the blockchain-based device authentication across the lower layer and upper layer of cloud applications.
(4) BCoT Gateway: In our framework, the BCoT Gateway is essentially an IoT gateway [54] with blockchain node capability. BCoT Gateway can provide the functionalities of protocol conversion and device management:

The BCoT Gateway manages the sensor node connected to acquire the node's identification, status and properties, and realizes remote startup, shutdown, control, and analysis.

The BCoT Gateway supports protocol interworking between the traditional network and IoT physic network, which includes Zigbee, Z-Wave, and MQTT.

(5) Traffic Flow Analyzing: This module monitors the behavior of an individual IoT device and sends a device traffic flow feature to the Smart Contract via blockchain transaction.
(6) Smart Contract and Interface: The device identity authentication mechanism described in this paper is realized by a single smart contract. The IoT device's identity information and related operations are defined in smart contracts and triggered by blockchain transactions. The smart contract enforces the access permission policies through defined operations and ensures that only authorized entities could modify or access the device identity information.

Once the smart contract is deployed, it will generate a unique contract address. We specify the contract address and Application Binary Interface (ABI) of the deployed contract in the web3.py interface, so the traffic flow analyzing module can trigger smart contract through blockchain transactions to verify device identity.

3.2. Decentralized Identity Authentication Mechanism

The procedure of the decentralized identity authentication mechanism has three phase:

In the initialization phases, (a) BCoT Gateways join the blockchain network so that each of them will keep a copy of the blockchain. (b) Smart contracts are deployed on the blockchain, and each BCoT Gateway records its contract address and ABI. (c) A blockchain externally owned account (EOA) is created and bounded to each BCoT Gateway.

In the device registration phase, the management entity of the system extracts the traffic flow features of IoT devices and trains the model, then triggers smart contracts through blockchain transactions, and uploads device identity information and weight information to the smart contract. The device identity information will be synchronized to all blockchain nodes when a consensus is reached.

In the device authentication phase, when a device is connected to the network, BCoT Gateway extracts the traffic flow features of the device through a traffic flow analyzing module, then calls the smart contract to identify the types or to detect whether the identity of the device is fraudulently through the web3.py interface.

3.3. Device Authentication Model

In our device authentication model, we define a device fingerprint to discriminate types of IoT devices.

The fingerprint represents the unique network traffic pattern of the device. When an IoT device connects to the gateway, the device traffic will follow a specific process established by the device manufacturer. This process usually consists of a distinguishable communication sequence initiated by an IoT device, and our fingerprint attempts to capture this characteristic sequence.

The IoT Devices reduce the rate of sending data packets, which can be used to determine whether the initialization phase is complete.

In the proposed device authentication mode, let D be an IoT device, let Ω be the universal set of devices, let $\mathcal{C} = \{C_1, C_2, \ldots, C_k\}$ be all types the of devices, let $P^D = \{p_1, p_2, \ldots, p_n\}$ be the data packets during the initialization phase, let $\overrightarrow{FP^D}$ be the fingerprint of device D, let $\overrightarrow{FP_C}$ be the fingerprint of types of device C.

Our device authentication model can be divided into two parts:

Register: Register and identify the types of new devices that are discovered in the network. For an unknown device D_1 with fingerprint FP_1, determine the type of the device C_1, which is defined by:

$$J_1(D_1, FP_1) = C_1$$

Fraud Detection: Fraud detection verifies and confirms the identity of registered IoT devices. For an IoT device D_2 with fingerprint FP_2 that claims to be type C_2, determine whether the identity of the device is correct. This model is defined by:

$$J_2(D_2, FP_2, C_2) = \begin{cases} 1, & \text{if device type matched} \\ 0, & \text{else} \end{cases}$$

3.3.1. Device Fingerprint

Features that are used to build a fingerprint are shown in Table 2.

The feature vector constituted by a packet p_i can be expressed as:

$$f_i = \{f_{i,1}, f_{i,2}, f_{i,3}, \cdots, f_{i,16}\}, \quad i \in \{1, \cdots, n\}$$

Hence, the behavior of the device during the initialization phase can be described by a $n * 16$ feature matrix:

$$F = \begin{bmatrix} f_{1,1} & f_{1,2} & \cdots & f_{1,16} \\ f_{2,1} & f_{2,2} & \cdots & f_{2,16} \\ \vdots & \vdots & & \vdots \\ f_{n,1} & f_{n,2} & \cdots & f_{n,16} \end{bmatrix}$$

Consider that the number of packets sent in the initialization phase of the device, n, is also an important feature, so $\overrightarrow{FP^D}$ is given by:

$$\overrightarrow{FP^D} = \left\{ \sum_j f_{j,1}, \sum_j f_{j,2}, \sum_j f_{j,3}, \cdots, \sum_j f_{j,16}, n \right\}, \quad j \in \{1, \cdots, n\}$$

Hence, $\overrightarrow{FP_C}$ is given by:

$$\overrightarrow{FP_C} = mean(\overrightarrow{FP^D}), \quad D \in \Omega_{C_i}$$

Table 2. Description of the packet features.

Type	Features	Representation
Link layer protocol (2)	ARP/LLC	packet number
Network layer protocol (3)	IP/ICMP/EAPoL	packet number
Transport layer protocol (2)	TCP/UDP	packet number
Application layer protocol (9)	HTTP/HTTPS/DHCP /BOOTP/SSDP/DNS /MDNS/NTP/TELNET	packet number
–	Packet length	number of packets in a pcap file

3.3.2. Weight Assignment

The importance of each feature in device fingerprints should be evaluated from three perspectives (as shown in Table 3):

Table 3. The components of weight.

Components	Description
Discrimination	The association between a feature and corresponding category
Stability	The stability of a feature in the same category
Sensitivity	The sensitivity of the feature to change

(1) Discrimination. Discrimination here refers to the degree of association between a feature and corresponding category.

The maximum information coefficient (MIC), proposed by David [55], is used to measure the discrimination of IoT devices and is widely used for feature selection in machine learning. In our application scenario, devices that have the same type should generate traffic flow with the same features in the same phase. The number of connected IoT devices will keep growing over time, so it conforms to the characteristics of the MIC "big data set". The MIC is obtained by the following equation:

$$I[x;y] \approx I[X;Y] = \sum_{X,Y} p(X,Y) \log_2 \frac{p(X,Y)}{p(X)p(Y)}$$

$$MIC[x;y] = \max_{|X||Y|<B} \frac{I[X;Y]}{\log_2(\min(|X|,|Y|))}$$

where X is the column vector composed of the values of attribute x in all samples, and Y the column vector composed of labels corresponding to each sample. B is the auxiliary variable that is usually set to the 0.6 power of the amount of data sets.

