Hierarchical Blockchain-based Group and Group Key Management Scheme Exploiting Unmanned Aerial Vehicles for Urban Computing

GABIN HEO\textsuperscript{1}, KIJOON CHAE\textsuperscript{2}, AND INSHIL DOH\textsuperscript{3}

\textsuperscript{1}Division of Artificial Intelligence and Software, Ewha Womans University, Seoul, Korea
\textsuperscript{2}Department of Computer Science and Engineering, Ewha Womans University, Seoul, Korea
\textsuperscript{3}Department of Cyber Security, Ewha Womans University, Seoul, Korea

Corresponding author: Inshil Doh (e-mail: isdoh1@ewha.ac.kr).

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea Government [Ministry of Science and ICT (MSIT)] under Grant NRF-2019R1F1A1063194 and Grant NRF-2020R1A2C1006497.

\textbf{ABSTRACT} In urban computing composed of various Internet of Things (IoT) devices, data collected from IoT as learning data for other IoT devices in a circular structure. Therefore, data are shared resources and crucial in urban computing. In particular, group communication is essential for effective data sharing and updating software in an IoT system. However, changing the group key is imperative for data security in group communication whenever the group member changes. When the movement of IoTs is high, group key redistribution must be repeated because the group membership keeps changing. In changing and redistributing the group keys, overhead occurs, inevitably degrading system performance. In addition, since much data traffic generates in IoT devices’ crowded areas, transmission delay and data loss could degrade system performance. Due to IoT’s low power and low capacity characteristics, one-to-one communication is inefficient, requiring efficient group management and group key management. This work proposes a hierarchical blockchain-based group and group key management scheme to establish an efficient communication environment in urban computing. We adopted blockchains to track the movement and density of IoT and secure node authorization. Using the upper layer blockchain, the unmanned aerial vehicle (UAV) determines the movement and density of IoTs. Using the lower-layer blockchain, base stations (BSs) identify the IoT’s movement of information in each group. We included only nodes determined to be safe in the group. Through the hierarchical blockchain, while protecting the IoT’s privacy, we can record the information in the blockchain to determine the mobility of nodes and the node density of a group. We set several experimental environments, and analyzed the efficiency by simulating the addition of secondary gateways and group integration. As a result, we showed that our proposal establishes an efficient communication environment and reduces computational and communication overhead.

\textbf{INDEX TERMS} group management, group key management, urban computing, internet of things, blockchain, unmanned aerial vehicle, data security, privacy

\section{I. INTRODUCTION}

Recently, the era of the Internet of Everything (IoE) has arrived, in which not only people and things but also everything else is connected through the Internet. Therefore, every piece of information (data) becomes an important resource in the information age. In urban computing composed of various IoT devices, the cyclic structure of the data enables its use for learning for other IoT devices [1]. IoT sensors collect various data types, such as temperature, humidity, illumination, movement, and location. The IoT system analyzes and learns these data to create new information utilized within the urban system [2]. For example, in the case of a smart vehicle system, it is possible to determine the traffic congestion by using data such as vehicle GPS, speed, and destination. As a result, it reduces the risk of traffic accidents and helps identify the most appropriate route in case of an emergency.
In particular, when the data are various and abundant, the information generated is more accurate. However, when using IoT data, there are security problems. IoT devices have become a constant observer of users’ lives, exposing private data to malicious attackers [4]. With continuous IoT data exposure, malicious attackers can learn the user’s private information such as behavior, habits, hobbies, work, health status, etc. [5]. In addition, in urban computing that utilizes real-time data, if users manipulate data collected by IoTs, serious security threats occur; thus, it requires data accuracy. However, when the area becomes full of data traffic in urban computing, data transmission delay and data loss occur, resulting in poor data accuracy and system performance degradation. Thus, data traffic control is important [6]. Blockchain is emerging as a technology that guarantees data accuracy and integrity. It ensures transparency, data integrity, and high security by allowing all participants to own a transaction information ledger. In addition, the transaction’s reliability is secure because it occurs without a third party. However, due to the characteristic that all nodes must have the same data, there is a need for lightweight data. In addition, personal information leakage problems occur because anyone can check transaction information [7]. This problem is fatal for IoT devices where data security is essential, and it is also a lethal issue for an urban computing environment composed of various IoT devices. Therefore, users should seriously consider measures for protecting privacy in the blockchain.

Additionally, due to IoT’s low-power and low-capacity characteristics, there is a need for group management mechanisms for efficient group communication [8]. Data must also be encrypted and transmitted in group communication for IoT data protection. Since the group key for encryption must be shared only by group members, the system requires an efficient group key mechanism [9]. In this work, we consider urban computing composed of IoT devices such as wearable devices including various sensors, smart home devices, and smart vehicles. We also assume that groups are divided geographically for group communication, and that each group utilizes the data generated by IoT to create new information necessary for the area. For example, temperature and humidity data mean IoT data when predicting the weather, and weather means new information. We exploit unmanned aerial vehicles (UAVs) to protect IoT data and construct an efficient communication environment.

