BUAV: A Blockchain Based Secure UAV-Assisted Data Acquisition Scheme in Internet of Things

Anik Islam and Soo Young Shin

Abstract: **Internet of things (IoT), mobile edge computing (MEC), and unmanned aerial vehicle (UAV) have attracted significant attention in both industry and academic research. By consolidating these technologies, IoT can be facilitated with improved connectivity, better data transmission, energy saving, and other advantages. However, the communication between these entities is subject to potential cyber threats. In addition, the integrity of the data must be maintained after storing into local storage. Blockchain is a data structure that supports features like pseudonymity, data integrity etc. This paper represents a blockchain based data acquisition process in which information is gathered from IoTs using UAV as a relay and is securely kept in blockchain at MEC server. In the proposed scheme, data are encrypted prior to transfer to MEC server with the assistance of a UAV. Upon receiving the data, MEC server validates the data and the identity of the sender. Successful validation is followed by stocking of the data into blockchain, subsequent to obtaining consent from the validators. Security analysis is conducted in order to show the feasibility of the proposed secure scheme. Finally, the performance of the proposed scheme is analyzed via simulation and implementation.**

Index Terms: **Blockchain, Internet of things, MEC, security, unmanned aerial vehicle.**

I. INTRODUCTION

INTERNET of things (IoT) has generated considerable attention not only in industrial fields but also in academic research. IoT is a network of objects, equipped with sensors, actuators, software, etc., that are connected via Internet. Moreover, these objects can collect information from their surroundings and can interchange these data among each other [1]. Based on the promising possibilities attributed to IoT, an estimation of appending 30 billion devices in the network by 2020 has been made [2]. However, IoT is not only confined to sensing and information interchange. Currently, IoT can make spontaneous decisions based on the sensed data. IoT has improved the quality of our lives by contributing to the development of several different sectors including healthcare [3]–[8], agriculture [9]–[12], smart homes [13]–[15], smart grids [16]–[20], and others [21].

Unmanned aerial vehicles (UAVs) is an example of an emerging technology that achieved widespread popularity owing to its potential for diverse applications. Initially, UAVs were confined to applications within the military. However, the use of these devices has subsequently been extended to civil applications [22]. The advantages of UAVs is their ease of deployment, low maintenance and acquisition cost, and the ability to access almost any areas [23]. Moreover, some UAVs contain sensors, actuators, high computational power, etc., which not only can accumulate data along with spatial information but also can process it, reach decisions and act according to the decision [24]–[26]. UAVs can be posted to locations that are difficult to access in order to provide assistance to IoTs. Occasionally, UAVs can operate as a server while providing services to IoTs. In addition, UAVs can spread network connection to IoTs for the non-line-of-sight zones. Moreover, UAVs can save the energy of devices by assisting in forwarding the data and can quickly be replaced in the occurrence of a fault, while providing assistance to IoTs. UAVs can also cover a large area which can help to optimize the deployment cost.

Mobile edge computing (MEC) is another promising initiative that draws cloud computing facilities closer to user proximity at the edge of the mobile network and provides services by utilizing radio access networks (RANs) [27]. The purpose of MEC is to provide ultra-low latency, high bandwidth, and real-time services to mobile subscribers [28]. However, collecting data from IoTs via the cloud creates latency issues, along with challenges related to low throughput and single-node failure [29]. MEC is a potential candidate for alleviating the aforementioned issues by providing real-time data acquisition services from IoT devices via the utilization of UAVs. However, the data acquisition process is prone to threats like the man-in-the-middle, spoofing etc. Moreover, the collected data may experience unauthorised modifications in the storage, which may lead to uncertainty regarding the integrity of the data. Therefore, a secure scheme for data acquisition via UAV is required.

Among the existing studies, in [30], a UAV based fog computing platform that aids IoT devices has been proposed. In addition, an architecture in which information is obtained from IoT using UAV (sometimes UAV to UAV) and stored in the cloud has been appended. In [31], UAV served as a fog node so that it could provide real-time assistance to IoT devices. However, this may consume a significant amount of energy and may reduce the flight time of UAV. In [31], an approach was proposed in which UAVs are deployed based on client positions so that the client can perform queries and acquire data from IoTs with less

Manuscript received February 1, 2019; approved for publication by Naït-Abdesselam Farid, Division SI.

This research was supported by the MSIT (Ministry of Science, ICT), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2019-2014-1-00639) supervised by the IITP (Institute for Information & communications Technology Planning & Evaluation).

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. 2019R1A2C1089542).

This work was supported by Priority Research Centers Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (2018R1A6A1A03024003).

A. Islam and S. Y. Shin are with the WENS Laboratory, Department of IT Convergence Engineering, Kumoh National Institute of Technology, Gumi 39177, South Korea, (email: {anik.islam, wdragon}@kumoh.ac.kr).

S.Y. Shin is the corresponding author.

Digital object Identifier: 10.1109/JCN.2019.000050

delay. In [31], UAV served as a gateway for data acquisition and UAVs were controlled using UAVs control server. In that proposed scheme, data are stored in the global server that serves the client by obtaining data from UAVs. In [32], a UAV-assisted data collection scheme for a wireless sensor network (WSN) was presented. To perform data acquisition, they first divided that region into multiple areas. Subsequently, they deployed UAVs based on the plan established while dividing the region. In [33], a smart farming scheme was presented in which UAV was utilized to communicate with ground sensors and aid these sensors. In [33], a testbed considering IEEE 802.15.4-based communication between UAV and ground sensors was implemented. A prototype was presented in wherein a UAV served as a base station (BS) and performed wireless power transfer and communications with sensors simultaneously, in [34]. In [35], a RESTful approach was presented in which UAV equipped with IoT relays data to cloud services along with a vertical handover mechanism in which UAV can shift between different modes of communication such as Wi-Fi and satellite for beyond-line-of-sight communication issues, to increase reliability. However, the aforementioned schemes did not consider any security issues while deploying UAVs. In addition, they did not add information regarding data management after collecting from IoTs, which may create an issue concerning the integrity of stored data.

Blockchain is a distributed ledger technology that is shared among peers in a network where the peers hold the same copies of the ledger [36]. Blockchain was first introduced by Satoshi Nakamoto in a white paper, called "Bitcoin: A peer-to-peer electronic cash system", in 2008 [37]. Initially, blockchain was confined to the documentation of financial transactions. However, owing to its promising functionality (e.g., immutability, smart contract, etc.), this technology is currently in use in the recording of medical data, tracking of products in the supply chain, etc. [38]. In blockchain, data are deposited in a block and each block has a unique hash, which is generated based on the content of the block [39]. Each block points to the previous block's hash; thus, blocks stay in the chain, and that is why it is called blockchain. Blockchain adopts asymmetric encryption to maintain security [40]. Each user holds a private key and a public key that is generated from the private key. Every user uses this public key as an identity in the network, and thus, the identity of the user remains obscured [41]. However, blockchain also utilizes a Merkle tree [39]. A Merkle tree is a structure that is generated from the hash of the data. Any alteration in the data causes a change in the Merkle tree. By utilizing the Merkle tree, blockchain preserves the integrity of data. However, to add a block to the chain, every node, called a miner or validator, must agree on the validity of the block [42]. These characteristics can be possible solutions against the aforementioned issues (i.e., cyber threats and data integrity) in data acquisition from IoT using a UAV.

A blockchain based data acquisition scheme is proposed in which data are gathered from IoTs via UAV and kept securely in blockchain at MEC, to alleviate the aforementioned issues. The major contributions of this paper are summarized as follows.

- A scheme is proposed to perform data acquisition from IoT devices.
- A discussion on security based on different vulnerabilities is provided.
- The impact of η-hash bloom filter[1] is simulated both in MATLAB for edge server and Python for UAV.
- We have implemented the proposed scheme and the performance is examined by throughput, processing time, energy consumption, and latency.
- We have implemented blockchain using ethereum and the performance is investigated in terms of write, read, latency, and delay.

The rest of the sections are arranged as follows: Section II represents the data acquisition scheme. In Section III, the process of data acquisition is depicted. Section IV illustrates a security analysis of the proposed scheme. The simulation and experimental setup are discussed in Section V. Finally, Section VI summarizes the mains conclusions and examines future research directions.

II. PROPOSED SCHEME

We devise a blockchain based data acquisition scheme (termed as "BUAV") that facilitates the collection of data from IoTs using a UAV, as presented in Fig. 1.

A. Basic Idea

BUAV concentrates on data acquisition from IoT devices using UAVs as well as the deposition of the collected data in a secure way to avoid compromising data integrity and BUAV also ensures secure data transmission. It was assumed that all valid IoT devices and UAVs are registered in the scheme. In BUAV, IoT devices intend to transmit data to the MEC. Before transferring data, IoT device performs encryption. This data is then transmitted to UAVs along with their identity. Upon receiving the data, UAVs perform decryption and check the validity of the identity of the sender. For validating devices, η-hash bloom filter is adopted in BUAV. However, if the identity is not valid, then UAVs discard the data and append that invalid identity to a vulnerable list. If the invalid identity continues to transmit data, then UAVs block the communication channel for that identity for a specific period. In BUAV, UAV decrypts and validates the data in order to prevent malicious devices at an early stage. If a malicious device continuously sends invalid data and UAV just forwards it to the server without any validation then MECS gets a lot of traffic containing invalid data. But, MECS cannot block the sender because data are coming through the UAV (if MECS wants to block the sender then it has to block UAV) and only UAV can identify the sender. That is why UAV decrypts the data and prevents malicious devices from reaching the server. However, for a valid identity, UAVs forward the data to the MEC server. When MEC server receives the data, MEC server verifies the validity of the sender. In case of a valid identity, MEC stores the data in blockchain. When data is added to the network, block sealing process is initiated. Upon receiving the acknowledgement from other validators, data is added to the network including a private cloud.

[1]Bloom filter is a data structure to check the existence of an item in a set consuming less memory [43].