Let $\overrightarrow{dics}^C$ be the *discrimination* vector and given by:

$$\overrightarrow{dics} = \{MIC[x_1;y], MIC[x_2;y], \ldots, MIC[x_{17};y]\}$$

(2) Stability. Stability refers to the change of a feature in the same category. A device may be classified into the wrong category due to poor stability of its feature field. Therefore, the stability of each feature needs to be considered.

We use the coefficient of variation (CV), a dimensionless quantity, to measure the stability of a feature.

CV is only defined when the average is not 0, but there are several features of which the average is 0. In the IoT scenario, the standard deviation will be 0 if the average of a feature is 0. So a supplementary definition is made to make CV meaningful when the average is 0. For a feature i with average μ and standard deviation σ, its CV_i is:

$$CV_i = \begin{cases} \frac{\sigma}{\mu} & \mu \neq 0 \\ 0 & \mu = 0 \end{cases}$$

The stability of the feature i in device type C can be expressed as:

$$stab_i = \begin{cases} 1 - CV_i & CV_i < 1 \\ 0 & CV_i \geq 1 \end{cases}$$

Let $\overrightarrow{stab^C}$ be the stability vector for device type C and given by:

$$\overrightarrow{stab^C} = \{stab_1, stab_2, stab_3 \ldots, stab_{17}\}$$

Hence, the stability of all types of device $\overrightarrow{stab}$ is given by:

$$\overrightarrow{stab} = mean(\overrightarrow{stab^C}), \quad C \in \Omega$$

(3) Sensitivity. Sensitivity is defined as a measure of how sensitive the feature is to change. Features with a lower frequency should be sensitive to changes; on the contrary, higher frequency features are relatively insensitive to changes.

For example, when a device is infected by the Mirai virus, numerous Telnet requests will appear on the network. In our scenario, protocols like TELNET should not or rarely appear, so that the infected device may be identified through the TELNET protocol [56].

The proportion of the occurrence times of each protocol in P is given by the following equation:

$$\overrightarrow{\mathrm{F}_{occ}} = \left\{ \frac{\sum_j f_{j,1}}{n}, \frac{\sum_j f_{j,2}}{n}, \frac{\sum_j f_{j,3}}{n}, \cdots, \frac{\sum_j f_{j,17}}{n}, 1 \right\}, \quad j \in \{1, \cdots, 17\}$$

Let $\overrightarrow{sen^C}$ be the sensitivity vector of types of device C and given by:

$$\overrightarrow{sen^C} = \left\{ \frac{1}{1 + \frac{\sum_j f_{j,1}}{n}}, \frac{1}{1 + \frac{\sum_j f_{j,2}}{n}}, \frac{1}{1 + \frac{\sum_j f_{j,3}}{n}}, \cdots, \frac{1}{1 + \frac{\sum_j f_{j,n}}{n}}, \frac{1}{2} \right\}, \quad j \in \{1, \cdots, 17\}$$

(4) Weight of Fingerprints. In summary, the weight $\overrightarrow{weight^C}$ corresponding to a type of device C is given by:

$$\overrightarrow{\boldsymbol{W}^C} = \alpha * \overrightarrow{dics} + \beta * \overrightarrow{stab} + \gamma * \overrightarrow{sen^C} \quad (\alpha + \beta + \gamma = 1)$$

Here, the values of α, β, γ can be freely specified; in this paper, we set $\alpha = 0.4$, $\beta = 0.3$, $\gamma = 0.3$.

3.3.3. Arbitration

(1) Register: To identify the type of a new device that is discovered in the network, the weighted distance between the devices is needed, and devices of the same type will have a minimum weighted distance. For a newly connected device D_x and a certain type of device $C \in \Omega$, the distance vector will be:

$$\overrightarrow{\boldsymbol{Dis}} = |FP^{D_x} - FP^C|$$

The device type of D_x should be C that minimizes the d in the universal set Ω, the weighted distance of device D and type of device C is:

$$d(D_x, C) = \overrightarrow{\boldsymbol{Dis}} \cdot \overrightarrow{\boldsymbol{W}^C}$$

(2) Fraud Detection: To verify and confirm the identity of registered IoT devices. Let *ind* be the fraud indicator, which is used to determine whether the identity of a registered device has been fraudulently used.

The standard deviation of device type C is $\overrightarrow{std(C)} = \{\sigma_1, ...\sigma_{17}\}$, so that *ind* can be defined by:

$$ind(C) = \overrightarrow{std(C)} \cdot \overrightarrow{W^C}$$

Therefore, whether device D_x belongs to category C can be derived from:

$$F_2(D_x, FP^{D_x}, C) = \begin{cases} 1, & distance(D_x, C) < ind(C) \\ 0, & \text{else} \end{cases}$$

4. Implementation

We develop a prototype of BCoT Sentry for testing and evaluation. The deployment of the system is shown in Figure 2. In this paper, we use an Ubuntu virtual machine to simulate the function of the BCoT Gateway that provides a Python3 environment.

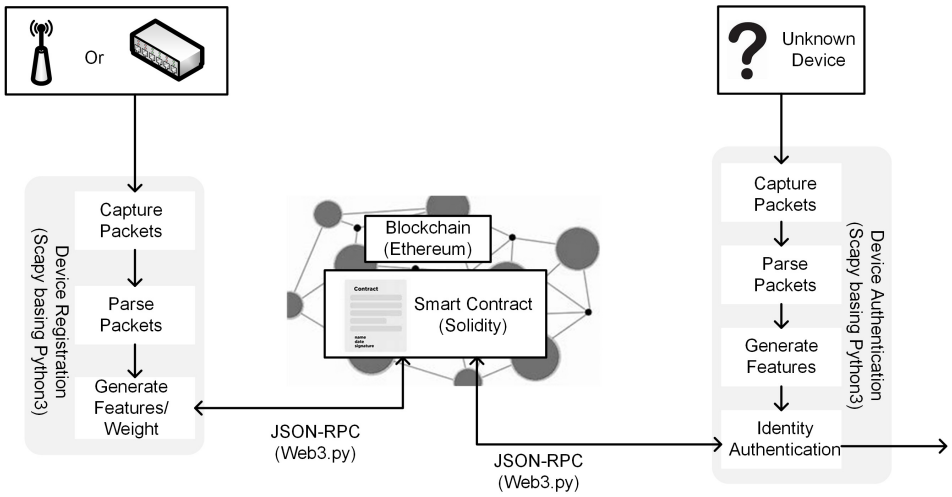

Figure 2. System implementation.