People apply UAVs in various fields due to their diversity, flexibility, and reduced cost [10]. Moreover, individuals and organizations use UAVs for prompt service recovery after disasters, data collection, and transmission for IoT networks due to their high mobility and ease of deployment [11]. However, since data are important resources in urban computing, data transmission is impossible if data communication fails to occur, which leads to significant damage. In this case, using UAVs with high mobility at the data communication failure area can supplement the existing system or increase the connectivity among devices. Through this, it is possible to minimize the damage in the communication failure area. Therefore, we increase communication efficiency in urban computing by exploiting UAVs.

This paper places UAVs among IoTs and the cloud to protect IoT data and efficiently manage the group. UAVs exist at the group management level, and they form a blockchain. The UAVs determine the density and movement of IoT devices in subgroups by collecting information on the movement path of IoT devices and group key changes. In dense groups of nodes, UAVs move in to mitigate data traffic. Two groups are combined to reduce group key rekeying overhead when many IoT devices move in and out between two adjacent groups [12]. The base station (BS) and IoT form a subgroup, and BSs generate another blockchain to store IoT device information for secure node authorization. Since privacy is paramount in IoT, information-sharing about IoT is only done among BSs. Furthermore, there is enhanced security and privacy since highly mobile and hijackable UAVs do not know the information of IoTs [13]. Consequently, only safe IoT can participate in group communication, and UAVs manage the groups to establish an efficient communication environment.

We can summarize the main contributions of this paper as follows.

- We analyzed the problems of the urban computing environment. A transmission delay in a group with heavy traffic, following IoT movement and privacy issues, is critical.
- We configured an experimental environment for efficient communication in urban computing and proposed a group and group key management mechanism utilizing UAVs to increase communication efficiency.
- We offer a hierarchical blockchain for group and group key management. Furthermore, by separating the blockchain into two-layered chains, we strengthen the IoT device information’s privacy and security.
- We simulated our proposal and calculated the overhead for group key rekeying. Our proposal increases the system performance by utilizing the secondary gateway and reducing the group key rekeying overhead, respectively.

We organized the remainder of this paper as follows. Section II introduces the related work on urban computing, group key management, and blockchain. Then, we describe our system model, threat model, and system requirements in Section III. Section IV describes our proposal’s overall structure, as well as the group and group key management scheme. Section V shows the simulation results and the analysis. Finally, Section VI concludes by discussing our implications and future work.

II. RELATED WORK

In this section, we explain blockchain, urban computing, and group key management, the background of the proposed system.

A. BLOCKCHAIN

Blockchain is one of the core technologies of ICT (Information & Communication Technology) and is attracting atten-
tion as next-generation network technology. Blockchain is a distributed database with no central server. In the blockchain, data are called a ‘block’. Each block is connected and stored in a chain form through a hash value. Blocks in the blockchain are composed of block headers and block bodies [14]. The block header stores information about the block, version, Merkle root, and creation time, and the block body stores transactions. Everyone can check transaction information through the transaction in the block body [15], [16]. In addition, participants in the blockchain verify and store all transaction contents. Therefore, when a malicious user wants to modify transaction contents, it should modify the ledger of all participants to modify one transaction. Since this is practically impossible, transaction modification attacks are unlikely to succeed. Due to these characteristics, the blockchain guarantees transparency and integrity.

However, because all participants share all information recorded in a block in a blockchain network, if there is a large amount of data attached, it must be replicated to all nodes, leading to network overhead [17]. In addition, there is also limited capacity for information upload to a blockchain. In particular, when a transaction records and identifies the movement path of frequently moving IoTs, many transactions occur, and network overhead may increase due to information sharing with several nodes.

Blockchain also has privacy issues. The issues in information technology (IT) systems are also an important element of privacy requirements in Europe’s recently enacted general data protection regulation (GDPR) [18], [19]. The blockchain’s transparency is a designated weakness in terms of privacy. It is impossible to guarantee data deletion because the blocks are chained and distributed. In some cases, the UAVs could be in a ‘NoCon (No control)’ situation that is out of control due to communication distance or other abnormalities. If they cannot return to their designated places, they could be lost or hijacked [20], [21]. In the worst case, a malicious user could steal the lost UAVs with IoT information. In our proposal, we construct hierarchical blockchains to solve these blockchain problems.

B. URBAN COMPUTING

Urban computing collects and analyzes data generated within a city to solve the city’s major issues [22], [23]. After correctly classifying and reprocessing IoT data, the user receives information at the required time [24]. Each IoT device generates various types of data and forms a cyclic structure where other IoTs use the data for learning [25]. In urban computing, where various devices are networked and share various kinds of information, data security is essential, and there must be a guarantee of integrity that verifies that no data tampering has occurred. When utilizing real-time data, data integrity is undoubtedly an important factor, so it is essential to trace the flow of collected data.

However, the traditional system has some problems. Traditional urban computing continuously collects data from all devices. A database stores these data suitable for each characteristic [1]. Since the database integrates and manages the data, it is inefficient to trace each data flow. In addition, private data can leak in data acquisition and transmission, and a malicious attacker can hijack data [26]. The database returns the query results, and a malicious attacker can use the query result to obtain the data from a specific person. We consider group communication in urban computing to protect data privacy and increase data utilization. In group communication, the density of nodes within a corresponding group refers to the density of data generated by each node. When the number of BSs is limited, problems such as data omission can occur due to data density. These problems are fatal for user convenience in urban computing, so improving the communication environment is very important. Our proposal establishes an efficient group communication environment by determining the movement of IoT and the density within the group and managing the appropriate group and group keys.

