Blockchain-Enabled SDN for Securing Fog-Based Resource-Constrained IoT

Sudip Misra, *Senior Member, IEEE*, Pallav Kumar Deb, *Student Member, IEEE*, Nidhi Pathak, *Student Member, IEEE*, and Anandarup Mukherjee, *Student Member, IEEE*

***Abstract*—Software-Defined Network (SDN) is vital in simplifying the dynamic network characteristics and device management. However, the centralized architecture of SDN opens the scope for malicious attacks on the controllers. To mitigate such attacks in real-time, we propose an SDN architecture for resource-constrained devices in a fog-enabled IoT environment using a private blockchain (pBC) network. We exploit the decentralized nature of pBC for enabling resource-constrained SDN controllers towards transparently setting flow rules for fog nodes and other devices in the network. In case the miners identify faulty flow rules, pBC allows the SDN devices/fog nodes to retract back to an earlier flow rule while raising a flag against the alleged controller. Additionally, since data in pBC are accessible by all the candidates having the same genesis file, they are readily available to malicious users. Towards this, we further propose encrypting the data before inserting them into the blocks, which helps in securing the data from undesired users. Through the extensive deployment of our proposed fusion, we observe CPU usage of 30% among the devices and latencies in the range of milliseconds, which presents the feasibility of our system with minimum delay. We also observe a reduction in energy consumption by more than 90%, compared to traditional SDN.**

***Index Terms*—Internet of Things, fog computing, software-defined networking, blockchain, encryption, security**

I. Introduction

SDN offers simplicity in performing network operations by separating the control and data planes in IoT environments. The controllers in a typical SDN environment are responsible for setting rules for the user devices and switches/routers for their network operations and usually have high computational configurations. Attacks on such controllers have the potential to disrupt the entire network irrevocably. On detecting anomalies, the administrators need to restore the controller as well as modify the network to resume functioning. To deal with such issues, we present an *affordable* and *easy-to-deploy* architecture for resource-constrained fog nodes to act as SDN controllers while dealing with high traffic from the user devices. In this work, we present an implementation of these resource-constrained fog nodes as controllers that set flow rules for the other devices in the network, separating the control plane from the data plane. The centralized architecture of SDN is also vulnerable to issues related to the single point of failure. We overcome this by implementing a private blockchain (pBC) among our devices, which decentralizes our system. This setup is comparable to that of a typical SDN architecture. However, our primary focus is on the implementation and deployment of an autonomous, decentralized, low-cost, and sustainable SDN for resource-constrained devices, while securing the controllers and their corresponding flow rules.

Sudip Misra, Pallav Kumar Deb, and Anandarup Mukherjee are with the Department of Computer Science and Engineering at Indian Institute of Technology Kharagpur, India

Nidhi Pathak is with the Advanced Technology Development Center at Indian Institute of Technology Kharagpur, India

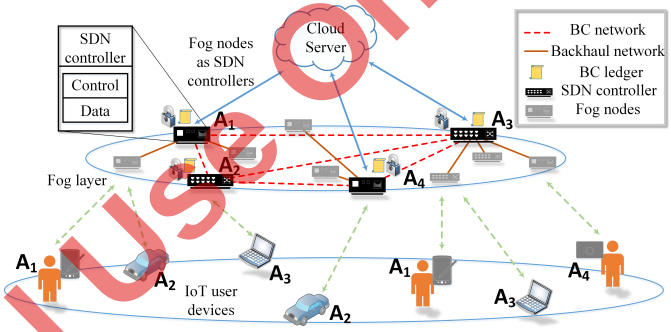


Fig. 1. Proposed fusion of pBC with SDN in a fog-enabled IoT environment serving heterogeneous applications

