IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 14 July 2022, accepted 8 August 2022, date of publication 18 August 2022, date of current version 7 September 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3199880

RESEARCH ARTICLE

Distributed Ledger Technology Based Architecture for Decentralized Device-to-Device Communication Network

SHU-PING LU[1], CHIN-LAUNG LEI[1], CHENG-YUN HO[2], SHY-SHANG HWANG[2], AND HSIN-CHEN CHEN[3]

[1]Department of Electrical Engineering, National Taiwan University, Taipei City 10617, Taiwan
[2]Industrial Technology Research Institute, Hsinchu 310401, Taiwan
[3]Phison Electronics Corporation, Miaoli County 350402, Taiwan

Corresponding author: Shu-Ping Lu (d03921013@ntu.edu.tw)

This work was supported in part by the Ministry of Economic Affairs of Taiwan (MOEA) under Grant 107-EC-17-A-24-1520, Grant 109-EC-17-A-24-1520, and Grant 105-EC-17-A-03-I1-0002.

ABSTRACT Due to the mobility of devices, device-to-device (D2D) communication is a promising fifth-generation (5G) technology in dynamic environments for improving message transmission efficiency for group communication. Additionally, all services in an ad hoc network are current Vehicle Ad Hoc Network (VANET) applications. Therefore, D2D communication has been introduced in ad hoc environments to reduce latency during vehicle conversations, such as autonomous vehicle solutions and drone fleet management for cellular vehicle-to-everything (C-V2X) modules and Internet of Drones (IoDs) networks. However, providing secure and effective group communication is an urgent challenge. To solve these problems, we propose a dynamic group management solution based on distributed ledger technology. This study demonstrates that a distributed ledger-based hierarchical architecture for dynamic group management is faster and more adaptable without compromising security and performance. Furthermore, the proposed method can facilitate the transfer of direct communication data without a centralized database, thereby reducing the chance of a single point of failure. In addition, the research was tested by a third party that has established close cooperation with world-leading automotive electronics suppliers in Taiwan.

INDEX TERMS Device-to-device communications, peer-to-peer (P2P), vehicle-to-everything (V2X), distributed ledger technology, security, cryptography.

I. INTRODUCTION

In the Internet of Things (IoT) era, the fifth generation (5G) network provides a completely mobile and connected society for billions of connected objects [1] [2], [3], [4], [5]. Currently, promising applications in ad hoc networks are increasing. In addition, today's information and computing systems are distributed in nature. For example, intelligent vehicles are experiencing revolutionary growth in research and industry. In terms of road safety, high data transfer and network communication, all services in an ad hoc network are critical for current vehicular ad hoc network (VANET) applications such as a cellular vehicle-to-everything (C-V2X) [6], [7] module in automation solutions. The C-V2X can be used for drone fleet management and autonomous cars. Moreover, ad hoc networks can be organized using D2D communication, which can be used for emergency communications. Therefore, device-to-device (D2D) communication was introduced in the V2X environment to reduce delays during vehicle conversation. However, *dynamic topology and member management* is a major issue in ad hoc network groups. As the number of nodes increases, the dynamic system becomes more complex and requires dynamic management to ensure safety.

Due to the lack of network protection, as the number of ubiquitous devices continues to increase, the network becomes more vulnerable to attacks. Consequently, security is a crucial issue for the rapidly evolving D2D network, which

The associate editor coordinating the review of this manuscript and approving it for publication was Diana Gratiela Berbecaru.

is essential to provide services in a small coverage area [8], [9], [10]. Therefore, the system needs to provide timely and reliable service for dynamic group member management. However, the salient features of ad hoc networks bring challenges in achieving these goals, including *authentication and availability* [11], [12].

Figure 1 shows ad hoc network issues. The connectivity between nodes is inevitably affected by the movement, arrival and departure of nodes. Since the current applications in the ad hoc network are naturally scattered [13], [14], an authenticator is necessary to conduct dynamic and distributed management. However, by using centralized authenticators and fully distributed authenticators, existing methods are being challenged. A centralized authenticator uses static configuration, while the distributed authenticator lacks the efficiency of updating the group key. For example, trusted agencies are responsible for the registration of roadside units (RSU) and vehicles and provide support for necessary communication assistance [15]. In addition, the group key in Vehicular Ad Hoc Network (VANET) cannot be updated correctly [16]. In other words, the traditional security mechanisms used in VANET still has defects [17], [18]. Distributed ledger technology contains specific and verifiable records of all transactions that have been made within a distributed system [19]. However, a fully distributed blockchain authenticator is very slow to update authentication information.