C. GROUP KEY MANAGEMENT

Group communication is required to effectively perform functions such as data sharing and software updates in IoT systems. Therefore, we need an efficient group key management mechanism because one-to-one communication is inefficient due to IoT’s low power and capacity [27]. A group key means a key shared by all nodes belonging to the same group for communication. A stolen key from one group could expose all group information. Therefore, group key management is a significant factor in group communication. In particular, according to node movement, group key management must satisfy forward and backward secrecy [28].

1) Forward secrecy: When a node leaves a group, it will not have any access to the future key.
2) Backward secrecy: When a node newly joins a group, it will not have access to the previous information in the group.

In group communication, when a group member is changed due to a node joining or leaving, the group key should be changed for group security. When a node leaves a group, the group key should be rekeyed because the leaving node should no longer be able to decrypt the group’s communication. Additionally, when a node joins a group, the group key should be updated because the newly joining node must not decrypt the previous communication of the group [29].

Research on group key management in the IoT environment is ongoing [30], [31]. These studies use the logical key hierarchy (LKH) key management method. LKH stores encryption keys in a binary tree structure, which reduces communication costs by linking multiple encryption keys with each user [32]. The leaf node contains the users, and all users share the group key (GK) in the root node. The user also has a key-encryption key (KEK) in the path between itself and the root. A KEK is a key that encrypts a GK. Each user’s KEK is encrypted with the user’s shared secret key (SSK) and then distributed. After that, the GK is encrypted with the
Cities have specific security issues, such as data theft and vulnerabilities that various cyber attackers can exploit. In group communication of IoT systems, there are security threats. The subsections define threat models and system requirements. Our system model, threat model, and system requirements. Our previous work [12], proposed a group combination mechanism partly applied in this work for system performance improvement.

III. SYSTEM MODEL

In this paper, we propose an efficient group and group key management scheme using UAVs in the dynamic IoT system. Before presenting our proposal, we introduce the system model, threat model, and system requirements. Our system is composed of a global group and a local group. The global group covers the areas managed by a UAV, while the local group covers the areas controlled by a BS. IoT devices in the local group generate data through sensors and share it with group members. BSs send IoT data to the cloud since IoT has limited computing, storage, and energy resources. Our system assumed that the BS has sufficient computation and storage resources and is reliable. The UAVs of the global group manage the group. We assume that UAVs are unreliable because of their susceptibility to hijacking.

Urban computing is a dynamic IoT system where IoT device numbers in a group frequently change over time. That is, IoT devices can join or leave a group at any time. There are several threats in this dynamic system, and group communication should proceed while protecting IoT data from these threats. The subsections define threat models and system requirements that can occur in a dynamic environment.

A. THREAT MODEL

In group communication of IoT systems, there are security vulnerabilities that various cyber attackers can exploit. In particular, urban systems such as smart homes and smart cities have specific security issues, such as data theft and privacy issues. Possible security threats are as follows.

- Unauthorized access: Unauthorized access refers to theft of the credentials of legitimate accounts or illegal utilization of the resource. Effective authentication and access control mechanisms can prevent this attack. Unauthorized access target the confidentiality and authenticity of the system.

- Malicious insiders: Malicious insiders mean that system members maliciously attack the system or leak data. Insider attackers want access to resources they are no longer authorized to access. To accomplish this, they often cooperate with other members. This attack can affect all of the security requirements.

- Data thefts: Urban computing processes a lot of data and sensitive data. In particular, data in transit is even more vulnerable to attacks than data at rest. Data encryption, efficient authentication, and privacy mechanisms prevent this threat. Data thefts target the confidentiality.

- Service interruption attacks: Service interruption attacks mean that legitimate users cannot use the service by artificially making the servers or network too busy to respond. Service interruption attacks target availability.

B. SYSTEM REQUIREMENTS

We analyze system requirements for group communication in urban computing. In general, the system for secure group communication should meet the following requirements [9], [31].

- Security requirement: The system performs encryption for data security in group communication. In particular, according to node movement, group key management must satisfy forward and backward secrecy. Forward secrecy is achieved by ensuring that the leaving node can not decrypt the future exchanged message. And backward secrecy is achieved by ensuring that the joined new node cannot decrypt the previous communication.

- Efficiency requirement: The efficiency of the group communication system is justified by minimum overhead cost [31]. In group communication, when a group member is changed, the group key should be changed. If group key updates are frequent, computation and communication overhead increase. Therefore, the system should minimize overhead cost by reducing the group key update frequency for improved efficiency.

- Performance requirement: In group communication, the rate of change in group membership affects performance. Frequent membership changes reduce system availability. Therefore, the system should improve performance by minimizing membership change for system availability.
A. SYSTEM DESIGN

We propose a hierarchical blockchain-based system for an efficient communication environment in urban computing. Figure 1 shows the overview of the proposed system. The system consists of three layers. First, the IoT Network Layer, the lowest layer, consists of IoT devices and BSs. They are geographically divided and form local groups based on the BSs. The next layer is the UAV layer. The UAV is a member of the UAV layer and manages the global group. Further, it forms a global group with local groups and communicates with the BSs within the same group. Finally, the top layer, the Cloud Layer, communicates with the BSs under the same group and stores all the data. In particular, since our proposed system stores all of the blockchains, data tracking is possible to ensure data integrity. Table 1 shows the roles and descriptions of the components of our proposal.