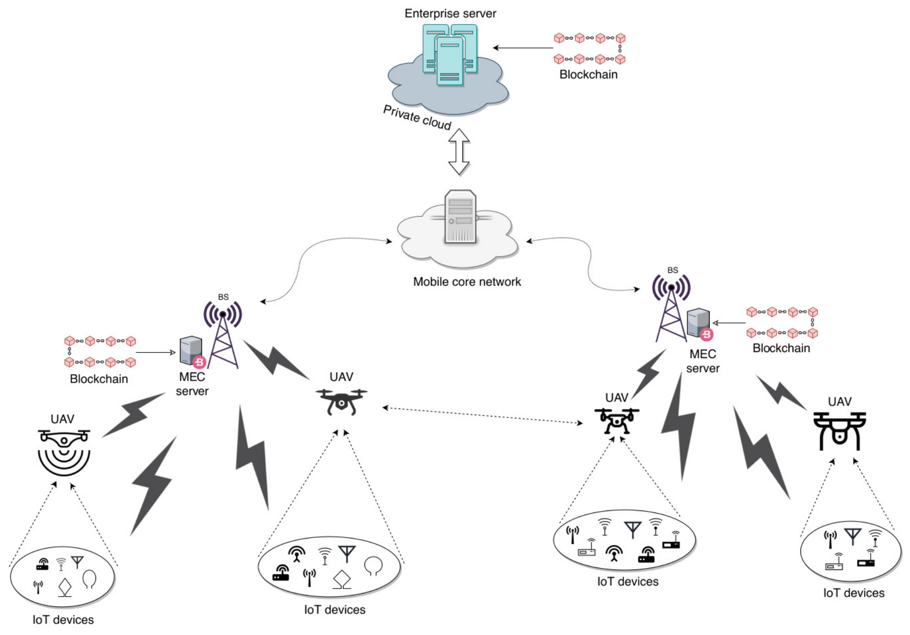


Fig. 1. Data acquisition from IoTs to MEC server using a UAV.

B. Entities

BUAV consists of four entities: IoT device (IoTD), UAV, MEC server (MECS), and private cloud (PC). The entities of BUAV are depicted below:

- IoTD: IoT device collects data and transmits it to the MECS. This device consists of sensors and actuators and it can both sense the environment and perform tasks according to the instruction.
- UAV: UAV receives data from IoTD and relays these data to the server. UAV also authorizes requests prior to forwarding to the server. Every UAV maintains a list of IoTDs and other UAVs, which it utilizes to perform the authentication process using η-hash bloom filters before forwarding data to MECS.
- MECS: MEC server collects data from IoTD via UAV that are subsequently stored in blockchain. MEC not only maintains basic information (i.e., device name, mac address, device type, etc.) but also spatial information (i.e., latitude, longitude, etc.) that assists MEC server in maintaining the data acquisition process. MEC server also validates UAV and IoTD before including data into blockchain.
- PC: The private cloud is composed of private servers; the PC is linked with MECS via backhaul networks. Since the PC is also serving as a client of blockchain, it also gets the update when new data is added to MECS.

C. Notations

A list of common notations including their description is compiled in Table1.

Table 1. Notations with description.

Notation	Description
σ, Υ, $\mho$	IoT Device, MEC server, UAV.
$\hat{S}_{\hbar}$, $\wp$, ϱ	Random salt hash, Private key, Public key.
$\psi(.)$, $\Gamma(.)$, κ_s	Hash function, Key generator, Key size.
$\beta F(.)$, $\Theta(.)$	Bloom filter, Signature generator.
$\xi(.)$, $\zeta(.)$	Encryption function, Decryption function.
$\omega(.)$	Signature verification function.
$\partial\check{V}(.)$	Validate blocks proposed by a validator.
V_L, B_L	Vulnerable list, Blocked list.
$\tau_{\hat{t}t}^{\{a\to b\}}$	Transmission delay from a to b.

III. COMMUNICATIONS IN THE PROPOSED SCHEME

A. Registration

We assumed that valid IoTDs and UAVs are registered users. Prior to registration in the system, UAV or IoTD generates a unique hash based on the mac address of the device δ_{mac}, timestamp τ_c, and random salt hash $\hat{S}_{\hbar}$. Let h be the hash,

$$h = \psi(\delta_{mac}, \tau_c, \hat{S}_{\hbar}) \mid \psi : \{0,1\}^* \to \{0,1\}^l, \tag{1}$$

where h is used to generate private key $\wp_\kappa$ and subsequently, a public key ϱ_κ is created from $\wp_\kappa$. Let $\hat{E}$ is a point on the elliptic curve containing (x, y),

$$\begin{aligned} \wp_\kappa &= \Gamma(h) \mid \Gamma : \{0,1\}^* \to \{0,1\}^{\kappa_s}, \\ \hat{\varrho}_\kappa &= \wp_\kappa \otimes (\hat{E}_x, \hat{E}_y). \end{aligned} \tag{2}$$

After generating keys, IoTD transmits ϱ_κ to MECS along with device information (i.e., δ_{mac}, latitude, longitude, etc.). $\hat{\varrho}_\kappa$ is used as an identity of a IoTD. MECS then stores these information in a smart contract $\mathtt{C}_\delta$ of blockchain.

B. Data Generation

BUAV utilizes η-hash bloom filters ($\eta = 1, 2, \cdots$) to validate IoTDs before forwarding data to MECS. MECS fetches the device list from $\mathtt{C}_\delta$. Let $\bar{\eth}$ be the list,

$$\overline{\mathfrak{d}} = \bigcup \mathfrak{C}_\delta(\mathfrak{d}^{\mathfrak{k}}), \mathfrak{k} \leq \mathfrak{l}_{\mathfrak{r}}.$$

Here, $\mathfrak{k}$ is the location containing $\langle lat, lon\rangle$ and $\mathfrak{r}$ is radius of the location of $\mathfrak{l}$. Prior to validating IoTD, UAV requires filter table $\overline{\Im} \mid \Im \in \{0,1\}$. When a UAV registers or connect with MECS, MECS generates $\overline{\Im}$ and shares it with UAV. Let $\overline{\Im}$ is the generated table,

$$\overline{\Im} = \bigcup_{i=1}^{n} \bigcap_{j=1}^{\eta} \beta\digamma_j(\mathfrak{d}_{\mathfrak{i}}) \\ \mid \beta\digamma : \{0,1\}^* \to \nu \cap \Im_\nu \in \{0,1\} \cap k \in \{0,1,2\}, \tag{3}$$

where n is the total number of devices. η is the number of hashes that are employed in the process. Bloom filter has a false positive issue that can be reduced by controlling the number of hash functions and data size [43]. However, there is a limit to the use of hash functions in the bloom filter. Let n is number of $\mathfrak{d}$, and p is the acceptance of false positive rate [44],

$$m = \left\lceil \frac{-n \times \ln(p)}{\ln(2)^2} \right\rceil \mid p \in (0,1], \eta = \left\lfloor \frac{m}{n} \times \ln(2) \right\rfloor, \tag{4}$$

where η is the maximum number of hash function that is suitable for $\beta\digamma$.

C. Data Transmission

In BUAV, a IoTD σ starts transmitting a data α to MECS Υ via UAV $\mho$. Prior to transmitting θ, σ creates a signature ε from α by employing the private key $\wp^\sigma_\kappa$ of σ,

$$\varepsilon = \Theta_{\wp^\sigma_\kappa}(\psi(\alpha)) \mid \psi : \{0,1\}^* \to \{0,1\}^l, \\ \Theta : \{0,1\}^* \to \{0,1\}^l. \tag{5}$$

After generating ε, σ encrypts α along with ε and $\hat{\varrho}^\sigma_\kappa$ by using the public key $\hat{\varrho}^\mho_\kappa$ of Ψ. Let β is the encrypted data,

$$\beta = \xi_{\hat{\varrho}^\mho_\kappa}(\alpha, \varepsilon, \hat{\varrho}^\sigma_\kappa). \tag{6}$$

Subsequently, σ transmits β to $\mho$. Let τ_p is the total processing time for preparing data at σ,

$$\tau^\sigma_p = \tau^\sigma_\varepsilon + \tau^\sigma_\beta. \tag{7}$$

Here, τ^σ_ε and τ^σ_β is the processing time for ε and β, respectively. Let $T^{\{\sigma\to\mho\}}$ be the time for the transmission of α from σ to $\mho$,

$$T^{\{\sigma\to\mho\}} = \tau^\sigma_p + \tau^{\{\sigma\to\mho\}}_d, \tag{8}$$

where $\tau^{\{\sigma\to\mho\}}_d$ is the transmission delay from σ to $\mho$. Upon receiving β, $\mho$ decrypts β by employing the private key $\wp^\mho_\kappa$ of $\mho$, as shown in Algorithm 1. Let $\breve{d}$ is the decrypted data,

$$\breve{d} = \zeta_{\wp^\mho_\kappa}(\beta). \tag{9}$$

After obtaining $\breve{d}$, $\mho$ first checks B_L and subsequently, it checks the validity of $\hat{\varrho}^\sigma_\kappa$ by utilizing $\beta\digamma$. Let $\ddot{f}$ is the validity result,

Algorithm 1 data processing in UAV.

```
Input: Encrypted received data.
Output: Validity of IoT device.
1:  $\breve{d} \leftarrow \zeta_{\wp^\mho_\kappa}(\beta)$.
2:  if $\breve{d}_{\hat{\varrho}^\sigma_\kappa} \notin B_L$ then
3:      $\ddot{f} \leftarrow \bigcap_{j=1}^{\eta} \beta\digamma_j(\breve{d}_{\hat{\varrho}^\sigma_\kappa})$.
4:      if $\ddot{f} == 1$ then
5:          $\gamma \leftarrow \xi_{\hat{\varrho}^\Upsilon_\kappa}(\breve{d}_\alpha, \breve{d}_\varepsilon, \breve{d}_{\hat{\varrho}^\sigma_\kappa}, \breve{d}_{\hat{\varrho}^\mho_\kappa})$.
6:          $\mho \xrightarrow{forwards} \Upsilon$.
7:      else
8:          $\breve{d}_{\hat{\varrho}^\sigma_\kappa} \xrightarrow{include} V_L$.
9:          if $Count(V_L^{\hat{\varrho}^\sigma_\kappa}) \geq \Re^\mho_n$ then
10:             $\breve{d}_{\hat{\varrho}^\sigma_\kappa} \xrightarrow{include} B_L, V_L^{\hat{\varrho}^\sigma_\kappa} \to \varnothing$.
11:         end if
12:     end if
13: end if
```

$$\ddot{f} = \bigcap_{j=1}^{\eta} \beta\digamma_j(\hat{\varrho}^\sigma_\kappa) \text{ where } \ddot{f} \in \{0,1\}. \tag{10}$$