4.1. Device Registration

Scapy [57] is a Python program and library that enables the user to send, sniff, and dissect and forge network packets. This capability allows the construction of tools that can probe, scan, or attack networks.

IoT devices will follow the procedure established by the manufacturer and register themselves to the network. The characteristic network traffic flow will be generated. We use the Scapy tool to collect and analyze traffic flow to get the feature vector of IoT devices and the corresponding weight vector.

4.2. Smart Contract Interface

Web3.py [58] is a Python library for interacting with Ethereum. It is commonly found in decentralized apps (dapps) to help with sending transactions, interacting with smart contracts, reading block data, and a variety of other use cases. The original API was derived from the Web3.js Javascript API but has since evolved toward the needs and creature comforts of Python developers.

The feature vector and weight vector will be uploaded to the blockchain in the form of transactions through the JSON-RPC interface, which is achieved through web3.py in the python3 environment.

4.3. Blockchain Network

The Ethereum Virtual Machine (EVM) used in this paper is Geth with the Golang programming language.

We develop a proof of concept (PoC) implementation of the BCoT Sentry in an Ethereum private chain under a generic genesis block in order to test and evaluate it. In the private blockchain, five BCoT Gateways participate in competitive mining as a full-featured blockchain node. We set the time to generate a new block to about 5 s by adjusting the difficulty of mining. The communication with the blockchain is supported by the API provided based on the HTTP-RPC interface.

4.4. Smart Contract

Solidity [59] is a statically-typed curly-braces programming language designed for developing smart contracts that run on the EVM.

The smart contract in our framework is implemented using Solidity. The device identity information and authentication operations are shown in Figure 3. We assign access rights to the functions in the contract to protect the device's identity authentication information.

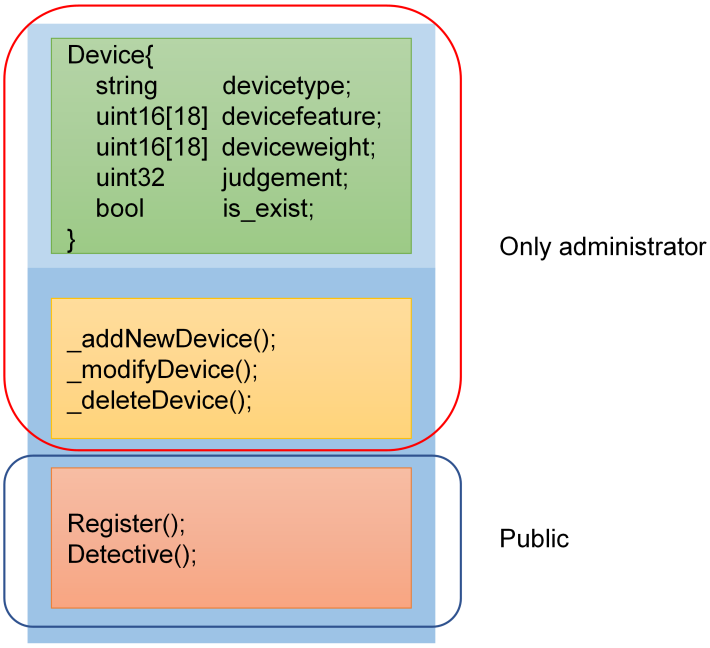

Figure 3. Some details of the smart contract.

Since Solidity does not support floating-point data types, we need to find alternative representation. We build an IoT device authentication model and also modified the device features and weights by reserving a fixed number of decimal places for float numbers and multiplying them by a factor that always converts them to integers.

5. Evaluation

5.1. Dataset

The public dataset used in our work comes from [31], which includes traffic flow data of 27 types of devices that are representative of the devices commonly seen in the consumer market. In order to enable each tested device to generate enough training data, the setting process is repeated 20 times. The traffic flow data during each initialization process is packaged into a *pcap* file.

Most of these devices are connected to the network via WiFi or Ethernet, while a few devices use other IoT protocols (such as ZigBee, Z-Wave) to connect to the network indirectly through a HUB.

5.2. Evaluation Setting

All experiments were performed on a server with 36 hyperthreading Intel(R) Xeon(R) Gold 6140 CPU @ 2.30 GHz cores, 128 GB of memory, and VMware ESXi™ 6.7.0 was used to build a computer virtualization platform.

We deployed 5 virtual machines as the baseline environment (as shown in Figure 4), each of them configured with a 2-core CPU, 2 GB RAM, and a hard disk space of 40 GB, running Ubuntu 16.04.2 LTS with GUN/Linux 4.8.0-36-generic kernel. All of them were full nodes (miners) of our private blockchain where a new block was generated in 5 s.

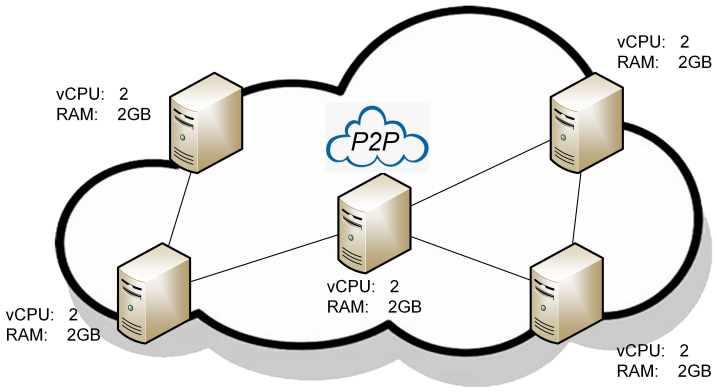

Figure 4. Settings of evaluations.

5.3. Result Analysis

First, we extracted the features of the IoT devices and designed the corresponding weights. The discrimination and stability of different protocols are shown in Table 4. It is worth noting that the TELNET protocol does not appear in the data set, which leads to a situation where the discrimination is 0 while the stability is 1.

Table 4. Discrimination and stability of different protocols.