For an efficient communication environment in urban computing, our proposal operates the system considering the density and mobility of the IoT devices in each group. The experimental environment of groups configured in the system is as follows.

- High density and high movement
- High density and low movement
- Low density and high movement
- Low density and low movement

When the density of IoT devices is high, there is concentrated data traffic, and data loss and delay problems may occur. In our proposal, another UAV belonging to another group could move in to act as a secondary gateway. In addition, when IoT movement is high in a regional group, the overhead for group key updates increases. In this case, the adjacent two groups are temporarily combined to decrease group key rekeying overhead.

B. HIERARCHICAL BLOCKCHAIN SYSTEM

Our proposal divides blockchain into a global chain shared by the UAVs and a local chain shared by the BSs under the same global group. Since UAVs are highly mobile and susceptible to hijacking, UAV records only group key update information on the global chain in this system. In the global chain, UAVs do not know the information and data of IoT devices. This approach enables smooth group management while ensuring the privacy of IoT devices. In addition, since the local group’s group keys are unrevealed, security against key theft can be guaranteed. Figure 2 shows the configuration of the global chain among the UAVs, and Figure 3 shows the configuration of the local chain shared by the BSs belonging to the same global group. In the next section, we take a closer look at each blockchain.

1) Global Chain Among the UAVs

The global chain among the UAVs is a blockchain for seamless data communication by determining the IoT movement and group density. Each UAV creates a transaction whenever the group key requires updating in the local group, which is a subgroup. The transaction includes the group key update information, group ID, and IoT movement path. Table 2 shows the transaction information of the global chain.

When the group key is updated in the local group, that group’s BS transmits group key update information to the UAV. Then the UAV creates a transaction based on the information received from the BS. In this case, two groups (source group and destination group) perform group key rekeying due to IoT movement. Between the two groups, the BS of the destination group transmits information to the UAV. This transmission prevents transaction duplication and records movement paths (source and destination) in one transaction. To record the movement path of IoT, the destination BS transmits the ID of the source BS (Sour_BS_ID) to the UAV. The continuous confirmation of two specific BSs in transactions means that many IoT devices move between the two groups. Thus, we can determine the movement of IoT devices between the two groups by confirming the transactions in the
FIGURE 1. The overall structure of the proposed system consists of three layers. Each layer has a separate blockchain. Nodes in the blockchain have ledgers for each blockchain.

FIGURE 2. Global chain: UAVs are global chain nodes.

FIGURE 3. Local chain: BSs in the same global group are nodes in the local chain.

global chain. Moreover, the frequent occurrence of the same destination BS in transactions means that many IoT devices are incoming to that group. Thus, we can also determine the density of a group by confirming the transactions in the global chain.

2) Local Chain Among the BSs
The local chain of BSs belonging to the same global group is a blockchain for secure devices authorization that checks whether an IoT device behaves abnormally. Each BS creates a transaction whenever group members change. A local chain transaction records the group’s IoT devices and the ID of an IoT that has behaved abnormally. Table 3 shows the transaction information of the local chain. The BS stores the current group member list, including the moved IoT, as a transaction. If an IoT in the group behaves abnormally, the system records the IoT’s ID, and when it moves to another group, it will be considered unauthorized in the new group.

C. GROUP AND GROUP KEY MANAGEMENT SCHEME
TABLE 3. Transaction components of a local chain of BS belonging to the same global group.

| Field     | Description                                                                 |
|-----------|-----------------------------------------------------------------------------|
| Time      | Transaction occurrence time                                                  |
| Dest_BSID | BS ID of destination in IoT movement path and transaction occurrence subject |
| Sour_BSID | BS ID of source in IoT movement path                                         |
| GroupID   | ID of local group to which BS belongs                                        |
| IoTList   | List of group members belonging to the local group                           |
| AbnormalIoTID | ID of IoT that has behaved abnormally                                      |

1) Group Management

Our system is composed of a global group and a local group for efficient group management. The global group covers the areas managed by a UAV, and the local group covers the areas controlled by a BS. Our system determines the IoT movement and group density by confirming the blockchain ledger of each layer. When a group is overwhelmed by heavy traffic, the UAV of a group with relatively few transactions moves in. It acts as a traffic mitigation agent by acting as a secondary gateway. Another UAV in the geographically close group temporarily controls the group from which the UAV moves out.

Figure 4 shows a scenario for UAV movement. If heavy data traffic occurs in a group, the UAV of group A moves to relieve the overload in the heavy traffic area. Thereafter, the UAV of group B, geographically closer to the corresponding local group (Group A_1 to Group A_3), temporarily processes the transactions of group A and group B. Selecting the proper UAV and handing over control is not within the scope of our work.

2) Group Key Management

When a large group uses the same communication key, and if a malicious attacker steals one group’s key, then the malicious attacker has access to the information of many BSs and subordinate IoT devices, resulting in a serious security problem. Therefore, group communication occurs in this system’s smaller local group unit. In our work, we partly adopted the group key mechanism of [30]. Table 4 summarizes the notations used in the group key mechanism.