If $\ddot{f}$ is 0 then $\hat{\varrho}^\sigma_\kappa$ is not valid. $\mho$ discards $\breve{d}$ instantly and places $\hat{\varrho}^\sigma_\kappa$ under V_L. If $\hat{\varrho}^\sigma_\kappa$ continues to transmit invalid requests then after a threshold number of requests $\Re^\mho_n$, $\mho$ adds $\hat{\varrho}^\sigma_\kappa$ to B_L. However, for the valid $\hat{\varrho}^\sigma_\kappa$, $\mho$ encrypts the data along with $\hat{\varrho}^\mho_\kappa$, α, $\hat{\varrho}^\sigma_\kappa$, and ε by using the public key of Υ. Let γ is the encrypted data,

$$\gamma = \xi_{\hat{\varrho}^\Upsilon_\kappa}(\alpha, \varepsilon, \hat{\varrho}^\sigma_\kappa, \hat{\varrho}^\mho_\kappa). \tag{11}$$

Subsequently, $\mho$ forwards γ to Υ. Let τ_p is the total processing time for preparing data at $\mho$,

$$\tau^\mho_p = \tau^\mho_{\breve{d}} + \tau^\mho_{\ddot{f}} + \tau^\mho_\gamma, \tag{12}$$

where $\tau^\mho_{\breve{d}}$, $\tau^\mho_{\ddot{f}}$, and $\tau^\mho_\gamma$ is the processing time for $\breve{d}$, $\ddot{f}$ and γ, respectively. Let $T^{\{\mho\to\Upsilon\}}$ is the time to transmit γ from $\mho$ to Υ,

$$T^{\{\mho\to\Upsilon\}} = \tau^\mho_p + \tau^{\{\mho\to\Upsilon\}}_d, \tag{13}$$

where $\tau^{\{\mho\to\Upsilon\}}_d$ is the transmission delay from $\mho$ to Υ.

When Υ receives γ, Υ decrypts γ by employing the private key $\wp^\Upsilon_\kappa$ of Υ, as shown in Algorithm 2. Let $\tilde{d}$ is the decrypted data,

$$\tilde{d} = \zeta_{\wp^\Upsilon_\kappa}(\gamma). \tag{14}$$

After getting $\tilde{d}$, Υ first checks the B_L and subsequently, Υ checks the validity of $\hat{\varrho}^\mho_\kappa$ and $\hat{\varrho}^\sigma_\kappa$ by using $\beta\digamma$. Let $\check{f}$ is the validity result,

$$\check{f} = \bigcap_{j=1}^{\eta} \beta\digamma_j(\hat{\varrho}^\mho_\kappa) \cap \beta\digamma_j(\hat{\varrho}^\sigma_\kappa), \\ \text{where } \check{f} \in \{0,1\}. \tag{15}$$

If $\check{f}$ is 0 then either $\hat{\varrho}^\sigma_\kappa$ or $\hat{\varrho}^\sigma_\kappa$ is not valid. Υ discards $\tilde{d}$ instantly and put $\hat{\varrho}^\mho_\kappa$ in V_L. If $\hat{\varrho}^\mho_\kappa$ continues to transmit invalid

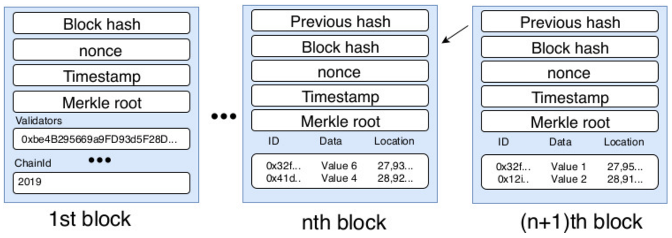


Fig. 2. Data management in blockchain.

requests then after a threshold number of requests $\ddot{\Re}_n^{\Upsilon}$, Υ puts $\hat{\varrho}_\kappa^{\mho}$ in B_L and the channel is blocked for a threshold amount of time $\ddot{\Re}_\tau^{\Upsilon}$. However, for the valid $\hat{\varrho}_\kappa^{\mho}$, Υ continues to process $\tilde{d}$. Before adding to storage, Υ checks the integrity of α by verifying the signature of σ. Let $\ddot{v}$ is verification result,

$$\ddot{v} = \omega_{\hat{\varrho}_\kappa^\sigma}(\alpha, \varepsilon) \text{ where } \ddot{v} \in \{true, false\}. \tag{16}$$

If $\ddot{v}$ is true, then Υ transmits α for validating and adding in the blockchain. Let τ_p is the total processing time for validating data at Υ,

$$\tau_p^{\Upsilon} = \tau_{\tilde{d}}^{\Upsilon} + \tau_{\check{f}}^{\Upsilon} + \tau_{\ddot{v}}^{\Upsilon}, \tag{17}$$

where $\tau_{\tilde{d}}^{\mho}$, $\tau_{\check{f}}^{\mho}$, and $\tau_{\ddot{v}}^{\mho}$ are the processing time for $\tilde{d}$, $\check{f}$, and $\ddot{v}$, respectively. Let $T^{\{\sigma\to\mho\to\Upsilon\}}$ is the total time to transmit and process α successfully from σ to Υ via $\mho$ including the validation process. By adding (8) and (13), it can be written that,

$$T^{\{\sigma\to\mho\to\Upsilon\}} = T^{\{\sigma\to\mho\}} + T^{\{\mho\to\Upsilon\}}. \tag{18}$$

When α is valid, Υ starts to add α to the blockchain $\mathbb{B}$. We considered a private blockchain network in which the identity of validators $\ddot{\nu}$ (those that create blocks) is known to everyone. In BUAV, each block holds a $\mathcal{T}'$ number of transactions, as illustrated in Fig. 2. In Fig. 2, the first block is the genesis block that contains the list of validators, chainId (i.e., network id), etc. After that, each blockchain contains the data of IoT devices. However, a validator proposes a block $\varkappa$ and other validators validate the block. This process is performed in a round-robin algorithm. Let $\hat{f}$ contains the validation result,

$$\hat{f} = \bigcap_{z=1}^{\ddot{m}} \partial\check{V}_z(\varkappa_\alpha), \tag{19}$$

where $\ddot{m}$ is the total number of validators. Once, every validator validates the block, the block is appended in the chain.

D. UAV Requisition

If σ disconnects with $\mho$ and cannot reconnect with $\mho$, σ transmits request $\ddot{r}$ to Υ to send information regarding new $\mho$ in order to facilitate the transfer of data. There exists a direct communication link between IoTD and MECS. However, IoTD utilizes that channel only when IoTD cannot locate any UAV nearby. Moreover, transmitting data through that channel drains more energy than the communication channel that exists between IoT and UAV. Therefore, continuous transmission can decrease the lifetime of the battery of IoTD. In order to increase the lifetime,

Algorithm 2 data validation in MECS.

```
Input: Encrypted received data.
Output: Validity of IoT device and UAV.
1:  $\tilde{d} \leftarrow \zeta_{\wp_\kappa^\Upsilon}(\gamma)$.
2:  if $\tilde{d}_{\hat{\varrho}_\kappa^\mho} \notin B_L$ then
3:      $\check{f} \leftarrow \bigcap_{j=1}^{\eta} \beta F_j(\tilde{d}_{\hat{\varrho}_\kappa^\mho}) \cap \beta F_j(\tilde{d}_{\hat{\varrho}_\kappa^\sigma})$.
4:      if $\check{f} == 1 \land \omega_{\tilde{d}_{\hat{\varrho}_\kappa^\sigma}}(\tilde{d}_\alpha, \tilde{d}_\varepsilon) == true$ then
5:          $\tilde{d}_\alpha \xrightarrow{inserts} \mathbb{B}$.
6:      else
7:          $\tilde{d}_{\hat{\varrho}_\kappa^\mho} \xrightarrow{include} V_L$.
8:          if $Count(V_L^{\hat{\varrho}_\kappa^\mho}) \geq \ddot{\Re}_n^\Upsilon$ then
9:              $\tilde{d}_{\hat{\varrho}_\kappa^\mho} \xrightarrow{include} B_L, V_L^{\hat{\varrho}_\kappa^\mho} \to \varnothing$.
10:         end if
11:     end if
12: end if
```

BUAV utilizes UAV for relaying data. Prior to sending $\ddot{r}$, σ creates a signature $\ddot{\varepsilon}$ from $\ddot{r}$ by employing the private key $\wp_\kappa^\sigma$ of σ,

$$\begin{gathered} \ddot{\varepsilon} = \Theta_{\wp_\kappa^\sigma}(\psi(\ddot{r})) \mid \psi : \{0,1\}^* \to \{0,1\}^l, \\ \Theta : \{0,1\}^* \to \{0,1\}^l. \end{gathered} \tag{20}$$

Subsequently, σ encrypts $\ddot{r}$ along with $\hat{\varrho}_\kappa^\sigma$ and $\ddot{\varepsilon}$ by employing $\hat{\varrho}_\kappa^\Upsilon$. Let $\hat{\beta}$ is the encrypted data,

$$\hat{\beta} = \xi_{\hat{\varrho}_\kappa^\Upsilon}(\ddot{r}, \ddot{\varepsilon}, \hat{\varrho}_\kappa^\sigma). \tag{21}$$