Protocols	Discrimination	Stability
ARP	0.8567	0.5540
LLC	0.5555	0.8068
IP	0.8741	0.3977
ICMP	0.6492	0.8519
EAPoL	0.8516	0.6648
TCP	0.8869	0.5943
UDP	0.8086	0.5039
HTTP	0.8926	0.8501
HTTPS	0.9285	0.8019
DHCP	0.8432	0.5693
BOOTP	0.8432	0.5693
DNS	0.7929	0.6232
NTP	0.7925	0.7318
TELNET	0.0000	1
Packet length	0.9292	0.7661

Gas [21] is used to measure the "workload" of a behavior or a series of behaviors in Ethereum. Figure 5 shows the execution result of the operation that needs to modify the data on the blockchain in proposed model. The Gas consumption is shown in Table 5.

status	true Transaction mined and execution succeed
transaction hash	0x68a0947672b93ac02d1a5d8cb25b826d9b821fce726f9e800f3a9dedd0206cf5
contract address	0xd9145CCE52D386f254917e481eB44e9943F39138
from	0x5B38Da6a701c568545dCfcB03FcB875f56beddC4
to	DeviceVerified.(constructor)
gas	3000000 gas
transaction cost	1487038 gas
execution cost	1080766 gas
hash	0x68a0947672b93ac02d1a5d8cb25b826d9b821fce726f9e800f3a9dedd0206cf5
input	0x608...60033
decoded input	{}
decoded output	-
value	0 wei

(**a**)

status	true Transaction mined and execution succeed
transaction hash	0x47d080bf9fc2f5efd905410577fa8678dcd28121ab72deb791c724b7d380343f
from	0x5B38Da6a701c568545dCfcB03FcB875f56beddC4
to	DeviceVerified._addNewDevice(string,uint32,uint16[18],uint16[18]) 0xd9145CCE52D386f254917e481eB44e9943F39138
gas	3000000 gas
transaction cost	291998 gas
execution cost	262726 gas
hash	0x47d080bf9fc2f5efd905410577fa8678dcd28121ab72deb791c724b7d380343f
input	0xeb0...00000
decoded input	{ "string _devicename": "aria", "uint32 _judgement": 55557, "uint16[18] _devicefeature": [593, 0, 4000, 7, 207, 3020, 973, 1107, 0, 773, 773, 200, 0, 0, 0, 0, 107, 0], "uint16[18] _weight": [81, 82, 84, 86, 86, 83, 92, 97, 104, 91, 91, 99, 104, 103, 104, 70, 84, 70] }
decoded output	{}
logs	[{ "from": "0xd9145CCE52D386f254917e481eB44e9943F39138", "topic": "0xd125a37b49a474966ae59cc7a52ef6b5b0df63a5058124da100cacf810f945a8", "event": "newDevice", "args": { "0": "aria", "_devicename": "aria" } }]
value	0 wei

(**b**)

status	true Transaction mined and execution succeed
transaction hash	0xe3dbdccb3c2069e8eaea285555cada04b6700409a1dcb0dc255ba970d1bce941
from	0x5B38Da6a701c568545dCfcB03FcB875f56beddC4
to	DeviceVerified._modifyDevice(string,uint32,uint16[18],uint16[18]) 0xd9145CCE52D386f254917e481eB44e9943F39138
gas	3000000 gas
transaction cost	160963 gas
execution cost	131691 gas
hash	0xe3dbdccb3c2069e8eaea285555cada04b6700409a1dcb0dc255ba970d1bce941
input	0x069...00000
decoded input	{ "string _devicename": "aria", "uint32 _judgement": 55558, "uint16[18] _devicefeature": [593, 0, 4000, 7, 207, 3020, 973, 1107, 0, 773, 773, 200, 0, 0, 0, 0, 107, 0], "uint16[18] _weight": [81, 82, 84, 86, 86, 83, 92, 97, 104, 91, 91, 99, 104, 103, 104, 70, 84, 70] }
decoded output	{}
logs	[{ "from": "0xd9145CCE52D386f254917e481eB44e9943F39138", "topic": "0x359283c54874a84aec03ac627311f0f6b93130ba088b05788c1e27c11bba7275", "event": "DeviceModified", "args": { "0": "aria", "_devicename": "aria" } }]
value	0 wei

(**c**)

status	true Transaction mined and execution succeed
transaction hash	0x58179c59bade6ad6004142caffd9a6ee1011326371fa80ec69fb50b8c35280ef
from	0x5B38Da6a701c568545dCfcB03FcB875f56beddC4
to	DeviceVerified._deleteDevice(string)
gas	3000000 gas
transaction cost	33301 gas
execution cost	11261 gas
hash	0x58179c59bade6ad6004142caffd9a6ee1011326371fa80ec69fb50b8c35280ef
input	0x894...00000
decoded input	{ "string _devicename": "aria" }
decoded output	{}
logs	[{ "from": "0xd9145CCE52D386f254917e481eB44e9943F39138", "topic": "0x2e2c62407bf129c3de2419c381debf78f6aa24dd9481f255939de54509d22e96", "event": "DeviceDelete", "args": { "0": "aria", "_devicename": "aria" } }]
value	0 wei

(**d**)

Figure 5. (**a**) Describe the result of smart contract deployment. (**b**) Describe the result of add device fingerprint to smart contract. (**c**) Describe the result of modify device fingerprint in smart contract. (**d**) Describe the result of delete device fingerprint in smart contract.

Table 5. Gas consumption.

Type	Transaction Cost	Execution Cost
Create Contract	1,487,038	1,080,766
Add Device Fingerprint	291,998	262,726
Modify Device Fingerprint	160,963	131,691
Delete Device Fingerprint	33,301	11,261

We evaluated the accuracy of our model and the method from [31] on the same data set. We performed a five-fold cross-validation on the data set. The results (as shown in Figure 6) show that in 17 of 27 types of devices, our mechanism achieved parallel results, but in the remaining 10 types, our method achieved a significant lead, although our feature vectors have a lower dimensionality. The reason is that our model uses a better feature extraction method: the features extracted by our model come from all the network traffic packets of the device in the initialization stage, while our counterparties only utilize the first 12 packets of this stage.