Figure 5 shows the key management structure according to the scenario. There are three nodes in the initial stage (Figure 5 (a)). When adding a new node N_4 (Figure 5 (b)), the BS gets N_4’s SSK and updates the GK and KEK along the path from N_4 to the root. The BS encrypts and transmits the KEK using the SSK of N_4. After that, the newly updated GK’ is encrypted with KEK_{34} and transmitted to N_4. At this time, the existing nodes hash the GK to generate GK’ and update the group key as a whole. Existing nodes can reduce communication and computation costs by updating the group key through a simple hash without receiving GK’ from the BS. Figure 5 (c) shows that N_2 leaves the group after adding N_4. When a node leaves the group, and if it changes the group key by hashing, the leaving node also knows the corresponding value. In this case, the group key should not compute the group key by hashing. Instead, the BS generates GK’”, encrypts it with KEK, and transmits it to each node.

This method has the advantage of using a hash function to lower the overhead. However, when group membership
keeps changing, this mechanism is also inefficient. Since many IoT devices have high mobility in urban computing, a mechanism that considers movement is necessary. Therefore, we propose a mechanism for efficient group key management in the environment for a high movement of IoT.

Algorithm 1 Key update when Device $i$ in Group A_1 leave

1. BS $\xrightarrow{broadcast}$ All: "Leaving device $i$ is no longer available"
2. BS $\xrightarrow{broadcast}$ IoTs in group A_1: GK’ and KEK’s update.
3. BS $\xrightarrow{Local}$ chain: Transaction generation.
4. BS $\xrightarrow{unicast}$ UAV_A: Key update information.
5. UAV $\xrightarrow{Global}$ chain: Transaction generation.

Figure 6 shows the process for changing the group key when a leaving node. The BS verifies information about the node by querying the blockchain. If the record confirms that node $i$ came from the previous group, it sets an accept message and the SSK. Then, the group key change message is sent to the existing nodes in group A_1, and the GK is updated. Similarly, existing nodes hash the old GK to create a new GK’.

Algorithm 2 Key update when Device $i$ joins Group A_1

1. BS $\xrightarrow{unicast}$ node $i$: SSK establishing with new node $i$.
2. BS $\xrightarrow{broadcast}$ IoTs in Group A_1: Group key updates,
   $GK' = H(GK)$.
3. BS $\xrightarrow{unicast}$ nodes: KEK’ encrypted by either SSK or shared KEK.
4. BS $\xrightarrow{Local}$ chain: Transaction generation.
5. BS $\xrightarrow{unicast}$ UAV_A: Key update information.
6. UAV $\xrightarrow{Global}$ chain: Transaction generation.

Algorithm 3 Key update when group A_1 and group A_2 combine

1. UAV $\xrightarrow{multicast}$ BSs: Group combination message.
2. UAV $\xrightarrow{multicast}$ BSs: Random seed ($x$) for GK’ updates.
3. UAV $\xrightarrow{multicast}$ BSs: "Compute GK’ via $x$ then GK’ update in each group."
4. BS $\xrightarrow{Local}$ chain: Transaction generation.
5. UAV $\xrightarrow{Global}$ chain: Transaction generation.

Figure 7 shows the process for changing the group key when a member is changed due to node joining. The BS verifies information about the node by querying the blockchain. If the record confirms that node $i$ came from the previous group, it sets an accept message and the SSK. Then, the group key change message is sent to the existing nodes in group A_1, and the GK is updated. Similarly, existing nodes hash the old GK to create a new GK’.

Figure 8 shows the process for changing the group key when combining two groups with highly mobile IoTs. The UAV designates two groups whose group key is frequently changed due to frequent mobility through the global chain.
and sends an integration request. When the BSs of both groups respond to the integration request, the UA V transmits the seed value for generating the same group key. The BS creates a transaction stating that the group has been combined and the group key has been changed, records it in the local chain, and the UA V records the information that the two groups are combined in the global chain. Each BS in the integrated group creates a list of nodes communicating with itself for a certain period as a transaction and informs the UA V. At this time, the UA V divides the group if there is not much change in the node list with which each BS in the combined group communicates. Then each BS generates a random seed and regenerates the group key. Algorithm 3 expresses this.

After combining two groups, if node movement becomes low again for a certain time, they could divide the groups into two again. We can track node movement through blockchain records. In our work, we considered two adjacent group combinations. We will research multiple group combinations and divisions in our future work.

V. EXPERIMENTAL RESULTS AND PERFORMANCE ANALYSIS

A. EXPERIMENTAL SETTINGS

For experimentation, we used a virtual machine equipped with an Intel Core i7-10700 64-bit CPUs@2.90 GHz, 24GB of RAM, and 200GB hard drive space on Ubuntu 20.04.2 LTS. We also used ns-3 version 3.32 for group management simulation in urban computing [36]. To evaluate the computational overhead of the group key mechanism and measure the performance of the blockchain system, we employed the miracl core library and the open-source version of Hyperledger caliper 0.3, respectively [37], [38]. Miracl core supports encryption and hash functions and is a multi-lingual encryption library independent of architecture. Hyperledger caliper benchmarks blockchains to measure performance and provides results in the form of reports. We adopted these simulators to analyze the effectiveness of group and group key management. First, we established a communication environment using ns-3 to analyze the efficiency of group management. Table 5 shows the environment configurations.