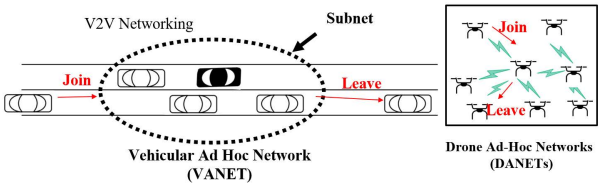


FIGURE 1. Ad hoc network Issue.

To solve the problem of *dynamic topology and dynamic group management*, this research proposes a hybrid and effective security protocol based on distributed ledger technology and edge center architecture. In addition, distributed ledger technology with consensus algorithms enables nodes in ad hoc networks (such as fleet managers) to improve security, convenience and efficiency.

In summary, in view of the limitations of centralized and fully distributed networks, the research proposed a hybrid architecture based on distributed ledger technology to improve security and efficiency. The article makes the following contributions.

- A *distributed ledger based* hierarchical architecture for dynamic group management without compromising security and performance. The dynamic membership extension of *consensus algorithm with dynamic group keys* allows the set of nodes in the group to change over time.
- The proposed method can facilitate the transmission of direct communication data without a centralized database, thereby reducing the chance of a single point of failure.
- The research is carried out in the OMNET++ simulator and implemented in an experimental environment. The experimental results are also tested by a third party.

The remainder of this paper is organized as follows. Section II introduces the problem and related work. Design goals, threat model, network model, security requirements, system model and preliminaries are carried out in Section III. In Section IV, the proposed protocol is conducted and discussed in detail and then evaluated in Section V. In addition, Section VI has some discussion and future directions. Finally, Section VII summarizes this article.

II. PROBLEM DESCRIPTION AND RELATED WORK

In this section, the problem of dynamic group management in wireless ad hoc networks is described, existing research is briefly reviewed, and the differences between our work and existing methods are emphasized.

A. PROBLEM DESCRIPTION

Thus, it is essential to perform the authentication process safely and efficiently. In addition, the IEEE 1609.2 standard defines secure message format and suggests that message processing shall be within a validity period of less than five seconds for authentication credentials [20]. However, ad hoc networks do not have a fast and reliable mechanism to protect group management in dynamic topology and member management. For example, device-to-device (D2D) communication requires effective security solutions for proximity-based direct communication without the assistance of cellular networks [21].

B. RELATED WORK

Blockchain provides access to distributed ledgers in a trusted and secure manners [22], [23]. Therefore, multiple industries and organizations are studying the implementation of blockchain technology for intelligent vehicles. The vehicle network used the blockchain in the self-organizing network to communicate between vehicles [24], [25], [26]. Other studies applied blockchain to provide decentralization, traceability, and immutability in the Internet of Things [27], [28]. Feng *et al.* proposed a blockchain-based distributed collocation storage architecture for data security process platform of Wireless Sensor Network (WSN) [29]. Moreover, several studies have been considered for group key management of connected devices [30], [31], [32]. The group key can be updated for authenticated vehicles when vehicles join and leave the group [33]. However, this requires a third party trusted organization to distribute group keys for all vehicles, and there are security risks of key escrow and the complexity of certificate management. Abd-Elrahman *et al.* [34] proposed an identity-based encryption (IBE) group authentication scheme using multiple private key generators. Nevertheless, this scheme relies on external key servers that are

vulnerable to key escrow attacks. To alleviate the key update overhead, the decentralized architecture consists of a key distribution center and multiple subkey distribution centers for managing IoT groups [35]. Although some blockchain-based key management techniques for VANET have been proposed [25], [36], these methods lack automatic key renewal in dynamic applications [37]. Additionally, the problem of single point of failure remains. Nevertheless, a secure trust environment for D2D communication using dynamic group keys with dynamic consensus mechanism has not been discussed. In addition, traditional fully distributed blockchain authenticators use a slow consensus mechanism to update identity verification information.