Upon receiving $\hat{\beta}$, Υ performs a decryption by using $\wp_\kappa^\Upsilon$, as shown in Algorithm 3. Let $\acute{d}$ is the decrypted data,

$$\acute{d} = \zeta_{\wp_\kappa^\Upsilon}(\hat{\beta}). \tag{22}$$

After obtaining $\acute{d}$, Υ checks the validity of $\hat{\varrho}_\kappa^\sigma$ by utilizing βF. Let $\grave{f}$ is the validity result,

$$\grave{f} = \bigcap_{j=1}^{\eta} \beta F_j(\hat{\varrho}_\kappa^\sigma) \text{ where } \grave{f} \in \{0,1\}. \tag{23}$$

If $\grave{f}$ returns a value of 1, then Υ continues to process the request. Prior to searching $\mho$ for σ, Υ verifies the signature which is transmitted within the request. Let $\check{v}$ is verification result,

$$\check{v} = \omega_{\hat{\varrho}_\kappa^\sigma}(\ddot{r}, \ddot{\varepsilon}) \text{ where } \check{v} \in \{true, false\}. \tag{24}$$

If $\check{v}$ contains $true$ then Υ retrieves the spatial information of σ. Subsequently, Υ retrieves all UAVs from the list of devices $\ddot{\sigma}$. Let $\overline{\mho}$ is the collection of UAVs,

$$\overline{\mho} = \bigcup_{k==1} (\ddot{\sigma}^k). \tag{25}$$

After retrieving $\overline{\mho}$, Υ calculates the geo distance between σ and $\overline{\mho}$. Subsequently, Υ sorts the list in the ascending order based on the geo distance and picks the top threshold number $\Re_{gd}^{\mho}$ of $\mho$. Before transmitting the results, Υ first creates a signature ε_Υ by employing $\wp_\kappa^\Upsilon$,

Algorithm 3 UAV requisition procces in MECS.

```
Input: Encrypted received request.
Output: List of the nearest UAV.
1:  $\acute{d} \leftarrow \zeta_{\wp_\kappa^\Upsilon}(\hat{\beta})$.
2:  if $\acute{d}_{\hat{\varrho}_\kappa^\sigma} \notin B_L$ then
3:      $\grave{f} \leftarrow \bigcap_{j=1}^{\eta} \beta F_j(\acute{d}_{\hat{\varrho}_\kappa^\sigma})$.
4:      if $\grave{f} == 1 \wedge \omega_{\acute{d}_{\hat{\varrho}_\kappa^\sigma}}(\acute{d}_{\ddot{r}}, \acute{d}_{\ddot{\varepsilon}}) == true$ then
5:          $\overline{\mho} \leftarrow \bigcup_{k==1}(\ddot{\sigma}^k), \overline{\ddot{u}} \rightarrow \varnothing, \ddot{\sigma} \xleftarrow{retrieve} \acute{d}_{\hat{\varrho}_\kappa^\sigma}$.
6:          while $\ddot{u} \in \overline{\mho}$ do
7:              $\breve{a} \leftarrow 0.5 - \frac{\cos((\ddot{u}_{lat} - \ddot{\sigma}_{lat}) \times \frac{\pi}{180})}{2} + \cos(\ddot{\sigma}_{lat} * \frac{\pi}{180}) \times \cos(\ddot{u}_{lat} * \frac{\pi}{180}) \times \frac{1 - \cos((\ddot{u}_{lon} - \ddot{\sigma}_{lon}) \times \frac{\pi}{180})}{2}$.
8:              $\ddot{u}_{dist} \leftarrow 2 \times 6371 \times \arcsin(\sqrt{\breve{a}})$.
9:          end while
10:         $\overline{\mho} \leftarrow sort(\overline{\mho})_{prop=dist}^{ord=asc}, \overline{\mho}_{gd} \leftarrow \bigcup_{i=1}^{\Re_{gd}^{\mho}}(\overline{\mho}_i)$.
11:         $\varepsilon_\Upsilon \leftarrow \Theta_{\wp_\kappa^\Upsilon}(\psi(\overline{\mho}_{gd})), \hat{\gamma} \leftarrow \xi_{\hat{\varrho}_\kappa^\sigma}(\overline{\mho}_{gd}, \varepsilon_\Upsilon)$.
12:     end if
13: end if
```

$$\varepsilon_\Upsilon = \Theta_{\wp_\kappa^\Upsilon}(\psi(\overline{\mho}_{gd})) \mid \psi : \{0,1\}^* \rightarrow \{0,1\}^l, \\ \Theta : \{0,1\}^* \rightarrow \{0,1\}^l. \tag{26}$$

Here, $\overline{\mho}_{gd}$ is the list of selected $\mho$, where $\overline{\mho}_{gd} = \bigcup_{i=1}^{\Re_{gd}^{\mho}}(\overline{\mho}_i)$ After generating ε_Υ, Υ encrypts $\overline{\mho}_{gd}$ by employing $\hat{\varrho}_\kappa^\sigma$. Let $\hat{\gamma}$ is the encrypted data,

$$\hat{\gamma} = \xi_{\hat{\varrho}_\kappa^\sigma}(\overline{\mho}_{gd}, \varepsilon_\Upsilon). \tag{27}$$

After completing $\hat{\gamma}$, Υ returns $\hat{\gamma}$ to σ. Upon receiving $\hat{\gamma}$, σ first performs a decryption by using $\wp_\kappa^\sigma$,

$$\dot{d} = \zeta_{\wp_\kappa^\sigma}(\hat{\gamma}). \tag{28}$$

Subsequently, σ verifies the integrity from ε_Υ. Let $\dot{v}$ is the result of verification,

$$\dot{v} = \omega_{\hat{\varrho}_\kappa^\Upsilon}(\overline{\mho}_{gd}, \varepsilon_\Upsilon) \text{ where } \dot{v} \in \{true, false\}. \tag{29}$$

If $\dot{v}$ returns a value of $true$ then σ reads the list, connects with a suitable $\mho$, and continues to transmit data. Let $T^{\{\sigma\rightarrow\Upsilon\rightarrow\sigma\}}$ is the total time for requisitioning a UAV and get response from Υ,

$$T^{\{\sigma\rightarrow\Upsilon\rightarrow\sigma\}} = \tau_{\ddot{\varepsilon}}^{\sigma} + \tau_{\hat{\beta}}^{\sigma} + \tau_{\hat{t}t}^{\{\sigma\rightarrow\Upsilon\}} + \tau_{\acute{d}}^{\Upsilon} + \tau_{\grave{f}}^{\Upsilon} \\ + \tau_{\check{v}}^{\Upsilon} + \tau_{\overline{\mho}}^{\Upsilon} + \tau_{\ddot{u}_{dist}}^{\Upsilon} + \tau_{\varepsilon_\Upsilon}^{\Upsilon} + \tau_{\hat{\gamma}}^{\Upsilon} + \tau_{\hat{t}t}^{\{\Upsilon\rightarrow\sigma\}}. \tag{30}$$

IV. SECURITY ANALYSIS

A. Protection Against Eavesdroppers

In BUAV, data is transferred from σ to Υ via $\mho$ and in addition, σ sends the request for $\mho$ to Υ. The area between σ and $\mho$, $\mho$ and Υ, and σ and Υ are vulnerable to an eavesdropper ϑ. However, BUAV provides protection against eavesdroppers. In BUAV, every σ, Υ, and $\mho$ have their own $\wp_\kappa$ and a corresponding $\hat{\varrho}_\kappa$. When σ transmits α to $\mho$, σ encrypts the information by utilizing (6). Upon receiving β, it is decrypted by $\mho$ using (9) and the identity is verified using (10). After verification, $\mho$ re-encrypts information using (11) and forwards γ to Υ. While requesting the location of a new $\mho$, σ first encrypts the request using (21) and sends the request to Υ. Upon receiving $\hat{\beta}$, it is decrypted by Υ by using (22) and the identity is verified by employing (23). After processing the request, Υ encrypts the response using (27) and returns to σ. In order to read data between σ and $\mho$, ϑ requires knowledge of $\wp_\kappa^\mho$, which is only known to $\mho$. Without $\wp_\kappa^\mho$, ϑ cannot decrypt the data. If ϑ intends to read γ then $\wp_\kappa^\Upsilon$ is needed which is only known to Υ. In order to read $\hat{\beta}$ and $\hat{\gamma}$, ϑ needs to know $\hat{\varrho}_\kappa^\Upsilon$ and $\hat{\varrho}_\kappa^\sigma$, respectively, which are not publicly available. As such, ϑ cannot exercise eavesdropping in BUAV.

B. Key-Spoof Resistance

An attacker $\ddot{\vartheta}$ cannot directly perform eavesdropping because $\wp_\kappa$ is unknown to $\ddot{\vartheta}$. To determine $\wp_\kappa$, $\ddot{\vartheta}$ must guess $\wp_\kappa$. According to (1), $\wp_\kappa$ is generated using δ_{mac}, τ_c and $\hat{S}_\hbar$. τ_c and $\hat{S}_\hbar$ are totally random. Suppose, $\wp_\kappa$ is 32 bytes or 256 bits long. To predict the correct $\wp_\kappa$, $\ddot{\vartheta}$ has to go through the sequence of 256. For 256 bits, there are 2^{256} possible sequences and among them, only one can be the right $\wp_\kappa$. The probability of predicting $\wp_\kappa$ is $1/2^{256} = 2^{-256}$ and that is not practically feasible. Let, δ_{mac} be $\hat{\imath}$ bits, τ_c represents $\hat{\jmath}$ bits, and $\hat{S}_\hbar$ represents $\hat{k}$. If $\ddot{\vartheta}$ wants to guess the properties of $\wp_\kappa$, then the probability of guessing correclty is $1/2^{\hat{\imath}} \times 1/2^{\hat{\jmath}} \times 1/2^{\hat{k}} = 2^{-\hat{\imath}} \times 2^{-\hat{\jmath}} \times 2^{-\hat{k}}$, which is also not practically feasible.