In this work, to overcome the security issues of SDN, we propose a blockchain-enabled *secured distributed architecture* for SDN controllers operating in the *fog layer*. In the scenario outlined in Fig. 1, a set of SDN controllers $(\mathcal{C})$ serving K different applications exists, such that $\mathcal{C} = \{c_{A_1}, c_{A_2}, ...c_{A_K}\}$. Each of these controllers set flow rules for a set of q SDN switches/routers (fog nodes) $\mathcal{S} = \{s_1, s_2, ...s_q\}$ in the fog layer. We consider these SDN controllers to be in connection to one another over a *private blockchain* (pBC) network, irrespective of the assigned applications. In other words, the controllers communicate over the pBC network using east-west communications. The SDN controller c_X for the X^{th} application sets flow rules as $F_{t_a}^{c_X}$ with timestamp t_a. We propose the storage of these flow rules in the pBC in the form of blocks. Due to the features of pBC, such as transparency and its decentralized architecture, the flow rules are visible to the controllers and other entities that are part of the same pBC. Additionally, each of the controllers are in communication with the cloud, which keeps track of the modifications concerning the flow rules. In case the attackers get access to one of the controllers c_X^{atk} and set malicious flow rules $F_{t_a}^{c_X^{atk}}$, the miners in the pBC have the authority to discard $F_{t_a}^{c_X^{atk}}$, and raise a flag against c_X^{atk}. In case the miners fail to identify $F_{t_a}^{c_X^{atk}}$, the cloud servers responsible for

the particular application recognizes the anomaly, deletes the block corresponding to $F_{t_a}^{c_X^{atk}}$, and retracts back to the flow rule $F_{t_a-1}^{c_X}$, while raising a flag against the controller c_X^{atk}. It may be noted that due to the immutable nature of pBC, the attackers cannot change the blocks already present in the chain. Further, as pBC offers integrity to the contents in each block, they are accessible to all the users with the same genesis file. To cope with this, we propose encrypting the data before inserting them into the pBC. Such encryption schemes help in adding more security to the data.

Example Scenario: As illustrated in Fig. 1, the fog layer offers a plethora of services. Consider a request for application $\mathcal{A}$. The data from the user follows an optimized path directed by SDN controllers to the concerned applications (fog nodes) and vice versa [1]. In case the flow rules from such controllers are compromised, the data will not reach their destinations. Such threats are hazardous and need attention. Towards this, the pre-detection of faulty flow rules is beneficial. Additionally, once the network behavior degrades, it is essential to recover at the earliest. To address these issues, we propose the fusion of a pBC with SDN where – 1) the miners identify faulty flow rules before putting it in the block and 2) the BC provides easy retraction to correct set of flow rules.

A. Motivation

The current deployment of SDN technologies is dependent on switches/routers with high configurations. Since fog nodes usually consist of resource-constrained devices, we implement a new architecture with a set of lightweight routines for determining flow rules. Additionally, due to centralized controllers, SDN opens the scope for attacks from adversaries, which may bring down the entire network. Such threats mandate the need for new methods for securing communications. With the use of pBC, we offer a decentralized solution, which is *affordable* and *easily deployable* in the current network infrastructure. Further, as fog computing operates with low power consumption and latency for IoT environments [2], we implement and deploy the pBC in the fog layer. Each of the blocks in the pBC contains flow rules as its contents, which are visible to all the controllers and entities that are part of the pBC and have the same genesis file. In case of faulty flow rules, the pBC allows easy retraction to a previously working set of rules. However, the pBC does not offer security as the data are readily available to its participants with the same genesis file, which makes the data vulnerable to attackers. To overcome such attacks, we encrypt the data before entering them into the blocks. Encrypting data contents within each block increases the security of the pBC, which secures the network as a whole.

B. Contribution

In this work, we implement and deploy the fusion of pBC with SDN on a fog-enabled IoT environment for enhancing security for the SDN controllers. pBC allows easy retraction to the set of previously running flow rules. Further, encrypting the data contents of the blocks in a pBC controls unauthorized accesses. Towards this, the key contributions in this work are:

- ***Resource-constrained devices:*** SDN solutions are usually dependent on devices with high configurations. However, fog nodes are usually resource-constrained devices in terms of storage and computational capability, implying the need for lightweight routines for determining the flow rules. We also design the controllers to be in communication with concerned cloud servers for detecting anomalies when the miners fail to recognize faulty flow rules.
- ***Private Blockchain:*** We design the SDN controllers such that they share their flow rules via blocks in a pBC. With the implementation of pBC, in case the SDN controllers are compromised, the retraction to a previous set of flow rules is relatively more straightforward. Such easy retraction is possible as each block in the pBC has a link to the previous block. Additionally, the pBC network also helps in maintaining the integrity of the flow rules, which cannot be modified by adversaries.
- ***Encryption:*** Miners do not need to be provided with incentives for mining in pBC, which is an attractive feature. However, the pBC also has a few limitations, such as lack of access control as the data (flow rules) is available to all of its participants. To secure these flow rules from attackers, we encrypt the data before inserting it into the blocks. Although such encryption schemes add more overhead on the resource-constrained fog nodes, they help in securing the data against undesired access.

II. Related Work

SDN provides seamless networking by separating the control and data planes. However, the centralized architecture of SDN opens the scope for attacks. Towards this, researchers have been developing ways of securing the controllers as well as user devices and switches.

A. SDN and Security

TENNISON [3] introduces a decentralized SDN structure for enhancing security. The decentralized system also helps in reducing operations on the control plane by distributing the workload among the controllers. The recognition of the defecting devices is also of paramount importance. Towards this, the authors in [4] designed a method for recognizing compromised devices in real-time. They proposed deploying backup controllers that vigilantly monitor network updates from the primary controller and the subsequent behavior of the user devices, switches, and routers. Researchers have also developed a language-based policymaking scheme for securing the services [5]. They devised their scheme to secure end-to-end devices across SDNs operating in multiple domains.

B. Blockchain and SDN

Denial of Service (DoS) and Distributed DoS (DDoS) attacks are usually common in SDN architectures. Cochain-SC [6] combines intra and inter-domain DDoS mitigation schemes for providing affordable, efficient, and flexible security options

against DDoS attacks in SDN using blockchain. Data and files shared among devices in an SDN also need security against attackers. The authors in [7] proposed a blockchain-enabled SDN named BSS, for securing files and information in the data plane. Researchers have also been using blockchain for determining trust among devices that exchange data in an SDN. Nokia Bell Labs developed such a system named STewARD [8], for maintaining trust among devices in a home network, which the controllers use in forming network slices, respectively. Based on increasing or decreasing trust levels, STewARD keeps updating the network slices, respectively. SDN has also proved beneficial in 5G-vehicular ad-hoc networks (5G-VANETs) [9]. The authors used the decentralized nature of blockchain for enhancing the security framework of VANETS, which enabled the detection of malicious nodes with acceptable network performance. The authors in [10] developed a two-tier distributed architecture by deploying blockchain networks on both cloud and fog layers for securing the IoT network. Apart from these, researchers are also using blockchains for routing data packets in SDN [11]. The authors stored latencies in the blockchain network, which the autonomous systems access with ease. Such storage of information in a decentralized manner enabled latency-aware routing by the autonomous systems.

Synthesis: Researchers have been exploiting SDN for the ease of controlling the network. Additionally, the unique features of blockchain, such as its decentralized structure and security, have motivated researchers in creating a fusion of the two technologies (SDN and blockchain) for enhancing security in SDNs. However, the current literature has focussed primarily on the devices in the data plane and its trust level in the network. They have also used the blocks in the blockchain for securing the data from these user devices. In this work, we focus on securing the centralized SDN controllers from malicious attacks and its recovery in IoT environments. Further, data in pBC are readily available to all the participants with the same genesis file, which increases the risk of data access by undesirable users. Towards this, we propose encrypting the data before inserting them into the blocks of pBC.

III. System Model

In this section, we present our problem scenario and our proposed solution for securing the SDN controllers in a fog-enabled IoT environment. Additionally, we also present the encryption schemes incorporated for securing the flow rules in the pBC from malicious attackers.