We performed the simulation by increasing the number of nodes. Then, we simulated the encryption operation using the miracl core library to analyze the group key calculation overhead. Consequently, AES-256 of a 64-byte block takes 133.6 $\mu$s, and SHA-256 takes 103.8 $\mu$s. In addition, using Hyperledger Caliper, we measured the time for transaction creation and ledger query. Based on 100 nodes, the transaction creation time was 13,000 $\mu$s, and the query time was 200 $\mu$s. We accomplished symmetric key encryption using AES-256 and the hash function using SHA-256. We calculated the computation overhead for changing the group key based on these results.

We compared our proposed system with existing works. In existing works, the systems only communicate by group and do not determine node mobility or group density. On the other hand, our proposal considers node mobility and group density. Our system combines groups according to node mobility or adds a secondary gateway according to group density.

B. EFFICIENT AND PERFORMANCE ANALYSIS

1) Group Management Analysis

We set a threshold value for group density to analyze the efficiency of group management.

Figure 9 shows the throughput and packet loss rate according to the density of each group. The higher the density
FIGURE 10. Comparison of the packet loss rates of the existing works and the proposed system. (secondary gateways added)

FIGURE 11. Comparison of the packet throughput of the existing works and the proposed system. (secondary gateways added)

FIGURE 12. Comparison of the End-to-End (E2E) delays of the existing works and the proposed system. (secondary gateways added)

FIGURE 13. Comparison of the jitter of the E2E delays of the existing works and the proposed system. (secondary gateways added)

in the group, the higher the packet loss. The maximum throughput occurs when the group density is 80, and the throughput decreases thereafter. As the number of nodes in a group increases, this generates more data and the resource to process the data becomes insufficient. Therefore, we set the threshold density to 80 in our simulation. Figure 10 shows the packet loss according to the node density in the group. The packet-loss rate rapidly increases in the existing works as the node density becomes higher than 80. However, in our proposal, because we deploy the secondary gateway at a density of 80, the packet-loss proportion decreases as the secondary gateway processes more data.

Figure 11 shows the throughput according to the density of a group. As in the packet loss, throughput decreases from a density of 90. However, as the secondary gateway starts processing, the throughput increases. Figure 12 shows the average End-to-End (E2E) delay. The E2E delay is the time for a packet to transmit through the network from source to destination. Since the proposed system mitigates traffic through a secondary gateway when the density gets higher, E2E delay also decreases. Figure 13 shows the jitter of the E2E delay. Jitter delay refers to rapid variations in the delay at which a packet travels through a network [39]. It shows a deviation indicating how constant the delay is until the packet arrives at the destination. Even if the E2E delay is short, system performance degradation can occur if the fluctuation is severe. As a result of the simulation, the jitter value of the existing system gradually increases. Still, in the proposed system, the jitter value decreases after deploying the secondary gateway.

Figure 14 shows the useful traffic rate (UTR). Useful traffic is a performance indicator that shows the meaningful traffic among the network layers’ packets. UTR indicates the amount of data transmitted among the traffic from the source. The higher the value, the better the performance [40]. The useful traffic rate decreases as the group density increases.
in the existing works. However, in our proposal, the useful traffic decreases and then increases again with the processing of more data without loss due to the deployed secondary gateway.

2) Group Key Management Analysis

Our proposal temporarily combines two groups with highly mobile IoTs to reduce the group key update overhead. We assume two adjacent groups, \( A_1 \) and \( A_2 \), and 100 nodes in each group for the simulation. We did not consider membership withdrawal in this work. We compared the communication and computation overhead for the individual groups \( A_1 \) and \( A_2 \), as well as the combined ones.

We used the LKH structure for group key rekeying, which reduces communication costs by associating multiple KEKs with each node. The update communication for KEKs requires \( \log(n) \) multicasts for \( n \) nodes. Figure 15 shows the communication overhead according to group membership changes. Many rekeying messages are caused in individual groups as group membership changes increase. The reason that group \( A_2 \) has a higher overhead than \( A_1 \) is because of the movement assumption. In this assumption, nodes move from group \( A_2 \) to \( A_1 \), which means that nodes leave group \( A_2 \) to join group \( A_1 \). The BS of group \( A_1 \) broadcasts GK rekeying messages to the existing members and unicasts key information to the new node. The BS of group \( A_2 \) broadcasts GK rekeying messages and \( \log(n) \) multicasts to existing members for KEK rekeying. Thus, the communication overhead in group \( A_2 \) is higher. However, the average overhead of the two groups is closer to that of group \( A_1 \) because group \( A_1 \), with its many nodes, requires fewer rekeying messages. In the case of group combination, we require additional communication among UAVs. However, despite the additional UAV communication overhead, our proposal has low overhead when group membership changes are frequent.

For computational overhead, we assumed that the average number of nodes in each group is maintained even after the nodes move between two adjacent groups. This assumption is important because the number of nodes in each group affects the computational overhead. Group key rekeying for forward secrecy has a higher overhead than it does for backward secrecy because rekeying for backward secrecy uses a hash function to reduce computation. In addition, when a node joins a group, a blockchain query must check the node’s authorization. Finally, the local chain for a group combination should record the combination transaction. Therefore, we considered blockchain query time and transaction creation time. Figure 16 shows the computational overhead according to group combination. Group \( A_2 \) has higher computational overhead than \( A_1 \) because group \( A_2 \) node inflow is accomplished ahead of node outflow in our simulation. A new group key generates when two adjacent groups combine, and the blockchain records this transaction. Similar to communi-
cation overhead, computation overhead is much lower when group membership changes are frequent.