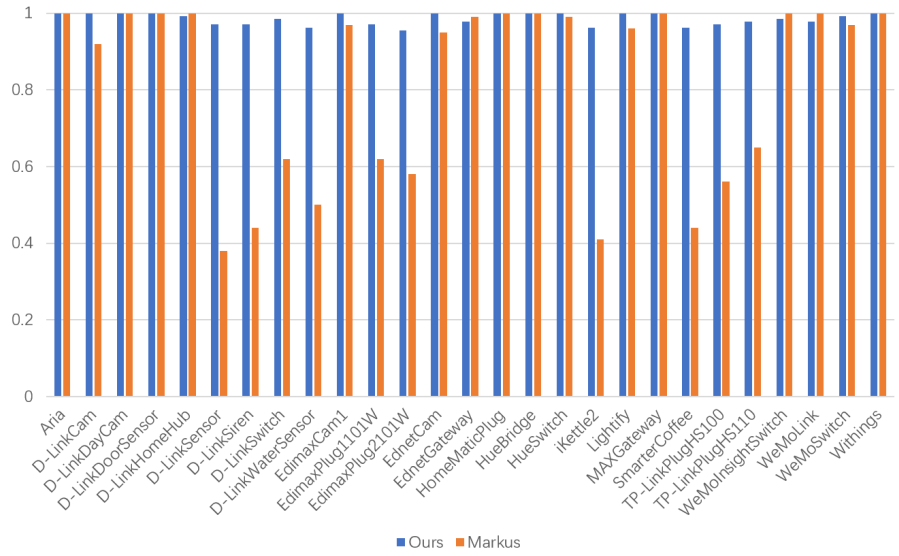

Figure 6. Method comparison.

Another experiment shows the accuracy of *Fraud Detection*, which is used to detect fraudulent device identity behavior, and the result is shown in Figure 7.

In this experiment, we first specified the device type C, and then randomly extracted 100 pcap files from the public data set to simulate the traffic flow data in the initialization phase of 100 IoT devices $D = \{D_1, D_2, \ldots, D_{100}\}$, so that these 100 devices include both normal and fraudulent identities. Finally, we used the model *Fraud Detection*: $J_2(D_i, feature^{D_i}, C), D_i \in D$ to determine whether the device identity is being used fraudulently.

The results in Figure 7 show that for 25 of the 27 types of IoT devices, the accuracy of detecting device identity fraud exceeds 80%, and 21 of which exceed 90%. However, large errors are shown on devices HueSwitch and D-Linkcam. We find that their traffic flow data are extremely unstable, resulting in a large variance in the sample data. As a result, devices that do not originally belong to HueSwitch and D-Linkcam are wrongly classified.

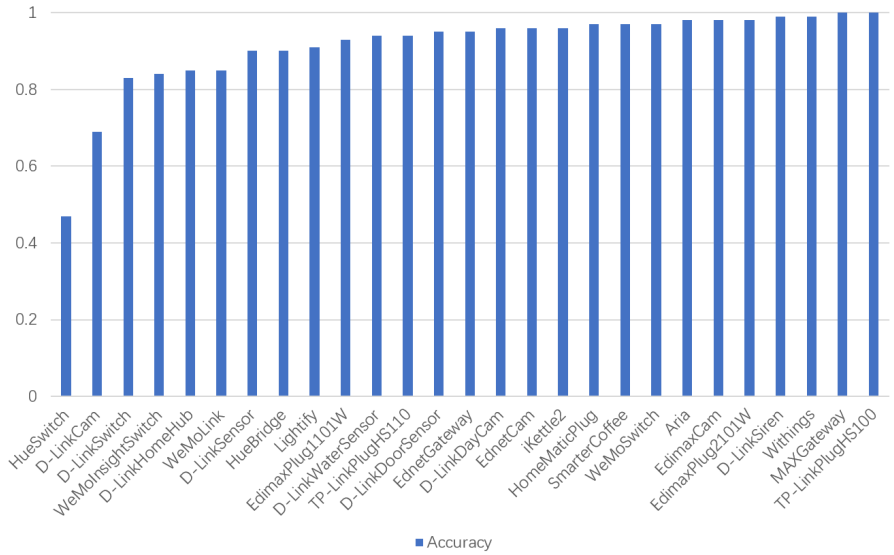

Figure 7. The accuracy of detecting fraudulent use of device identity.

5.4. Time Complexity

When we verify the identity of the IoT device, our model does not modify any data on the blockchain, which means that we can use the *call()* method to trigger the contract in order to save the transaction fee. The execution results of *Register()* and *Detective()* using *call()* are shown in Figure 8.

```
transaction hash  0xa874149619ec8438556b64dc5da50ed4dd353f8ca52d1e41bec9cc3b3e7ac08f
from              0x5B38Da6a701c568545dCfcB03FcB875f56beddC4
to                DeviceVerified.recognizeDevice(uint16[18])
                  0xd9145CCE52D386f254917e481eB44e9943F39138
transaction cost  133347 gas (Cost only applies when called by a contract)
execution cost    108619 gas (Cost only applies when called by a contract)
hash              0xa874149619ec8438556b64dc5da50ed4dd353f8ca52d1e41bec9cc3b3e7ac08f
input             0x780...00000
decoded input     { "uint16[18] _devicefeature": [ 593, 0, 4000, 7, 207, 3020, 973,
                  1107, 0, 773, 773, 200, 0, 0, 0, 0, 107, 0 ] }
decoded output    { "0": "string: _name aria" }
logs              []
```

(**a**)

```
transaction hash  0xa9a7a103ea9ae48ca2935e3d37e057a3419083eb3b01b5f8fb9eb8c44e9313c0
from              0x5B38Da6a701c568545dCfcB03FcB875f56beddC4
to                DeviceVerified.verifiedDevice(string,uint16[18])
                  0xd9145CCE52D386f254917e481eB44e9943F39138
transaction cost  134344 gas (Cost only applies when called by a contract)
execution cost    108784 gas (Cost only applies when called by a contract)
hash              0xa9a7a103ea9ae48ca2935e3d37e057a3419083eb3b01b5f8fb9eb8c44e9313c0
input             0xd75...00000
decoded input     { "string _devicename": "aria", "uint16[18] _devicefeature": [ 600, 0,
                  4000, 0, 300, 3000, 1000, 1100, 0, 800, 800, 200, 0, 0, 0, 0, 100, 0 ]
                  }
decoded output    { "0": "bool: res true" }
logs              []
```

(**b**)

Figure 8. (**a**) Execution results of *Register()*. (**b**) Execution results of *Detective()*.