A. Problem Scenario

As shown in Fig. 1, we consider an IoT environment where service providers offer a range of applications $\mathcal{A} = \{A_1, A_2, ..., A_z\}$. For operational simplicity, these service providers use SDN as it separates the control and data planes, respectively. We envision the controllers to operate in the fog layer, for directing the traffic to and from IoT user devices $\mathcal{U} = \{u_1, u_2, ..., u_s\}$ and the fog nodes $\mathcal{S} = \{s_1, s_2, ..., s_q\}$. Some of the fog nodes $\mathcal{C} = \{c_{A_1}, c_{A_2}, ...c_{A_K}\}$, where $\mathcal{C} \subset \mathcal{S}$, acting as SDN controllers are responsible for these applications, depending on their corresponding service providers and topology. The IoT users/devices request for a set of applications from $\mathcal{A}$. The SDN controllers set flow rules for these requests from $\mathcal{U}$ as well as $\mathcal{F}$. Due to the centralized architecture, these SDN controllers are of paramount importance, which opens the scope for threats from malicious attackers.

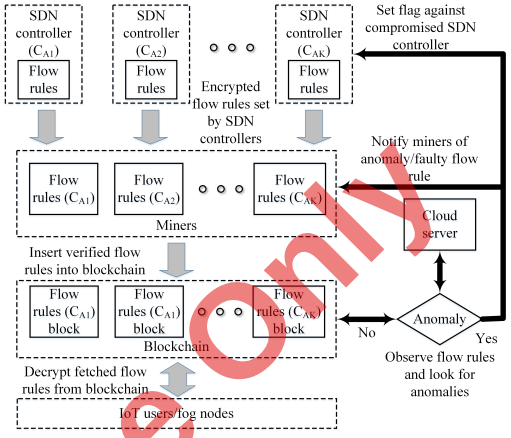


Fig. 2. Information flow among devices in the proposed pBC-enabled SDN system

B. Solution Approach

In this work, we make a trivial assumption that the SDN controllers are in connection with one another irrespective of the concerned applications via east-west communication. We also assume that the fog nodes acting as SDN controllers simultaneously execute networking operations as well as the routines for setting flow rules for the IoT users and other fog nodes. To secure these controllers, we propose decentralizing the SDN architecture by incorporating a pBC network among them. Blockchain is a public ledger that is distributed and decentralized, where the miners verify its contents. Analogous to this, the fog nodes in this work set flow rules and send it to pre-defined miners. Consider $F_{t_a}^{c_X}$ as a flow rule set by controller c for application $X \epsilon \mathcal{A}$ at timestamp t_a. In case the controller gets compromised (c_X^{atk}) at timestamp t_{a+1} and sets faulty flow rules $(F_{t_{a+1}}^{c_X^{atk}})$ for the pBC, the miners may be able to identify them. In case the miners identify the anomaly, the system does not need to change as the miners will not create any block for $F_{t_{a+1}}^{c_X^{atk}}$. On the other hand, in case the miners create new blocks for $F_{t_{a+1}}^{c_X^{atk}}$ and the fog nodes $\mathcal{S}$ adopt the new rules, we envision the cloud servers to identify the anomaly and order the SDN controllers to retract back to the block with flow rule $F_{t_a}^{c_X}$. This architecture of pBC makes the storage of the flow rules decentralized. It also ensures data integrity through its property of immutability. Fig. 2 illustrates the flow of information among the SDN controllers, miners, IoT devices/fog nodes, and the cloud server.

1) Three Step Retraction: In step 1 of Fig. 3, the cloud servers identify the anomaly and advise retraction to a previous

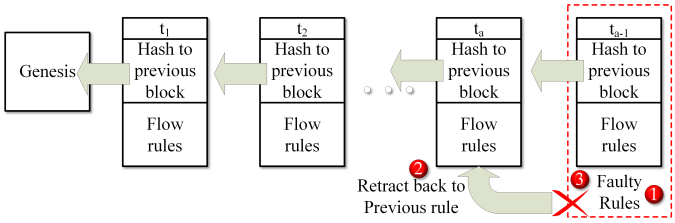


Fig. 3. Retraction to previously set flow rules in the blockchain in case of detecting anomalies