C. Data Tampering Resistance

In BUAV, σ transfers α to Υ, via $\mho$. When Υ receives γ from $\mho$, Υ first decrypts γ by employing (14) and subsequently, verifies the identity using (15). However, Υ requires verification that data originates from the original sender, σ. $\mho$ decrypts β by employing (9) and retrieves α. In this case, there is a vulnerability in that the data may experience alteration. Before transmitting α, σ creates a signature of α using (5). When Υ receives α, Υ verifies α by employing (16). If $\ddot{v}$ is $false$, then α is altered and Υ discards α. Υ only appends data to blockchain when $\ddot{v}$ is $true$. During the requisition of $\mho$, σ creates a signature by utilizing (20), so that, Υ can verify the authenticity of σ using (24). Based on this approach, BUAV prevents altered data from being appended to Blockchain. After storing data in traditional databases, there is the possibility that the data may be altered. Considering this issue, BUAV stores the data in Blockchain. When data arrives at Υ, after verification, Υ transmits the data to the blockchain network. Once the data is received in the network, validators create blocks after utilizing (19). Given that data is stored in the chain, every validator holds the same copy of the data. In blockchain, every block has a unique hash which is the identity of that block. The next block keeps the hash of the previous block and they are chained together. However, this hash is generated from the data, timestamp, nonce, etc. To main-

Table 2. Simulation parameters.

Parameters	Values
Devices, η	[500,$\cdots$,10000], [1, 2, 3]
MECS_{CPU}	Intel(R) Core(TM) i5-4670 CPU @ 3.40 GHz
$\text{MECS}_{\text{STORAGE,RAM,OS}}$	1 TB, 32 GB, Ubuntu 18.04.1
UAV_{CPU}	Broadcom BCM2837B0, Cortex-A53 (ARMv8) 64-bit SoC @ 1.4 GHz
$\text{UAV}_{\text{STORAGE,RAM,OS}}$	16 GB, 1 GB, Raspbian Stretch

tain integrity, blockchain uses a Merkle tree. This is a structure in which the leaf nodes holds the hash of the data. When an alteration is performed to the data, the hash of the tree is also changed. Subsequently, the hash of the block is also changed and thus, the chain of blocks break. To restore the chain, consent from the majority of the validator is required, which makes these changes technically unfeasible.

V. PERFORMANCE EVALUATION

This section represents the performance evaluation of BUAV. The performance is analyzed by simulation and experiment and the results are discussed is two subsections.

A. Simulation Results

A simulation was performed using MATLAB for estimating the effects in MEC server and another simulation was performed for estimating the effects in UAV. In UAV, Python was utilized for simulating results. The simulation parameters are provided in Table 2. For simulation purposes, an Intel(R) Core(TM) i5-4670 CPU @ 3.40 GHz was used as MEC server with 32 GB RAM. For UAV, Broadcom BCM2837B0, Cortex-A53 (ARMv8) 64-bit SoC @ 1.4 GHz was used with 1 GB RAM.

Fig. 3 demonstrates the results of the simulation that were performed in the MEC server. The time for processing the data of η-hash bloom filter that contains information on the different number of devices is shown in Fig. 3(a). Milliseconds was considered as the unit of the processing time. In Fig. 3(a), processing time increases with the increase in the number of devices. In addition, the value of η affects the processing time. Increasing the number of η results in an increase in the processing time. As a result, $\eta = 3$ requires more time than $\eta = 2$, and $\eta = 2$ requires more time-consuming than $\eta = 1$.

Fig. 3(b) illustrates the expected transmission of data containing information for the different devices that are generated from η-hash bloom filter. An increase in the number of devices also results in an increase in the data size. However, $\eta = 1, 2,$ and 3 produces the same amount of data because data in the list contains either 0 or 1 and the size of 0 and 1 is same. Therefore, differences in η have no impact on the data size generated from η-hash bloom filter.

Fig. 3(c) depicts the transmission of data generated from bloom filter and other data structure apart from bloom filter containing the information of the different number of devices. A hash table was considered as a non-bloom filter data structure.

Table 3. Experimental parameters.

Parameters	Values
η	1
Wi-Fi, Bluetooth	48 Mbps, Bluetooth 4.2
$\psi(.)$, Encryption	SHA-256, RSA (key = 1024)
IoTD_{n}	4
IoTD_{CPU}	Broadcom BCM2837B0, Cortex-A53 (ARMv8) 64-bit SoC @ 1.4 GHz
$\text{IoTD}_{\text{STORAGE,RAM,OS}}$	4 GB, 1 GB, Raspbian Stretch
UAV_{n}	1
UAV_{CPU}	Broadcom BCM2837B0, Cortex-A53 (ARMv8) 64-bit SoC @ 1.4 GHz
$\text{UAV}_{\text{STORAGE,RAM,OS}}$	16 GB, 1 GB, Raspbian Stretch

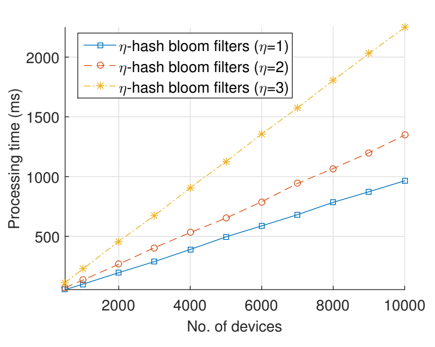
Given that, there is no influence of η on data size, $\eta = 1$ was used for this simulation. The amount of data in both the bloom filter and the hash table increases with an increase in the number of devices. However, the data size of the hash table increases dramatically in comparison with the bloom filter. In particular, the hash table is initially twice of the data size of the bloom filter. For more devices, this difference approaches thrice the data size of the bloom filter. Therefore, by using a bloom filter, BUAV compresses space in a UAV.

Fig. 4 demonstrates the simulation results that were obtained for the UAV. The time for validating the identity of the different devices utilizing η-hash bloom filter is shown in Fig. 4(a). Milliseconds was considered as the unit for the validation time. With the increase in the number of devices that require validation, the validation time also increases. The change in the value of η leads to a change in the validation time. Increasing the number of η requires more time for the validation process. This is the reason why $\eta = 3$ requires more time than $\eta = 2$ and $\eta = 2$ requires more time than $\eta = 1$.

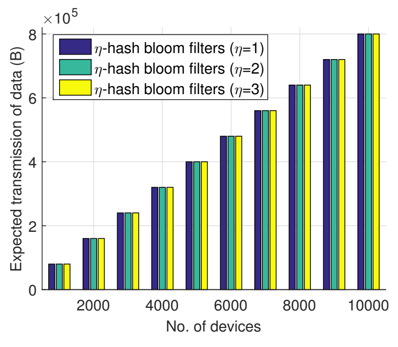
Fig. 4(b) illustrates the energy consumption during the validation process. Joule was considered as the unit of energy consumption. With the increase in the number of devices for the validation process, energy consumption also increases. Given that, the increase in η requires more time for the validation process, $\eta = 3$ consumes more energy than $\eta = 2$ and $\eta = 2$ consumes more energy than $\eta = 1$.

Fig. 4(c) represents the time required to validate the devices in the presence of malicious devices. This simulation was performed for $10,000$ devices. The increase in the percentage of malicious devices, even in the 100 percentage malicious devices scenario, does not affect the validation time. This is because, for both valid and malicious devices, η-hash bloom filter checks either 0 or 1. However, the validation time increases with an increase in η; but, it remains the same for different percentages of malicious devices.

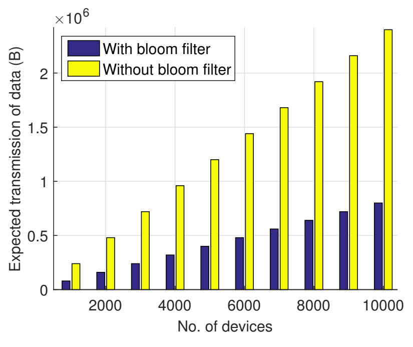
Fig. 5 demonstrates the rate of false positive for the different percentage of malicious devices. As bloom filter surrounds with the false positive issue, it is necessary to check the false positive rate. $10,000$ devices were considered for this simulation. In this simulation, the false positive rate is 0 for $\eta = 1, 2,$ and 3, even



(a)

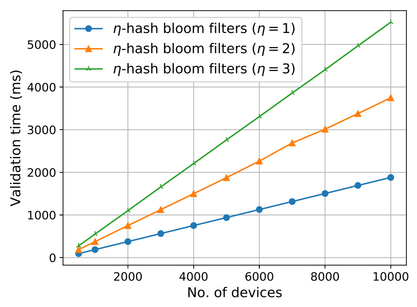


(b)

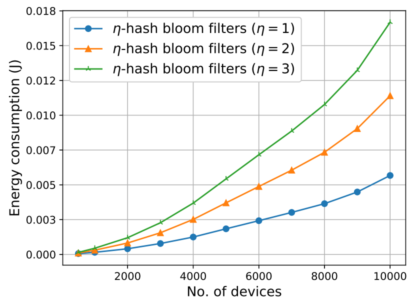


(c)

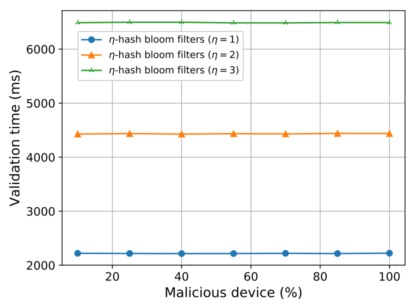
Fig. 3. Results of simulation that was performed in MEC server: (a) Time for processing devices, (b) expected transmission of processed data, and (c) expected transmission of processed data for η-hash bloom filter and for without η-hash bloom filter.



(a)



(b)



(c)

Fig. 4. Results of simulation that was performed in UAV: (a) Time for validating devices, (b) energy consumption during the validation, and (c) time for validating devices in the presence of malicious devices.