In the Ethereum private chain, the throughput of transactions depends on the block size and the time to generate new blocks. The problem of transaction delays due to congestion can usually be solved by increasing transaction fees.

We made 1000 calls to the functions *Register()* and *Detective()* on each BCoT Gateway, and obtained the average response time. We calculated the number of requests that each BCoT Gateway can respond to per second, and the result is shown in Figure 9.

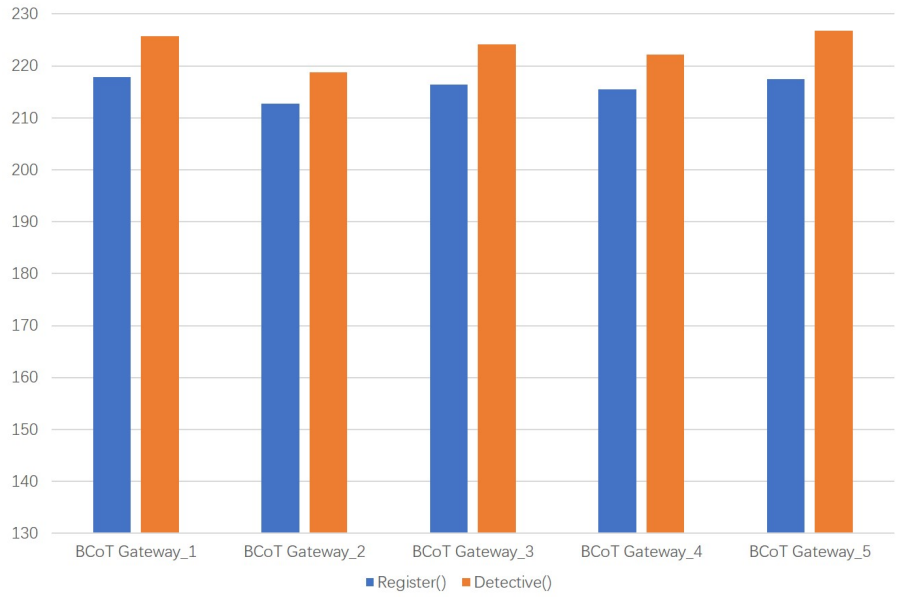

Figure 9. The number of requests responded by each BCoT Gateway per second.

Assuming that the type of IoT device is n and there are m IoT devices that require identity authentication, the two parts of our proposed IoT authentication model Register and Fraud Detection have a time complexity of $O(m * n)$, and $O(m)$.

6. Conclusions and Future Works

Blockchain is a promising security solution for IoT. However, the lightweight feature of IoT devices commonly fails to meet computational intensive requirements for a blockchain-based security model. In this paper, we propose BCoT Sentry, which uses BCoT Gateway to facilitate the recording of authentication transactions in a blockchain network. Furthermore, we introduce a novel device recognition model based on device traffic flow.

We implement a prototype to prove our design and validate the device recognition model on a public dataset. In terms of device recognition, accuracy was more than 95%, and 12 of 27 had 100%. In terms of fraudulent identity detection, our model has an accuracy of over 95% in 21 of 27 types of devices. The number of BCoT Gateways that can respond to *Register()* requests per second is about 215, and to *Detective()* is about 220. These results demonstrate the effectiveness of the proposed framework.

There is still room to improve the current work. Firstly, we tested our framework only on open datasets, and its effectiveness remains to be tested. Secondly, the identity authentication model we proposed is static in terms of the threshold setting and feature weight setting, which requires regular training to update the threshold and feature weight.

In our future work, we will deploy our framework in a real environment for further testing and study how to dynamically adjust the threshold value and feature weight when new data arrives to improve the performance of the model.

Author Contributions: Scheme design, L.G. and D.M.A.; implementation, L.G.; writing—original draft preparation, L.G.; writing—review and editing, D.M.A.; supervision, L.C.; project administration, L.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the West Light Foundation of the Chinese Academic of Sciences, under Grant 2017-XBZG-BR-001, in part by the major science and technology projects in Xinjiang Uygur Autonomous Region, under Grant 2020A03004-4.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