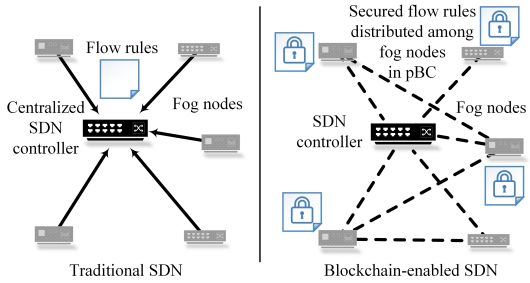


Fig. 4. Decentralization of flow rules as opposed to the traditional centralized SDN architecture

block (step 2). It may be noted that, as the retraction depends on the number of applications running in the pBC, the required block may not be the immediately preceding one. Finally, the cloud orders the miners to delete the block $F_{t_{a+1}}^{c_X^{atk}}$ (step3). In the future, we plan to autonomously guide the fog nodes to identify network stalling and denial of services.

2) Encryption of Flow Rules: Data from IoT devices may be sensitive and have consequences in case attackers get access to them. In other words, although pBC offers immutibibilty of the data, they do not offer data security from devices with the same hashkey and genesis file. To ensure data security, we propose encrypting the flow rules using algorithms such as AES-128, AES-256, or RSA before inserting them into the blocks. The choice of encryption schemes may vary depending on the device configurations and key exchange policies. In either case, encrypting the restricts undesirable access.

IV. Performance Evaluation

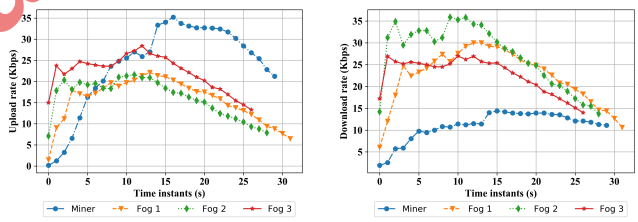
To evaluate our system in lab scale, we implement pBC on a set of resource-constrained Raspberry Pi devices as SDN controllers and fog nodes. On the other hand, we use systems with $i5$ processors as miners. Since we use pBC, we do not need to provide incentives for mining. We use Python 3.7 for programming the devices and go-ethereum for implementing the pBC network among them. Additionally, we use 256 bit AES and RSA algorithms for encrypting the flow rules from the SDN controllers and show the variation in performance. In this section, we present the performance demonstrated by the devices mentioned above on running our experiments. As we implement lightweight flow rules, an arbitrary subset of fog nodes may assume the role of SDN controllers. In this section, we refer to the devices as $\{\text{Fog } 1, \text{Fog } 2, ..., \text{Fog q}\}$, irrespective of their roles in the network.

A. Decentralization

The proposed fusion of pBC with SDN helps in decentralizing the traditionally centralized architecture. As shown in Fig. 4, traditional SDN controllers have central authority over the network (left). Attacks from malicious users disrupt the network in its entirety until repaired. On the other hand, our proposed system stores the flow rules on multiple devices with pBC ensuring its immutability (right). The failure of SDN controllers, in this case, does not cause loss of communication among the other devices while awaiting repair from administrators. As mention in Section III-B1, in case an SDN controller is compromised, the other fog nodes retract back to a correct set of flow rules in 3 steps. In the future, we plan to devise autonomous ways of detecting infected flow rules efficiently. Additionally, as data in pBC is readily available to all the participants, the use of encryption schemes helps in minimizing undesirable access to the flow rules.

TABLE I
Data rates while executing SDN-pBC architecture

Data rates		Miner	Fog 1	Fog 2	Fog 3
Upload	**min**	0.15	1.47	7.08	13.30
	max	35.20	22.10	21.60	28.40
	avg	23.91	15.49	16.17	22.08
Download	**min**	1.91	6.04	13.7	14.0
	max	14.4	30.0	35.9	27.0
	avg	11.01	22.21	27.04	22.67