In our simulation, when considering communication and computation overhead, group combination is efficient when the number of group membership changes is greater than 30. After the group combination, when the node movement ratio decreases for a certain time, the groups need to be separated again for better performance. Our research scope did not include group separation. We can expect system performance enhancement by proper group combination and separation based on group member movements. In the system design for group communication, we set up the groups properly for group management. Nevertheless, combining groups is more efficient when node mobility is high between two adjacent groups.

C. SECURITY ANALYSIS

Our proposal considers the group key update when nodes join and leave the system to ensure forward and backward secrecy. Additionally, we analyze security in terms of both information security requirements and IoT security threats.

1) Security analysis in terms of information security requirements

With the recent development of ICT technology, the information security requirements include additional items as well as the general security objectives (CIA, confidentiality, integrity, availability). The information system must satisfy confidentiality, integrity, availability, non-repudiation, accountability, and authenticity for ICT security [41].

- **Confidentiality**: Confidentiality means providing information only to authorized users. In our system, the nodes join a group after authentication. Only group members composed of authenticated nodes can communicate through a group key. Therefore, our proposal ensures confidentiality.

- **Integrity**: Integrity means that information is not changed. Our system records abnormal IoT information and IoT movement on the blockchain. Because node information is created as transactions are stored in a chain, forgery is impossible. Abnormal nodes maliciously try to join the group by forging previous records, but blockchain can prevent this. Therefore, our proposal ensures integrity.

- **Availability**: Availability means that protect timely and uninterrupted access to the system. Our system combines groups when group membership frequently changes due to node movement between two adjacent groups. When the groups are combined, each group reduces the frequency of group key updates because both groups use the same key. Because frequent group key updates waste resources, group combinations can avoid wasting resources. Therefore, our proposal ensures availability.

- **Non-repudiation**: Non-repudiation means preventing the denial of responsibility among the actual actions. Our system records the movement path and any abnormal behaviors of nodes on the blockchain. Because blockchain guarantees transparency, nodes cannot deny previous records. Therefore, our proposal ensures non-repudiation.

- **Accountability**: Accountability means tracking the user’s behavior. In our system, blockchain records all information, and the cloud stores all of the blockchains built on the system. Therefore, our proposal ensures accountability.

- **Authenticity**: Our system identifies malicious nodes by using abnormal node information recorded in the local chain and authorizes the abnormal node. Therefore, our proposal ensures authenticity.

2) Security analysis in terms of IoT security threats

We analyze security in terms of IoT security threats mentioned in section III.

- **Unauthorized access**: Our system determines the abnormal IoT through the local chain. As abnormal IoT cannot join the group, our system defends against unauthorized access threats.

- **Malicious insiders**: Our system ensures forward and backward secrecy, so nodes that are not group members cannot access the data in the group. In addition, if a node behaves abnormally in group communication, that node information is recorded in the local chain to prevent participation in other groups in the future. Therefore, our system can prevent a more significant risk by malicious insiders.

- **Data thefts**: Because an attacker can infer the sensitive information of the device owner through IoT data, data theft can be a serious privacy problem. Our system processes IoT data through a trusted BS. And we designed the system made untrusted UAVs cannot know IoT data. Therefore, our system can prevent the risk of data theft.

- **Service interruption attacks**: Our system reduces the probability of this attack because the system denies authorization of abnormal nodes through the blockchain. In addition, the cloud store IoT data and all blockchains in which IoT information (node movement path, abnormal IoT information). Even if service interruption attacks occur, users can continue to receive services through the cloud. Therefore, our system can reduce the risk of service interruption attacks.

VI. CONCLUSION

The importance of data is increasing in urban computing with a cyclic structure in which other intelligent devices use data generated by all objects for learning. In particular, when using real-time data, data accuracy is required to ensure data integrity. However, nodes crowded in an environment where all nodes generate data can lead to data loss and transmission delay problems, leading to performance degradation. In addition, group communication is also important for efficient communication of IoT devices, including up-
dating group keys according to device movement. However, updating group keys too frequently could decrease system performance. However, one could solve these problems and increase system performance by properly checking node movement and group density.

We proposed a hierarchical blockchain and UAV-based group and group key management system for efficient urban computing with highly mobile IoT nodes. When the node density in a group is high, another UAV moves into the area and alleviates the traffic as a secondary gateway. For efficient group management, when two adjacent groups have highly mobile IoT devices, which causes group key updates, two groups temporarily combine to decrease the group key update overhead. In addition, two layered blockchains ensure the privacy of the IoT devices. Even if malicious attackers hijack UAVs, these malicious attackers cannot access the IoT nodes’ information since the UAVs do not have this information. The high-layer blockchain among the UAVs only manages the information about group density and movement of IoTs while the low layer blockchain among the BSs manages abnormal IoT information. We simulated our proposal in various environments and showed that our system improves urban computing system performance. Our proposal is expected to be applied not only to urban computing but also to the highly mobile vehicular ad-hoc network (VANET). We will consider combining and separating more diverse groups in different environments in our future work.