Table 1 summarizes existing issues related to centralized and distributed authenticators. Table 2 lists the popular decentralized blockchain consensus protocols. Centralized authenticators use a static configuration, whereas distributed authenticators lack the efficiency to update group keys. However, the traditional security mechanisms used in VANET still have flaws, such as the potential risk of a single point of failure for remote servers [38], [39], [40]. Additionally, fully distributed blockchain authenticators are very slow to update authentication information [41], [42]. To confirm any single transaction in the system, Bitcoin's Proof-of-work (PoW) consensus takes an average of 10 minutes to resolve and requires six consecutive blocks [43]. Because of their high mobility, VANET nodes require shorter security protocol execution times to achieve the same throughput. Therefore, it is more difficult to provide secure network connectivity in a distributed architecture than in a centralized architectures [44], [45], [46]. For example, the high mobility of connected vehicles makes the proof-of-work (PoW) process difficult, as there is limited time for nodes to move to exchange new blocks for verification [47]. In addition, cloud data servers consume a large amount of energy each year in a centralized architecture. Instead, distributed architectures use spare onboard computing resources on connected vehicles to minimize the cloud server energy consumption [45], [48]. In addition, in the ad-hoc world of V2X and IoD (Internet of Drones) networks, cloud servers equipped with powerful computing units may have higher computing power. In contrast, computing power is limited and depends on the CPU frequency of an ad-hoc onboard computer with a fully distributed architecture [45], [49].

C. DIFFERENCES FROM EXISTING WORK

The centralized authenticator uses a static configuration and a fixed network topology requires a complete and continuous network infrastructure (ground control center) deployment. However, the distributed authenticator lacks efficiency in updating the group key. Furthermore, owing to the decentralized nature of edge-centric systems, single-point-of-failure limitations can be avoided. Aiming at the limitations of centralized and fully distributed networks, this study proposes a hybrid architecture based on distributed ledger technology. Hierarchical architecture helps address scalability and efficient resource utilization by reducing the communication load with the central authenticator. Therefore, to meet the requirements of dynamic topology and dynamic member management, this study provides a timely and reliable security protocol based on distributed ledger technology. When the communication network is unstable, distributed ledger technology can be used to ensure that group management works properly locally. For example, when the vehicle passes through the tunnel, the leader and the remote server are disconnected. When the connection between the fleet and the remote server is interrupted through the tunnel, the regional group fleet management can still be maintained and the safety of the fleet can be maintained. This allows the group manager to continue its operations even if it is temporarily isolated from the remote server. In other words, an unreliable network connection won't affect the work of the group manager, which can then work locally. Additionally, the local server periodically attempts to connect to the remote server. A network connection can be established to send periodic reports containing group-related information for all logged events.

The proposed method has to trade off scalability and performance efficiency to account for high throughput or security across a large number of nodes. In other words, a fully distributed blockchain system creates a larger computational and storage burden. This is because all parties need to come to a consensus and store the transaction. Therefore, we propose a hierarchical decentralized architecture for dynamic group management to reduce overhead. In addition, participants must register their identities on the local server/cloud server in advance to ensure a certain degree of security by supplementing node identity management.

TABLE 1. Existing issues for centralized and distributed authenticators.

Network Type	Centralized	Fully Distributed
Weakness	The potential risk of a single point of failure	Updating ad hoc network authentication info. is slow and difficult: 10 min
Strength	Updating ad hoc network authentication info. :$10\mu \sim 600ms$	There is no single point of failure for the authenticator
Mobility	High	Low
Energy Efficiency	Low	High
Computation Capability	High	Low
References	[38]–[40]	[41]–[43], [50]

TABLE 2. Comparisons of popular fully distributed blockchain consensus methods.

Project	Method	Latency	Mobility	Energy Efficiency	Comput. Capability
Bitcoin [41]	PoW	10 min.	Low	Low	High
Ethereum [42]	PoW	12 sec.	Medium	Low	High
Spacemint [51]	PoC	4 min.	Low	High	High
Algorand [52]	PoS	20 sec.	Medium	High	Low
Omniledger [53]	ByzCoinX	14 sec.	Medium	Medium	Medium

III. SYSTEM OVERVIEW

This section presents the design goals, threat model, network model, security requirements, system model and preliminaries.

A. DESIGN GOALS

This study has the following primary goals with respect to decentralization, security and performance.

- Decentralization: The network is not managed by a central party. Subnetting allows the network to be divided into groups. The desire to achieve decentralization is to provide efficient key management services and extend the blockchain to device-to-device communication applications.
- Secure transaction: Securely transmit and commit transactions within a group to ensure the confidentiality of resources and to protect data integrity.
- Low latency: This research aims to reach a consensus within five seconds to meet the requirements of the IEEE 1609.2 standard security mechanism.