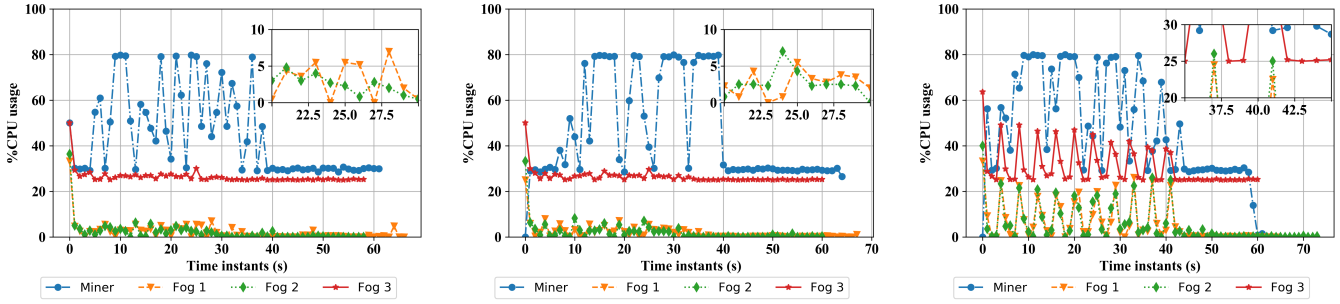
(a) Upload rate (b) Download rate

Fig. 5. Upload and download rates of resource-constrained fog nodes and miners while testing pBC-enabled SDN

B. Upload & Download Data Rates

In a pBC, as the devices in the network commit transactions, the concerned miners *mine* the data before inserting them into blocks. Towards this, we present the upload and download data rates of 3 fog nodes and a miner in Fig. 5. At the start of our implementation, the fog nodes set flow rules (in the form of transactions), which shows a relatively higher upload rates in the range of 25 Kbps, as shown in Fig. 5(a). On creation of a new block, the miner notifies all the other fog nodes. Due to this broadcast, we observe a rise in the upload rate upto 35 Kbps in case of the miner. As the fog nodes complete submitting transactions, their upload rates decrease.

Due to the broadcast from the miner, the download rates rises upto 35 Kbps in case of the fog nodes in Fig. 5(b). In this stage, as the miner does not receive any data from the



(a) No encryption (b) RSA (c) AES-256

Fig. 6. Comparison of CPU usage on running pBC without and with encrypting (RSA and AES-256) flow rules in the fog nodes

TABLE II
CPU USAGE (IN PERCENT) WHILE EXECUTING SDN-PBC ARCHITECTURE

Encryption type		Miner	Fog 1	Fog 2	Fog 3
None	min	28.5	0	0	25.0
	max	79.80	33.3	36.4	50.0
	avg	44.49	2.14	2.09	26.36
RSA	min	0	0	0	25.0
	max	79.90	25.0	33.30	50.0
	avg	46.08	1.89	2.03	26.42
AES-256	min	0	0	0	25
	max	80	33.30	40.0	63.60
	avg	47.77	5.99	5.38	30.59

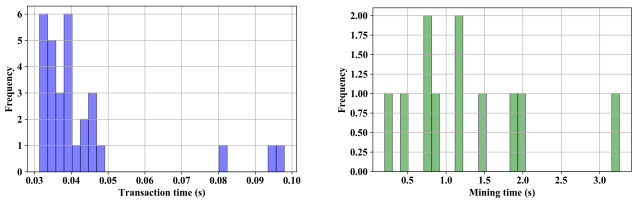
fog nodes, its download rate falls down to 10 Kbps. Although the rates vary in each of the fog nodes, the trend remains the same in each case. We attribute the difference in the download rates of the fog nodes to the behaviour of pBC as its inbuilt routines set the sending/receiving schedule. Table I depicts the observed minimum, maximum, and the average data rates.

Finally, on updating the pBC, the data rates in both the cases fall to the minimum. In case of no activity, the devices keep probing the pBC for any new updates.

C. CPU Usage

We run our experiments under two modes – 1) without encrypting and 2) by encrypting the flow rules in the SDN controllers and summarize the results in Fig. 6. In case of no encryption routine, the devices only run the pBC. In other words, the SDN controllers set the flow rules and submit them in the form of transactions, which the miners verify while mining. Since mining is a relatively complex operation, we observe 80% CPU utilization in the case of miners in Fig. 6(a). On the other hand, the fog nodes have much lower CPU utilization. However, compared to other fog nodes, Fog 3 demonstrates a higher CPU utilization of 25%. We attribute this behavior as the fog nodes perform multiple operations simultaneously, implying that the fog nodes perform SDN and pBC operations while serving other requests.