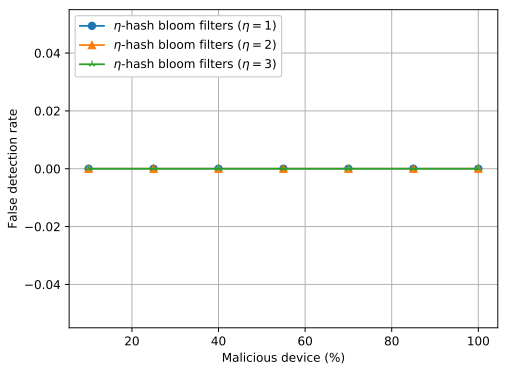


Fig. 5. Rate of false detection for validating devices.

in the 100 percentage malicious devices scenario.

B. Experimental Results

The experiment was performed indoor, as shown in Fig. 6. The parameters used in the experiment are described in Table 3. An Intel(R) Core(TM) i5-4670 CPU @ 3.40 GHz was used as MEC server with 32 GB RAM. Node.js was used to create the server. 4 raspberry pi 3 model b+ were utilized and sensors (i.e., flame, temperature, humidity, light) were attached in these devices. These devices were considered as IoTD with the primary task of sensing the environment and transmitting the result to the MEC server via UAV. The middleware in IoTD was built using Python. The communication between UAV and IoTD was performed over Bluetooth and the communication between UAV and MEC server was performed over Wi-Fi. Parrot Bebop 2 was used as a UAV and raspberry pi 3 model b+ was attached for maintaining communication with IoTD and MEC server, as shown in Fig. 6(a). The middleware for UAV was written using Python. IoTDs were deployed randomly and UAV was placed in the center of IoTDs, as shown in Figs. 6(b) and 6(c). The proposed scheme was built over ethereum. In ethereum, a private network was created, named BUAV-B, containing 10 validators. Geth was used as an ethereum client and web.js for RPC call. In order to set up 10 validators, 10 computers were used to create a local area network (LAN). Each pc contained geth and connected with each other. Proof of authority (PoA) was used for performing the consensus mechanism.

Fig. 7 shows the energy consumption for transferring data from a IoTD to other entities (e.g., UAV, MEC server, and cloud) directly. Intel(R) Xeon(R) Processor E5-2697A V4 @ 2.60 GHz with 32 GB RAM was considered for the cloud server and the cloud server is hosted in US server at 69.162.66.34. CentOS 7.5 was used as the OS in the cloud server. In order to connect to the cloud, the IoT device is connected with the ipTIME A2004NS-R router and the EFM ipTIME A2004NS-R is connected to the internet. To connect to the MEC server, an ipTime N100 mini was used to open an access point from the MEC server. To connect to the UAV, Bluetooth 4.2 was used. Fig. 7 shows that it requires

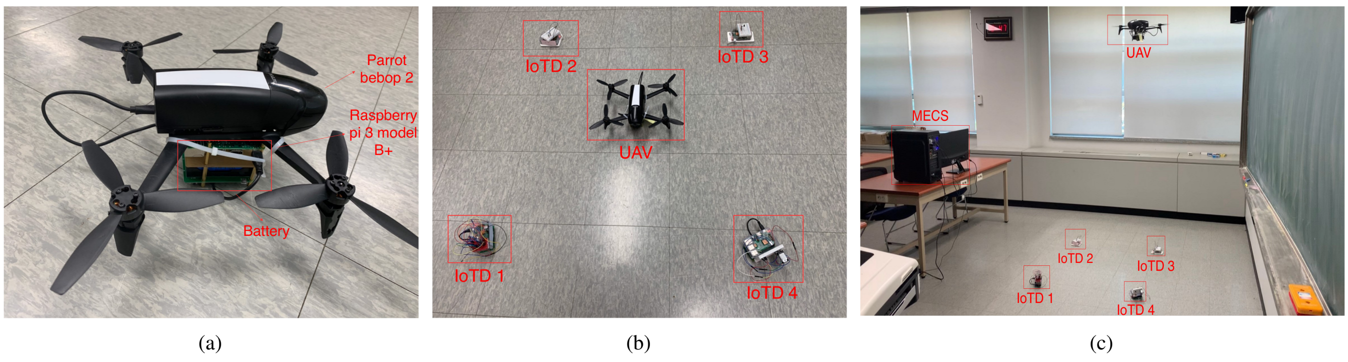


Fig. 6. Experimental setup: (a) Hardware details of UAV, (b) UAV with IoTDs, and (3) data acquisition from IoTDs to MEC server via UAV.

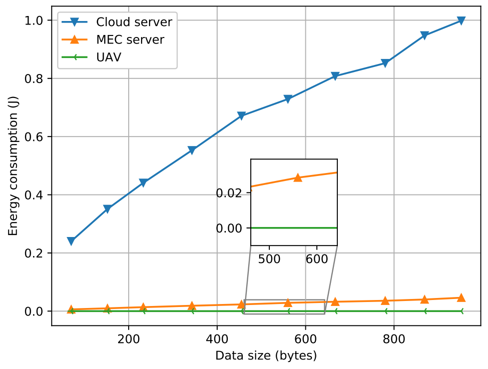


Fig. 7. Energy consumption of IoT devices to communicate with UAV, MEC server, and cloud server for different data sizes.

more energy to communicate with the cloud, and with the increase in data size, energy consumption also increases. On the contrary, communication with the MEC server consumes less energy although the energy consumption increases with an increase in the size of the data. Communication with UAV consumes less energy than the MEC server which is almost half of the MEC server because communicating using Bluetooth consumes less energy than Wi-Fi. With an increase in data size, the energy consumption increases very slowly in contrast with the MEC and cloud server. From the results, direct communication with the cloud server, IoTs spent more energy than with the MEC server and direct communication with the MEC, IoTs spent more energy than with UAV.

Fig. 8 demonstrates the result of the experiments performed in the network. In Fig. 8(a), the throughput of the network over different time is presented. The communication channel of BUAV is compared before and after the application of security. Over time, in both channel's throughput increases. However, the throughput of the channel without security is higher than the channel with security. This is because, after applying security, data has to be validated before being forwarded to the next hop.

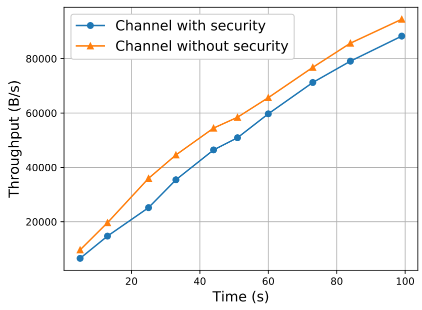
Fig. 8(b) illustrates the total amount of data received in MEC server. For this experiment, 1 and 3 were considered as valid IoTDs and 2 and 4 were considered as malicious IoTDs. This simulation was performed for 10 s. Data from IoT-1 and IoT-3 successfully reached MEC server because both of them are valid. As a result, both successfully passed the validation process in UAV and $10,370$ and $10,889$ bytes received at MEC server, respectively. However, data from IoT-2 and IoT-4 were discarded in UAV because they could not pass the validation because they are not in the device list. Therefore, IoT-2 and IoT-4 cannot reach MEC server.

Fig. 8(c) shows the time to process security actions in IoT device, UAV, and MEC server for different data sizes. Equation (7) was utilized for IoT device, (12) was utilized for UAV, and (17) was utilized for the MEC server in order to calculate the processing time. The processing time increased with the increase of the data size for IoT device, UAV, and MEC server. However, MEC server's computation power is much higher than that of UAV and IoT. That is why the processing time for MEC server is negligible in contrast with IoT device and UAV. However, UAV performs a decryption process which is slower than encryption. As such, the processing time is higher in UAV than IoT device.

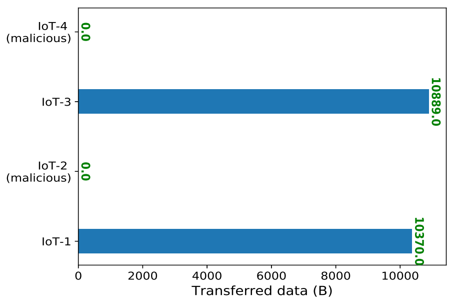
Fig. 8(d) represents the energy consumption during the processing of security actions in IoT device, UAV, and MEC server for different data sizes. Energy consumption in the CPU was considered while processing the security actions of the MEC server. Given that, the CPU has significant computational power, the processing time is much less in comparison with that of UAV and IoT device. However, MEC server requires a large amount of energy in order to maintain continuity. With the increase in the data size, the energy consumption of the IoT device, UAV and MEC server also increases. Given that, UAV has a decryption mechanism in security actions, it consumes more power than IoT device.

Fig. 8(e) depicts the processing time for performing individual security mechanism for different data sizes. Given that, MEC has significant computational power, the difference in the processing time between MEC and IoT/UAV is very high. It is notable that the processing time for every action increases with an increase in the data size. However, decryption requires more time than encryption and signing the data is more time consuming than verifying the data. In RSA, the private key is larger than the public key and due to its size, more time is needed for processing. Both decryption and signing use a private key in order to decrypt and sign, respectively. As such, decryption and signing require more time in contrast to encryption and verifying. However, this phenomenon is also visible in the MEC server. Despite its high computation capability, decryption and signing take more time than encryption and verifying.