References

1. Lu, Y.; Da Xu, L. Internet of Things (IoT) cybersecurity research: A review of current research topics. *IEEE Internet Things J.* **2018**, *6*, 2103–2115. [CrossRef]
2. Ahmed, H.I.; Nasr, A.A.; Abdel-Mageid, S.; Aslan, H.K. A survey of IoT security threats and defenses. *Int. J. Adv. Comput. Res.* **2019**, *9*, 325–350. [CrossRef]
3. Khelloufi, A.; Ning, H.; Dhelim, S.; Qiu, T.; Ma, J.; Huang, R.; Atzori, L. A Social Relationships Based Service Recommendation System For SIoT Devices. *IEEE Internet Things J.* **2020**, *8*, 1859–1870. [CrossRef]
4. Nižetić, S.; Šolić, P.; González-de, D.L.D.I.; Patrono, L. Internet of Things (IoT): Opportunities, issues and challenges towards a smart and sustainable future. *J. Clean. Prod.* **2020**, *274*, 122877. [CrossRef] [PubMed]
5. Li, X.; Wang, H.; Dai, H.N.; Wang, Y.; Zhao, Q. An analytical study on eavesdropping attacks in wireless nets of things. *Mob. Inf. Syst.* **2016**, *2016*, 4313475. [CrossRef]
6. Sapienza, A.; Bessi, A.; Damodaran, S.; Shakarian, P.; Lerman, K.; Ferrara, E. Early warnings of cyber threats in online discussions. In Proceedings of the 2017 IEEE International Conference on Data Mining Workshops (ICDMW), New Orleans, LA, USA, 18–21 November 2017; pp. 667–674.
7. Hammi, M.T.; Hammi, B.; Bellot, P.; Serhrouchni, A. Bubbles of Trust: A decentralized blockchain-based authentication system for IoT. *Comput. Secur.* **2018**, *78*, 126–142. [CrossRef]
8. Nofer, M.; Gomber, P.; Hinz, O.; Schiereck, D. Blockchain. *Bus. Inf. Syst. Eng.* **2017**, *59*, 183–187. [CrossRef]
9. Reyna, A.; Martín, C.; Chen, J.; Soler, E.; Díaz, M. On blockchain and its integration with IoT. Challenges and opportunities. *Future Gener. Comput. Syst.* **2018**, *88*, 173–190. [CrossRef]
10. Dai, H.N.; Zheng, Z.; Zhang, Y. Blockchain for Internet of Things: A survey. *IEEE Internet Things J.* **2019**, *6*, 8076–8094. [CrossRef]
11. Yang, Y.; Wu, L.; Yin, G.; Li, L.; Zhao, H. A survey on security and privacy issues in Internet-of-Things. *IEEE Internet Things J.* **2017**, *4*, 1250–1258. [CrossRef]
12. Farwell, J.P.; Rohozinski, R. Stuxnet and the future of cyber war. *Survival* **2011**, *53*, 23–40. [CrossRef]
13. Maggi, F.; Quarta, D.; Pogliani, M.; Polino, M.; Zanchettin, A.M.; Zanero, S. *Rogue Robots: Testing the Limits of an Industrial Robot's Security*; Trend Micro, Politecnico di Milano, Tech. Rep; Trend Micro: San Francisco, CA, USA, 2017.
14. Quarta, D.; Pogliani, M.; Polino, M.; Maggi, F.; Zanchettin, A.M.; Zanero, S. An experimental security analysis of an industrial robot controller. In Proceedings of the 2017 IEEE Symposium on Security and Privacy (SP), San Jose, CA, USA, 22–26 May 2017; pp. 268–286.
15. Gope, P.; Das, A.K.; Kumar, N.; Cheng, Y. Lightweight and Physically Secure Anonymous Mutual Authentication Protocol for Real-Time Data Access in Industrial Wireless Sensor Networks. *IEEE Trans. Ind. Inform.* **2019**, *15*, 4957–4968. [CrossRef]
16. Laufs, J.; Borrion, H.; Bradford, B. Security and the smart city: A systematic review. *Sustain. Cities Soc.* **2020**, *55*, 102023. [CrossRef]
17. Mohit, P.; Amin, R.; Biswas, G. Design of authentication protocol for wireless sensor network-based smart vehicular system. *Veh. Commun.* **2017**, *9*, 64–71. [CrossRef]
18. Nakamoto, S. *Bitcoin: A Peer-to-Peer Electronic Cash System*; Technical Report, Manubot. 2019. Available online: https://git.dhimmel.com/bitcoin-whitepaper/ (accessed on 9 May 2021).
19. Chohan, U.W. The double spending problem and cryptocurrencies. Available at SSRN 3090174. 2017. Available online: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3090174 (accessed on 9 May 2021).
20. Lamport, L.; Shostak, R.; Pease, M. The Byzantine Generals Problem. In *Concurrency: The Works of Leslie Lamport*; Association for Computing Machinery: New York, NY, USA, 2019; pp. 203–226.
21. Wood, G. Ethereum: A secure decentralised generalised transaction ledger. *Ethereum Proj. Yellow Pap.* **2014**, *151*, 1–32.
22. Androulaki, E.; Barger, A.; Bortnikov, V.; Cachin, C.; Christidis, K.; De Caro, A.; Enyeart, D.; Ferris, C.; Laventman, G.; Manevich, Y.; et al. Hyperledger fabric: A distributed operating system for permissioned blockchains. In Proceedings of the Thirteenth EuroSys Conference, Porto, Portugal, 23–26 April 2018; pp. 1–15.
23. Zheng, Z.; Xie, S.; Dai, H.N.; Chen, W.; Chen, X.; Weng, J.; Imran, M. An overview on smart contracts: Challenges, advances and platforms. *Future Gener. Comput. Syst.* **2020**, *105*, 475–491. [CrossRef]
24. Luu, L.; Chu, D.H.; Olickel, H.; Saxena, P.; Hobor, A. Making smart contracts smarter. In Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna, Austria, 24–28 October 2016; pp. 254–269.
25. Herlihy, M. Atomic cross-chain swaps. In Proceedings of the 2018 ACM Symposium on Principles of Distributed Computing, Egham, UK, 23–27 July 2018; pp. 245–254.
26. Kokoris-Kogias, E.; Jovanovic, P.; Gasser, L.; Gailly, N.; Syta, E.; Ford, B. Omniledger: A secure, scale-out, decentralized ledger via sharding. In Proceedings of the 2018 IEEE Symposium on Security and Privacy (SP), San Francisco, CA, USA, 20–24 May 2018; pp. 583–598.