REFERENCES

[1] Y. Zheng, L. Capra, O. Wolfdorf, and H. Yang, “Urban computing: Concepts, methodologies, and applications,” ACM Trans. Intell. Syst. Technol., vol. 5, no. 3, pp. 38–55, Sep. 2014.

[2] M. M. Rashid, J. Kamruzzaman, M. M. Hassan, S. Shahriar Shafin, and M. Z. A. Bhuian, “A survey on behavioral pattern mining from sensor data in Internet of Things,” IEEE Access, vol. 8, pp. 33318–33341, Feb. 2020.

[3] K. Su, J. Li, and H. Fu, “Smart city and the applications,” in Proc. Int. Conf. Electron., Commun. Control, Sep. 2011, pp. 1028–1031.

[4] S. Cha, T. Hsu, Y. Xiang, and K. Yeh, “Privacy enhancing technologies in the Internet of Things: Perspectives and challenges,” IEEE Internet Things J., vol. 6, no. 2, pp. 2159–2187, Apr. 2019.

[5] S. S. Albouq, A. A. A. Sen, A. Mamoun, N. M. Bahbouh, A. B. Alkhodre, and A. A. Alshamiti, “A double obfuscation approach for protecting the privacy of IoT location based applications,” IEEE Access, vol. 8, pp. 129415–129431, Jul. 2020.

[6] Y. Zheng, W. Wu, Y. Chen, H. Qu, and L. M. Ni, “Visual analytics in urban computing: An overview,” IEEE Trans. Big Data, vol. 2, no. 3, pp. 276–296, Sep. 2016.

[7] D. Wang, J. Zhao, and Y. Wang, “A Survey on Privacy Protection of Blockchain: The Technology and Application,” IEEE ACCESS, vol. 8, pp. 108766–108781, May. 2020.

[8] M. A. Kandi, H. Lakhlief, A. Bouabdallah and Y. Challal, “A versatile Key Management protocol for secure Group and Device-to-Device Communication in the Internet of Things,” J. Netw. Comput. Appl., vol. 150, pp. 102480, Jan. 2020.

[9] O. Cheikhrouhou, “Secure group communication in wireless sensor networks: A survey,” J. Netw. Comput. Appl., vol. 61, pp. 115–132, Feb. 2016.

[10] O. S. Oubbati, M. Atiquzzaman, P. Lorenz, M. H. Tareque, and M. S. Hussain, “Routing in flying ad hoc networks: Survey, constraints, and future challenge perspectives,” IEEE Access, vol. 7, pp. 81057–81105, Jun. 2019.

[11] D. S. Lakey, U. Sa’ad, N.-N. Dao, W. Na, and S. Cho, “Routing in flying ad hoc networks: A comprehensive survey,” IEEE Commun. Surveys Tuts., vol. 22, no. 2, pp. 1071–1120, 2nd Quart., 2020.
[37] **MIRACL core.** Accessed: Jun. 1, 2021. [Online]. Available: https://github.com/miracl/core

[38] N. Lincoln, *A blockchain performance benchmark framework Hyperledger Caliper*. Accessed: Jun. 1, 2021. [Online]. Available: https://hyperledger.github.io/caliper/

[39] M. Yuan, “Jitter buffer control algorithm and simulation based on network traffic prediction,” *Int. J. Wireless Inf. Netw.*, vol. 26, pp. 133-142, Sep. 2019.

[40] M. Z. Malnar and N. J. Jevic, “Novel ETX-based metrics for overhead reduction in dynamic ad hoc networks,” *IEEE Access*, vol. 7, pp. 116490-116504, Aug. 2019.

[41] R. Von Solms and J. Van Niekerk “From information security to cyber security,” *Comput. Secur.*, vol. 38, pp. 97-102, Oct. 2013.

---

**GABIN HEO** received her B.S. degree in the Department of Cyber Security at Kyungil University in 2018, and M.S. degree in the Computer Science and Engineering at Ewha Womans University in 2020. She is currently pursuing the Ph.D. degree in the Division of Artificial Intelligence and Software at Ewha Womans University, Seoul, Korea. Her research interests include blockchain, authentication, digital content security, privacy, and IoT security.

---

**KJOON CHAE** received the B.S. degree in mathematics from Yonsei University in 1982, an M.S. degree in computer science from Syracuse University in 1984. He received a Ph.D. degree in electrical and computer engineering from North Carolina State University in 1990. He is currently a professor in the Department of Computer Science and Engineering at Ewha Womans University, Seoul, Korea. His research interests include blockchain, security of FANET (Flying Ad Hoc Networks), sensor network, smart grid, CDN, SDN and IoT, network protocol design and performance evaluation.

---

**INSHIL DOH** received her B.S. and M.S. in Computer Science at Ewha Womans University, Korea, in 1993 and 1995, respectively. She received her Ph.D. degree in Computer Science and Engineering from Ewha Womans University in 2007. From 1995 to 1998, she worked at Samsung SDS of Korea. She was a research professor at Ewha Womans University in 2009, 2010, and at Sungkyunkwan University in 2011. She is currently an associate professor of the Department of Cyber Security at Ewha Womans University, Seoul, Korea. Her research interests include wireless network security, sensor network security, and IoT network security.

---

***