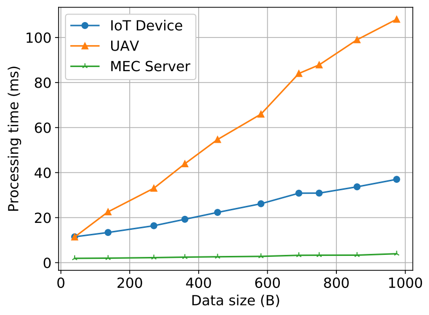
Fig. 8(f) represents the latency for searching UAV that varies



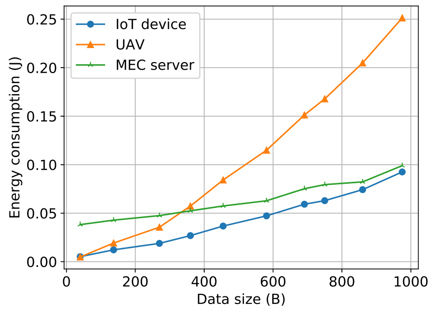
(a)



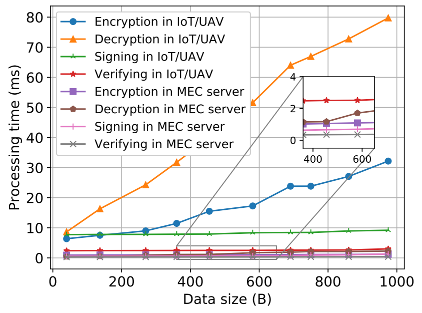
(b)



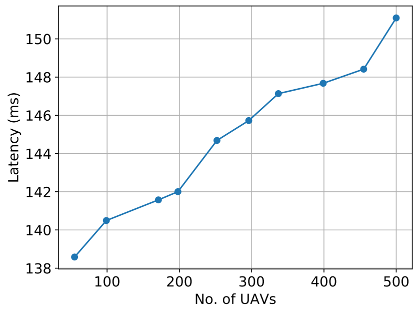
(c)



(d)

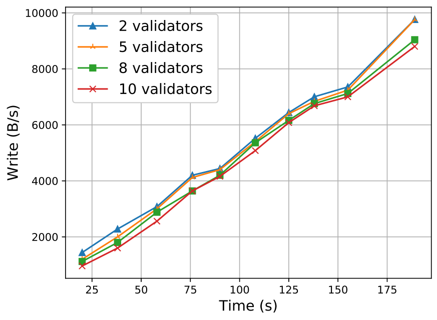


(e)

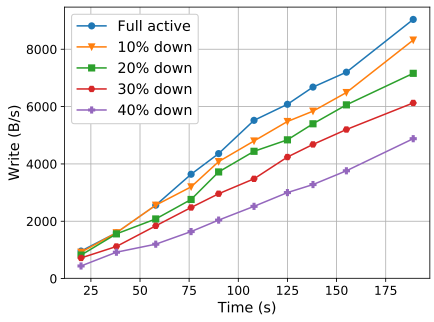


(f)

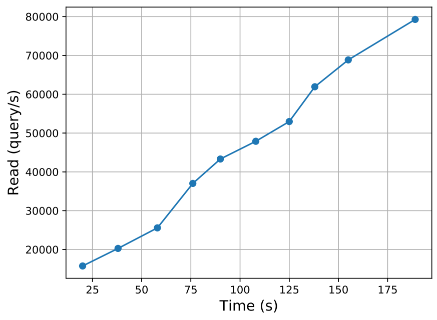
Fig. 8. Results of experiment that was performed in BUAV: (a) Throughput of the network, (b) transferred of data to MEC server both from valid devices and malicious devices, (c) time for processing the security modules on different data size, (d) processing time for individual security actions on different data size, (e) energy consumption for performing individual security actions on different data size, and (f) latency for requisitioning UAV to MEC server.



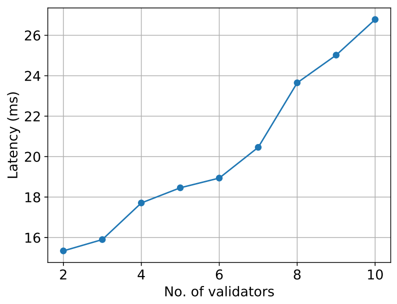
(a)



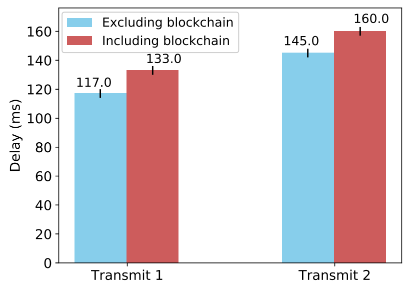
(b)



(c)



(d)



(e)

Fig. 9. Results of experiment that was performed in BUAV-B: (a) Data added in the blockchain over time, (b) data added while the absence of validators in the network over time, (c) data retrieved over time, (d) latency while appending a data over different validator, and (e) delay while transferring data via different channel w.r.t both including and excluding blockchain.

for the different number of UAVs. Equation (30) was utilized in order to calculate latency. However, with the increase in the number of UAVs, latency also increases. For more UAVs, it is required more time to calculate distance and more time to sort the data set which increases the latency. Overall, that increase in latency is very small.

Fig. 9 illustrates the result of the experiments performed in blockchain in MEC server. Fig. 9(a) shows the amount of data that was added in blockchain over time. 2, 5, 8, and 10 validator scenarios were considered during these experiments. With the increase in time, the amount of data addition also increased. However, these increases vary depending on the number of validators. The speed of adding data is almost the same for 2 and 5, and the speed of data addition is almost the same for 5 and 10. However, with the increase of the validator, the rate of addition of data in the blockchain decreases. In PoA, from the onset, everyone has the list of validators and blocks are appended in a round robin manner. Prior to the addition of the block in the chain, it is necessary to get an agreement by validators which increases the delay in the network and as a result of this delay, the amount of data also decreases.

Fig. 9(b) represents the changes of the data added in blockchain over time when considering the different percentage of validators that are not available for the validation process. 10 validators were considered while performing this experiment. The experiment was performed for the full active node (everyone is active), 10% down (10% of the total validators), 20% down (20% of the total validators), 30% down (30% of the total validators), and 40% down (40% of the total validators). However, in PoA, one validator proposes a block and the other validators sign it if the proposal is valid. This process proceeds one at a time in a round robin way. However, if a validator is not available for the validation process, he misses the chance to propose a block, his chance moves to the next validator, and this process goes on. But, this creates a delay in the network which causes a decrease in the amount of data added to the network in contrast with the full active node. With the increase in the percentage of unavailability of validators, the rate of addition of data in the network decreases.

Fig. 9(c) outlines the number of queries that can be performed at different times. In blockchain, everyone has the same version of the data and during data reading from blockchain, data were retrieved from the local copy of the requester. This is why the validators do not play a role during the reading of data from blockchain. However, how many queries can be performed at different times were considered. With the increase in time, the queries also increase.

Fig. 9(d) describes the latency associated with the addition of data in blockchain for the different validators. From 2 to 10 validators appearance were considered. In Fig. 9(d), latency increases with the increasing number of validators. It is notable that, to sign a block, a delay is created from the validators. Thus, an increase in the number of validators increases latency.

Fig. 9(e) presents the delay in transferring data from IoTD to MEC server. In this experiments, data were transmitted in two ways: (1) Transmit 1 (IoTD$\rightarrow$UAV$\rightarrow$MECS), and (2) Transmit 2 (IoTD$\rightarrow$UAV$\rightarrow$UAV$\rightarrow$MECS). Transmission by including blockchain (IoTD $\rightarrow$ UAV $\rightarrow$ MECS $\rightarrow$ Blockchain and IoTD $\rightarrow$ UAV $\rightarrow$ UAV $\rightarrow$ MECS $\rightarrow$ Blockchain) and excluding blockchain were considered. For the case without blockchain, the delay in Transmit 2 is higher than Transmit 1. This is because, in Transmit 2, the data has to pass through one more hop than in the case of Transmit 1. In addition, this data also experiences more security mechanisms than Transmit 1, which increases the delay in Transmit 2 in compare to Transmit 1. In comparison with the exclusion of blockchain, the incorporation of blockchain increases the delay in both Transmit 1 and Transmit 2. This is because the inclusion of blockchain introduces additional latency during the addition of the block in blockchain along with the security validation delay in MEC server.

VI. CONCLUSIONS AND FUTURE WORK

In this paper, a secure data collection scheme was introduced in which data are collected from IoT devices (IoTDs) with the assistance of unmanned aerial vehicle (UAV). After collection, the data are stored securely in blockchain at mobile edge computing (MEC) server. While transferring data from IoTDs, encryption is performed using UAV's public key and a signature is created using IoTD's private key. UAV receives that data and after the decryption, the identity of IoTD is validated using η-hash bloom filter. Subsequently, data are forwarded to MECS and after validating, it is stored by MEC in blockchain that is signed by the validators. Simulations were performed using MATLAB in order to examine the impact of η-hash bloom filter for both in MEC server and UAV. Implementation of BUAV was done and experiments were performed in that implementation in to order to test the feasibility. Some future directions are listed below:

- BUAV supports single UAV in the data acquisition which can be extended with multi-UAV to increase the quality of service. But, providing security for multi-UAV is very challenging which can be a future research topic.
- The dynamic distribution of IoT devices and mobility of UAV during secure data acquisition needs to be investigated which can be subjected to future works.
- BUAV does not consider the incentivization of the validator which is kept for future research.

REFERENCES

[1] P. Asghari, A. M. Rahmani, and H. H. S. Javadi, "Internet of things applications: A systematic review," *Computer Netw.*, vol. 148, pp. 241–261, 2019.

[2] J. Portilla, G. Mujica, J. Lee, and T. Riesgo, "The extreme edge at the bottom of the internet of things: A review," *IEEE Sensors J.*, vol. 19, no. 9, pp. 3179–3190, 2019.

[3] M. M. Alam, H. Malik, M. I. Khan, T. Pardy, A. Kuusik, and Y. L. Moullec, "A survey on the roles of communication technologies in IoT-based personalized healthcare applications," *IEEE Access*, vol. 6, pp. 36611–36631, 2018.

[4] D. He, R. Ye, S. Chan, M. Guizani, and Y. Xu, "Privacy in the internet of things for smart healthcare," *IEEE Commun. Mag.*, vol. 56, no. 4, pp. 38–44, Apr. 2018.

[5] A. A. Abdellatif, M. G. Khafagy, A. Mohamed, and C. Chiasserini, "Eeg-based transceiver design with data decomposition for healthcare IoT applications," *IEEE Internet Things J.*, vol. 5, no. 5, pp. 3569–3579, Oct. 2018.