B. SECURITY REQUIREMENTS

OSI Security Architecture Recommendation ITU-T X.800 defines this systematic approach and focuses on security attacks, mechanisms and services. To satisfy the security characteristics of the proposed method, Table 3 lists some potential threats and suggested countermeasures for D2D communication. Therefore, this research design refers to this security framework to illustrate cybersecurity and the concepts of threats and attacks. Accordingly, we employ distributed ledger technology and dynamic group management. Furthermore, the study assumes that cryptographic primitives are secure and that computing the Diffie-Hellman problem is difficult. This study considers five categories of security services: *authentication, access control, confidentiality, integrity, and non-repudiation*. For example, the ECDSA and symmetric key algorithms are used to satisfy the integrity and confidentiality of security services. *authorization violation, eavesdropping, masquerading, and modification attacks* are the security threats considered in this study.

TABLE 3. Some potential attacks in D2D communication and the proposed countermeasures.

	Threats	Consequences	Countermeasures
Authentication	1. Impersonation of legitimate users 2. Data forgery 3. Man-in-the-middle attack	1. Misrepresentation of user 2. Belief that false information is valid	1. Hash-based message authentication code 2. ECDSA digital signature
Integrity	1. Modification of user data 2. Modification of message traffic in transit 3. Malware attack	1. Loss of info. 2. Vulnerability to all other threats	1. Hash-based message authentication code 2. ECDSA digital signature
Confidentiality	1. Eavesdropping on the net 2. Malware attack	1. Loss of info. 2. Loss of privacy	1. Diffie-Hellman/ ECDH key exchange 2. Symmetric key algorithm
Availability and Dependability	1. Denial of service (DoS)	1. Disruptive 2. Annoying 3. Prevent user from getting work done	1. Dynamic distributed consensus 2. Firewalls, intrusion detection, etc. DoS

C. THREAT MODEL

D2D communication is wireless and usually introduces some security holes [54]. A useful classification method for security attacks is the term for passive and active attacks. Passive attacks learn or use information in the system, such as eavesdropping, without affecting system resources. Conversely, active attacks attempt to change system resources or affect their operations, such as masquerading and modifying messages. To avoid potential threats, the proposed protocol needs to resist attacks, such as authorization violations, eavesdropping, masquerading and forgery, and modifying attacks [55], [56]. Figure 2 shows the security testing scenarios for this method.

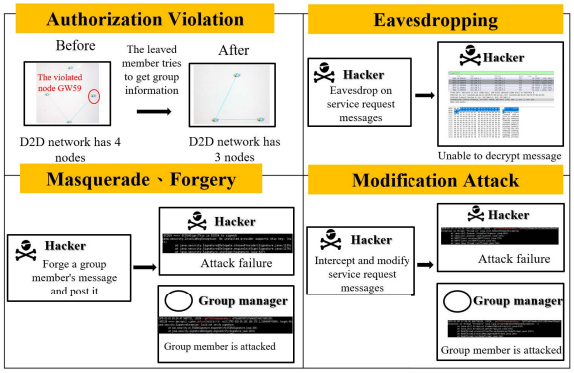


FIGURE 2. Security threats.

1) AUTHORIZATION VIOLATION

An attacker would like to illegally use a service or resource that is not available within the scope of the permission. Because the dynamic group key is calculated and updated using the current version of the dynamic group key and hash function, cracking the dynamic group key is difficult. Figure 3 shows the authorization violation test.

2) EAVESDROPPING

An attacker monitors the content of the network communication. The main countermeasure for eavesdropping is a

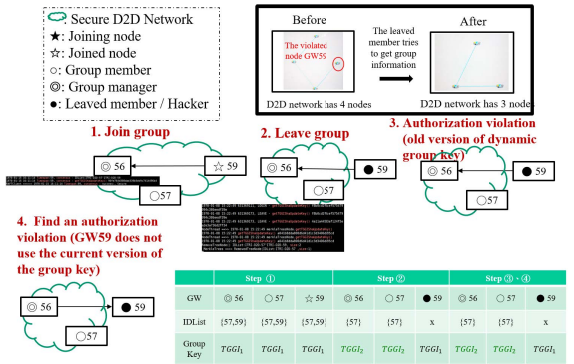


FIGURE 3. Authorization violation.

combination of encryption technology and dynamic group key management. Figure 4 shows the eavesdropping test results.