On the other hand, on encrypting the flow rules, we observe increased CPU utilization in Figs. 6(b) and 6(c). In the case of miners, although we observe the same maximum CPU utilization as in 6(a), the consumption is concentration at 80%. Such increase is due to the operations involved while decrypting the data before mining. Similarly, in the case of fog nodes, we observe an increase in CPU utilization. Since RSA is relatively lightweight, we observe a minor increase. However, in the case of AES-256, the devices perform rounds of operations while swapping rows and columns of the plain text according to the key length. Typically, AES-256 needs $12-14$ rounds for encryption and decryption. Thus, we observe a jump in the CPU utilization in all the devices. Although we observe increased utilization values, the maximum in the case of fog nodes is 62%. The fog nodes still have the potential to perform additional tasks. We observe that although there exists a tradeoff for enhancing security, the encryption schemes do not hamper the fog nodes' performance. Table II depicts the observed minimum, maximum, and the average CPU usage.



(a) SDN controllers (b) Miners

Fig. 7. Delays endured by resource-constrained SDN controllers and miners while testing pBC-enabled SDN

D. Delay

We perform an analysis of the delays demonstrated by 1) the resource-constrained SDN controllers in sending new flow rules into the pBC (transactions) and 2) the miners for mining (verifying) the new flow rules and adding them as new blocks into the pBC. Since we use heterogeneous types of devices in our experiments, we demonstrate the delays in the form of histograms in Fig. 7.

We observe in Fig. 7(a) that the delays for sending new flow rules are minuscule (in the range of milliseconds). The low values of delays is because of the lightweight flow rule setting routines. Such a range of delays illustrates the feasibility of encrypting the flow rules and its insertion into the pBC network by resource-constrained fog nodes as SDN

controllers. However, we observe relatively higher latencies in the range of 0.10 ms for some devices. We attribute these higher ranges of delays to inter-application switching within the devices as the fog nodes simultaneously perform other operations in addition to our proposed routine.

We present the delay in mining the flow rules in Fig. 7(b). We observe that the miners have most of their delays concentrated to 1 second and relatively sparse up to 2 seconds. We attribute the varying time densities to the variation in the flow rules as well as to the device executing other operations. In the miner, we also observe a delay of more than 3 seconds in a few instances due to the inter-application switching.

It may be noted that the delays are inclusive of the routines involved in encryption and decryption of the flow rules.

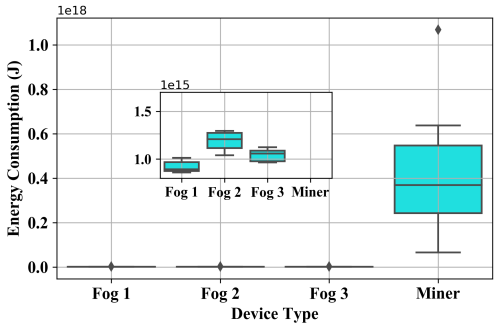


Fig. 8. Energy consumption by devices in the proposed SDN-pBC system

E. Energy Consumption

Inspired from the works of [12], we find the energy consumption in case of each device type as $E = \kappa(c_f)^2 n_{cycles}$, where κ is a constant for energy consumption depending on the material used, c_f is the device's CPU cycles per second, and n_{cycles} is the number of cycles performed during an operation. The use of comparatively powerful processors for mining leads to higher consumption of energy in Fig. 8. On the other hand, the resource-constrained SDN controllers/fog nodes consume a minimal amount of energy and are comparable to one other. Note that conventional SDN deployments use controllers with configurations similar to that of our miners. Towards this, we observe that using resource-constrained fog nodes as SDN controllers and its fusion with pBC offers savings in energy by more than 90%.