[6] M. Elhoseny, G. Ramírez-González, O. M. Abu-Elnasr, S. A. Shawkat, A. N, and A. Farouk, "Secure medical data transmission model for IoT-based healthcare systems," *IEEE Access*, vol. 6, pp. 20596–20608, 2018.

[7] E. Luo, M. Z. A. Bhuiyan, G. Wang, M. A. Rahman, J. Wu, and M. Atiquzzaman, "Privacyprotector: Privacy-protected patient data collection in IoT-based healthcare systems," *IEEE Commun. Mag.*, vol. 56, no. 2, pp. 163–168, Feb. 2018.

[8] P. Verma and S. K. Sood, "Fog assisted-IoT enabled patient health monitoring in smart homes," *IEEE Internet Things J.*, vol. 5, no. 3, pp. 1789–1796, June 2018.

[9] R. Dagar, S. Som, and S. K. Khatri, "Smart farming – IoT in agriculture," in *Proc. ICIRCA*, July 2018, pp. 1052–1056.

[10] S. J. Balakrishna, H. Marellapudi, and N. A. Manga, "IoT based status tracking and controlling of motor in agricultural farms," in *Proc. IEEE UPCON*, Nov. 2018, pp. 1–5.

[11] I. Zyrianoff, A. Heideker, D. Silva, and C. Kamienski, "Scalability of an internet of things platform for smart water management for agriculture," in *Proc. FRUCT*, Nov. 2018, pp. 432–439.

[12] V. Ramachandran, R. Ramalakshmi, and S. Srinivasan, "An automated irrigation system for smart agriculture using the internet of things," in *Proc. ICARCV*, Nov. 2018, pp. 210–215.

[13] Y. Meng, W. Zhang, H. Zhu, and X. S. Shen, "Securing consumer IoT in the smart home: Architecture, challenges, and countermeasures," *IEEE Wireless Commun.*, vol. 25, no. 6, pp. 53–59, Dec. 2018.

[14] A. Elsts *et al.*, "Enabling healthcare in smart homes: The sphere IoT network infrastructure," *IEEE Commun. Mag.*, vol. 56, no. 12, pp. 164–170, Dec. 2018.

[15] P. Wang, F. Ye, and X. Chen, "A smart home gateway platform for data collection and awareness," *IEEE Commun. Mag.*, vol. 56, no. 9, pp. 87–93, Sept. 2018.

[16] C. K. Lee, H. Liu, D. Fuhs, A. Kores, and E. Waffenschmidt, "Smart lighting systems as a demand response solution for future smart grids," *IEEE J. Emerging Sel. Topics Power Electron.*, p. 1, 2019.

[17] Y. Li, X. Cheng, Y. Cao, D. Wang, and L. Yang, "Smart choice for the smart grid: Narrowband internet of things (NB-IoT)," *IEEE Internet Things J.*, vol. 5, no. 3, pp. 1505–1515, June 2018.

[18] L. d. M. B. A. Dib, V. Fernandes, M. de L. Filomeno, and M. V. Ribeiro, "Hybrid plc/wireless communication for smart grids and internet of things applications," *IEEE Internet Things J.*, vol. 5, no. 2, pp. 655–667, Apr. 2018.

[19] Y. Sun, L. Lampe, and V. W. S. Wong, "Smart meter privacy: Exploiting the potential of household energy storage units," *IEEE Internet Things J.*, vol. 5, no. 1, pp. 69–78, Feb. 2018.

[20] H. A. H. Hassan, D. Renga, M. Meo, and L. Nuaymi, "A novel energy model for renewable energy-enabled cellular networks providing ancillary services to the smart grid," *IEEE Trans. Green Commun. Netw.*, vol. 3, no. 2, pp. 381–396, June 2019.

[21] D. E. Kouicem, A. Bouabdallah, and H. Lakhlef, "Internet of things security: A top-down survey," *Comput. Netw.*, vol. 141, pp. 199–221, Aug. 2018.

[22] L. Gupta, R. Jain, and G. Vaszkun, "Survey of important issues in UAV communication networks," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 2, pp. 1123–1152, Secondquarter 2016.

[23] S. Hayat, E. Yanmaz, and R. Muzaffar, "Survey on unmanned aerial vehicle networks for civil applications: A communications viewpoint," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 4, pp. 2624–2661, Fourthquarter 2016.

[24] Q. Zhang *et al.*, "IoT enabled UAV: Network architecture and routing algorithm," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 3727–2742, 2019.

[25] T. Lagkas, V. Argyriou, S. Bibi, and P. Sarigiannidis, "UAV IoT framework views and challenges: Towards protecting drones as "things"," *Sensors*, vol. 18, no. 11, Nov. 2018.

[26] S. K. Datta, J. Dugelay, and C. Bonnet, "IoT based UAV platform for emergency services," in *Proc. IEEE ICTC*, Oct. 2018, pp. 144–147.

[27] H. Elazhary, "Internet of things (IoT), mobile cloud, cloudlet, mobile IoT, IoT cloud, fog, mobile edge, and edge emerging computing paradigms: Disambiguation and research directions," *J. Netw. Comput. Appl.*, vol. 128, pp. 105–140, Nov. 2019.

[28] R. Roman, J. Lopez, and M. Mambo, "Mobile edge computing, fog et al.: A survey and analysis of security threats and challenges," *Future Generation Comput. Syst.*, vol. 78, pp. 680–698, Jan. 2018.

[29] Y. Ai, M. Peng, and K. Zhang, "Edge computing technologies for internet of things: A primer," *Digital Commun. Netw.*, vol. 4, no. 2, pp. 77–86, Apr. 2018.

[30] N. Mohamed, J. Al-Jaroodi, I. Jawhar, H. Noura, and S. Mahmoud, "UAVFog: A UAV-based fog computing for internet of things," in *Proc. IEEE SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI*, Aug. 2017, pp. 1–8.

[31] V. Sharma, R. Kumar, and R. Kaur, "UAV-assisted content-based sensor search in IoTs," *Electron. Lett.*, vol. 53, no. 11, pp. 724–726, Apr. 2017.

[32] S. Liu, Z. Wei, Z. Guo, X. Yuan, and Z. Feng, "Performance analysis of UAVs assisted data collection in wireless sensor network," in *Proc. IEEE VTC Spring*, June 2018, pp. 1–5.

[33] M. Bacco, A. Berton, A. Gotta, and L. Caviglione, "Ieee 802.15.4 air-ground UAV communications in smart farming scenarios," *IEEE Commun. Lett.*, vol. 22, no. 9, pp. 1910–1913, Sept. 2018.

[34] X. He, J. Bito, and M. M. Tentzeris, "A drone-based wireless power transfer anc communications platform," in *Proc. IEEE WPTC*, May 2017, pp. 1–4.

[35] A. S. Gaur, J. Budakoti, C. Lung, and A. Redmond, "IoT-equipped UAV communications with seamless vertical handover," in *Proc. IEEE IDSC*, Aug. 2017, pp. 459–465.

[36] X. Wang, X. Zha, W. Ni, R. P. Liu, Y. J. Guo, X. Niu, and K. Zheng, "Survey on blockchain for internet of things," *Comput. Commun.*, 2019.

[37] I. Makhdoom, M. Abolhasan, H. Abbas, and W. Ni, "Blockchain's adoption in IoT: The challenges, and a way forward," *J. Netw. Comput. Appl.*, vol. 125, pp. 251–279, Nov. 2019.

[38] Q. Feng, D. He, S. Zeadally, M. K. Khan, and N. Kumar, "A survey on privacy protection in blockchain system," *J. Netw. Comput. Appl.*, vol. 126, pp. 45–58, Nov. 2019.

[39] Y. Yu, Y. Li, J. Tian, and J. Liu, "Blockchain-based solutions to security and privacy issues in the internet of things," *IEEE Wireless Commun.*, vol. 25, no. 6, pp. 12–18, Dec. 2018.

[40] T. M. Fernández-Caramés and P. Fraga-Lamas, "A review on the use of blockchain for the internet of things," *IEEE Access*, vol. 6, pp.32979–33001, May 2018.

[41] A. Islam, M. B. Uddin, M. F. Kader, and S. Y. Shin, "Blockchain based secure data handover scheme in non-orthogonal multiple access," in *Proc. IEEE ICWT*, July 2018, pp. 1–5.

[42] A. Reyna, C. Martín, J. Chen, E. Soler, and M. Díaz, "On blockchain and its integration with IoT. challenges and opportunities," *Future Generation Comput. Syst.*, vol. 88, pp. 173–190, Nov. 2018.

[43] F. Grandi, "On the analysis of bloom filters," *Inf. Process. Lett.*, vol. 129, pp. 35–39, Jan. 2018.

[44] L. Luo, D. Guo, R. T. B. Ma, O. Rottenstreich, and X. Luo, "Optimizing bloom filter: Challenges, solutions, and comparisons," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 2, pp. 1912–1949, Dec. 2018.

Anik Islam was born in 1992. He received the B.Sc. in software engineering and M.Sc. degrees in computer science from American International University-Bangladesh (AIUB), Dhaka, Bangladesh, in 2014 and 2017, respectively. He is currently working toward the Ph. D. degree with the WENS Laboratory, Kumoh National Institute of Technology, Gumi, South Korea. He has more than 5 years of experience of working in the software development field. He has participated in various software competitions with good achievements. His major research interests include blockchain, internet of things, unmanned aerial vehicle, social internet of things, edge computing, and distributed system.

Soo Young Shin received his Ph. D. degree in Electrical Engineering and Computer Science from Seoul National University on 2006. He was with WiMAX Design Lab, Samsung Electronics, Suwon, South Korea from 2007 to 2010. He joined as Full-time Professor to School of Electronics, Kumoh National Institute of Technology, Gumi, South Korea. He is currently an Associate Professor. He was a Post Doc. Researcher at University of Washington, Seattle, WA, USA from 2006 to 2007. In addition, he was a visiting scholar to University of the British Columbia at 2017. His research interests include wireless communications, next generation mobile wireless broadband networks, signal processing, Internet of things, etc.