27. Karlsson, K.; Jiang, W.; Wicker, S.; Adams, D.; Ma, E.; van Renesse, R.; Weatherspoon, H. Vegvisir: A partition-tolerant blockchain for the internet-of-things. In Proceedings of the 2018 IEEE 38th International Conference on Distributed Computing Systems (ICDCS), Vienna, Austria, 2–6 July 2018; pp. 1150–1158.
28. Xiao, L.; Wan, X.; Lu, X.; Zhang, Y.; Wu, D. IoT security techniques based on machine learning: How do IoT devices use AI to enhance security? *IEEE Signal Process. Mag.* **2018**, *35*, 41–49. [CrossRef]
29. Conti, M.; Dehghantanha, A.; Franke, K.; Watson, S. Internet of Things security and forensics: Challenges and opportunities. *Future Gener. Comput. Syst.* **2018**, *78*, 544–546. [CrossRef]
30. Mnif, A.; Cheikhrouhou, O.; Jemaa, M.B. An ID-based user authentication scheme for Wireless Sensor Networks using ECC. In Proceedings of the ICM 2011 Proceeding, Hammamet, Tunisia, 19–22 December 2011; pp. 1–9.
31. Miettinen, M.; Marchal, S.; Hafeez, I.; Asokan, N.; Sadeghi, A.R.; Tarkoma, S. Iot sentinel: Automated device-type identification for security enforcement in iot. In Proceedings of the 2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS), Atlanta, GA, USA, 5–8 June 2017; pp. 2177–2184.
32. Peng, L.; Hu, A.; Zhang, J.; Jiang, Y.; Yu, J.; Yan, Y. Design of a hybrid RF fingerprint extraction and device classification scheme. *IEEE Internet Things J.* **2018**, *6*, 349–360. [CrossRef]
33. Venkatraman, S.; Kumar, P.A.R. Improving Adhoc wireless sensor networks security using distributed automaton. *Clust. Comput.* **2019**, *22*, 14551–14557. [CrossRef]
34. Stellios, I.; Kotzanikolaou, P.; Psarakis, M.; Alcaraz, C.; Lopez, J. A survey of iot-enabled cyberattacks: Assessing attack paths to critical infrastructures and services. *IEEE Commun. Surv. Tutor.* **2018**, *20*, 3453–3495. [CrossRef]
35. Yakubov, A.; Shbair, W.; Wallbom, A.; Sanda, D. A blockchain-based pki management framework. In Proceedings of the First IEEE/IFIP International Workshop on Managing and Managed by Blockchain (Man2Block) Colocated with IEEE/IFIP NOMS 2018, Tapei, Tawain, 23–27 April 2018.
36. Singla, A.; Bertino, E. Blockchain-Based PKI Solutions for IoT. In Proceedings of the 2018 IEEE 4th International Conference on Collaboration and Internet Computing (CIC), Philadelphia, PA, USA, 18–20 October 2018; pp. 9–15. [CrossRef]
37. Zheng, Z.; Xie, S.; Dai, H.N.; Chen, X.; Wang, H. Blockchain challenges and opportunities: A survey. *Int. J. Web Grid Serv.* **2018**, *14*, 352–375. [CrossRef]
38. Gervais, A.; Karame, G.O.; Wüst, K.; Glykantzis, V.; Ritzdorf, H.; Capkun, S. On the security and performance of proof of work blockchains. In Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna, Austria, 24–28 October 2016; pp. 3–16.
39. King, S.; Nadal, S. Ppcoin: Peer-to-peer crypto-currency with proof-of-stake. *Self-Publ. Pap. August* **2012**, *19*, 1.
40. Castro, M.; Liskov, B. Practical byzantine fault tolerance. *OSDI* **1999**, *99*, 173–186.
41. Liu, Y.; Wang, K.; Lin, Y.; Xu, W. LightChain: A Lightweight Blockchain System for Industrial Internet of Things. *IEEE Trans. Ind. Inform.* **2019**, *15*, 3571–3581. [CrossRef]
42. Koshy, P.; Babu, S.; Manoj, B. Sliding window blockchain architecture for internet of things. *IEEE Internet Things J.* **2020**, *7*, 3338–3348. [CrossRef]
43. Ellul, J.; Pace, G.J. Alkylvm: A virtual machine for smart contract blockchain connected internet of things. In Proceedings of the 2018 9th IFIP International Conference on New Technologies, Mobility and Security (NTMS), Paris, France, 26–28 February 2018; pp. 1–4.
44. Gochhayat, S.P.; Bandara, E.; Shetty, S.; Foytik, P. Yugala: Blockchain Based Encrypted Cloud Storage for IoT Data. In Proceedings of the 2019 IEEE International Conference on Blockchain (Blockchain), Atlanta, GA, USA, 14–17 July 2019; pp. 483–489.
45. Axon, L.; Goldsmith, M. PB-PKI: A Privacy-aware Blockchain-based PKI. In Proceedings of the 14th International Joint Conference on e-Business and Telecommunications—Volume 4: SECRYPT, (ICETE 2017), INSTICC, SciTePress, Madrid, Spain, 24–26 July 2017; pp. 311–318. [CrossRef]
46. Cruz, J.P.; Kaji, Y.; Yanai, N. RBAC-SC: Role-based access control using smart contract. *IEEE Access* **2018**, *6*, 12240–12251. [CrossRef]
47. Bouras, M.A.; Xia, B.; Abuassba, A.O.; Ning, H.; Lu, Q. IoT-CCAC: A blockchain-based consortium capability access control approach for IoT. *PeerJ Comput. Sci.* **2021**, *7*, e455. [CrossRef] [PubMed]
48. Cui, H.; Chen, Z.; Xi, Y.; Chen, H.; Hao, J. IoT data management and lineage traceability: A blockchain-based solution. In Proceedings of the 2019 IEEE/CIC International Conference on Communications Workshops in China (ICCC Workshops), Changchun, China, 11–13 August 2019; pp. 239–244.
49. Bouras, M.A.; Lu, Q.; Dhelim, S.; Ning, H. A Lightweight Blockchain-Based IoT Identity Management Approach. *Future Internet* **2021**, *13*, 24. [CrossRef]
50. Omar, A.S.; Basir, O. Capability-based non-fungible tokens approach for a decentralized AAA framework in IoT. In *Blockchain Cybersecurity, Trust and Privacy*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 7–31.
51. Guin, U.; Cui, P.; Skjellum, A. Ensuring proof-of-authenticity of iot edge devices using blockchain technology. In Proceedings of the 2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), Halifax, NS, Canada, 30 July–3 August 2018; pp. 1042–1049.

52. Alblooshi, M.; Salah, K.; Alhammadi, Y. Blockchain-based ownership management for medical IoT (MIoT) devices. In Proceedings of the 2018 International Conference on Innovations in Information Technology (IIT), Al Ain, United Arab Emirates, 18–19 November 2018; pp. 151–156.
53. Dorri, A.; Kanhere, S.S.; Jurdak, R. Towards an Optimized BlockChain for IoT. In Proceedings of the 2017 IEEE/ACM Second International Conference on Internet-of-Things Design and Implementation (IoTDI), Pittsburgh, PA, USA, 18–21 April 2017; pp. 173–178.
54. Zhu, Q.; Wang, R.; Chen, Q.; Liu, Y.; Qin, W. Iot gateway: Bridgingwireless sensor networks into internet of things. In Proceedings of the 2010 IEEE/IFIP International Conference on Embedded and Ubiquitous Computing, Hong Kong, China, 11–13 December 2010; pp. 347–352.
55. Reshef, D.N.; Reshef, Y.A.; Finucane, H.K.; Grossman, S.R.; McVean, G.; Turnbaugh, P.J.; Lander, E.S.; Mitzenmacher, M.; Sabeti, P.C. Detecting novel associations in large data sets. *Science* **2011**, *334*, 1518–1524. [CrossRef]
56. Kumar, A.; Lim, T.J. Early detection of Mirai-like IoT bots in large-scale networks through sub-sampled packet traffic analysis. In *Future of Information and Communication Conference*; Springer: Berlin/Heidelberg, Germany, 2019; pp. 847–867.
57. Scapy 2.4.5. Available online: https://scapy.readthedocs.io/en/latest/introduction.html (accessed on 30 March 2021).
58. Web3.py 5.17.0. Available online: https://web3py.readthedocs.io/en/stable/ (accessed on 30 March 2021).
59. Solidity 0.8.0. Available online: https://docs.soliditylang.org/en/v0.8.0/ (accessed on 30 March 2021).