V. Conclusion

In this work, we implemented a lightweight SDN system on a network of resource-constrained devices in the fog layer. To overcome the security issues in the centralized architecture of SDN, we implemented a pBC network among the devices to store the flow rules. Additionally, for dealing with undesired access of pBC data by users having the same genesis file, we encrypted the flow rules before inserting them into the blocks. Our proposed implementation of secured SDN readily decentralized the conventional SDN architecture, and its deployment in the fog layer significantly reduced the latency and energy consumptions. We conclude that it is possible to deploy our proposed solution into the current network infrastructure for segregating the data and control planes in resource-constrained devices while reducing time and energy.

In the future, we plan to extend our work by designing routines for autonomous identification of malicious flow rules in real-time. We also plan to study the nature and size of the blocks as the size of the network increases, along with the possibility of discarding the older blocks with time.

VI. Acknowledgement

The authors gratefully acknowledge SensorDrops Networks Pvt Ltd for lending their IoT platform for implementing the work reported in this paper. Additionally, the first author acknowledges the funding support received from INAE (Sanction letter no. INAE/121/AKF, Dt. 13-02-2019), and SERB/IMPRINT-II (Sanction letter no SERB/F/12680/2018-2019;IMP/2018/000451, Dt. 25-03-2019) for executing parts of this project.

References

[1] Y. Liu, Y. Kuang, Y. Xiao, and G. Xu, "SDN-Based Data Transfer Security for Internet of Things," *IEEE Internet of Things Journal*, vol. 5, no. 1, pp. 257–268, Feb. 2018.

[2] A. J. Kadhim and S. A. Hosseini Seno, "Maximizing the Utilization of Fog Computing in Internet of Vehicle Using SDN," *IEEE Communications Letters*, vol. 23, no. 1, pp. 140–143, Jan. 2019.

[3] L. Fawcett, S. Scott-Hayward, M. Broadbent, A. Wright, and N. Race, "Tennison: A Distributed SDN Framework for Scalable Network Security," *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 12, pp. 2805–2818, Dec. 2018.

[4] H. Zhou, C. Wu, C. Yang, P. Wang, Q. Yang, Z. Lu, and Q. Cheng, "SDN-RDCD: A Real-Time and Reliable Method for Detecting Compromised SDN Devices," *IEEE/ACM Transactions on Networking*, vol. 26, no. 5, pp. 2048–2061, Oct. 2018.

[5] V. Varadharajan, K. Karmakar, U. Tupakula, and M. Hitchens, "A Policy-Based Security Architecture for Software-Defined Networks," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 4, pp. 897–912, Apr. 2019.

[6] Z. Abou El Houda, A. S. Hafid, and L. Khoukhi, "Cochain-SC: An Intra- and Inter-Domain Ddos Mitigation Scheme Based on Blockchain Using SDN and Smart Contract," *IEEE Access*, vol. 7, pp. 98 893–98 907, Jul. 2019.

[7] S. R. Basnet and S. Shakya, "BSS: Blockchain Security Over Software Defined Network," in *Proceedings of International Conference on Computing, Communication and Automation (ICCCA)*, May 2017, pp. 720–725.

[8] M. Boussard, S. Papillon, P. Peloso, M. Signorini, and E. Waisbard, "STewARD:SDN and Blockchain-Based Trust Evaluation for Automated Risk management on IoT Devices," in *Proceedings of IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, Apr. 2019, pp. 841–846.

[9] L. Xie, Y. Ding, H. Yang, and X. Wang, "Blockchain-Based Secure and Trustworthy Internet of Things in SDN-Enabled 5G-VANETs," *IEEE Access*, vol. 7, pp. 56 656–56 666, Apr. 2019.

[10] P. K. Sharma, M. Chen, and J. H. Park, "A Software Defined Fog Node Based Distributed Blockchain Cloud Architecture for IoT," *IEEE Access*, vol. 6, pp. 115–124, Sep. 2018.

[11] A. Arins, "Blockchain based Inter-domain Latency Aware Routing Proposal in Software Defined Network," in *Proceedings of IEEE 6th Workshop on Advances in Information, Electronic and Electrical Engineering (AIEEE)*, Nov. 2018, pp. 1–2.

[12] T. X. Tran and D. Pompili, "Joint Task Offloading and Resource Allocation for Multi-Server Mobile-Edge Computing Networks," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 1, pp. 856–868, Jan. 2019.