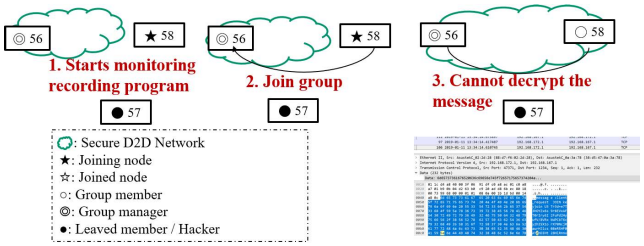


FIGURE 4. Eavesdropping.

3) MASQUERADE AND FORGERY

An attacker pretends to be a different entity, such as a group manager. For example, an attacker alters some portion of a legitimate message to produce an unauthorized effect. tn attacker intercepts and modifies the network's communication content and then resents it. The main countermeasures for masquerading and forgery are authentication, data integrity, non-repudiation and data confidentiality through digital signature technology and dynamic group key management. Figure 5 shows the masquerade and forgery tests.

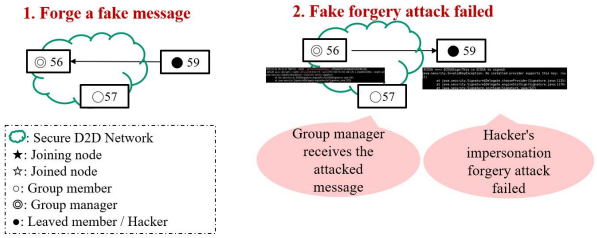


FIGURE 5. Masquerade and forgery.

4) MODIFICATION ATTACK

When the content of a data transfer is changed without detection, and resulting in unauthorized effects occurs, message modification happens. For example, an attacker intercepts and modifies the network's communication content and then resends it. The primary countermeasure for a modification attack is to use digital signature technology with dynamic group key management to perform authentication, data integrity, non-repudiation, and data confidentiality. Figure 6 shows the modification attack test.

D. NETWORK ARCHITECTURE

Figure 7 shows the architecture of the system. The N-tier architecture includes a cloud platform, local servers, edge devices, and user equipment. In cross view, the local server is the group manager of the gateway group *GWG*. In the vertical view, the UE group UEG_i consists of a group manager GW_i and multiple UEs. Note that the N-tier architecture

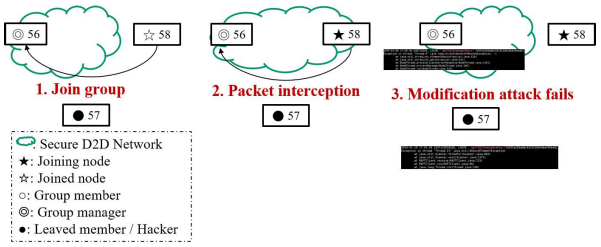


FIGURE 6. Modification attack.

can be controlled using remote cloud computing or local servers. Therefore, group permissions of local electronic devices enable decentralized control with short delays, fast processing, and fast transfer speeds. In other words, there are a group manager and n group members to process records to ensure the consistency of the system state. In each round, the group manager has a master copy of the data to be written. In addition, the group members maintain backup copies of data on the other nodes. That is to say, the Merkle root consensus checks to ensure that the digital digest data (i.e., hash data) is the same for every replica in the process. Therefore, each node's blockchain storage can be attributed to the overhead of system space complexity [57]. Furthermore, the space overhead of the proposed method can be reduced from the hierarchical structure. In other words, we need only to maintain the hash value of the backup copies. Because we have to compute only the root of the Merkle tree for validation. Therefore, the hash value is sufficient. The storage overhead for hash values is low.

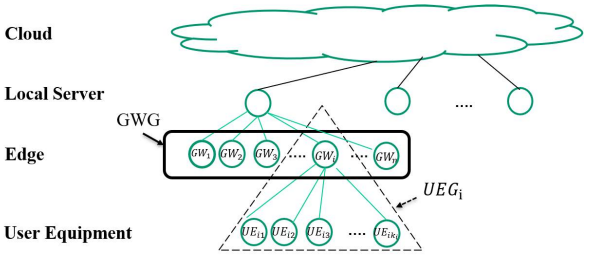


FIGURE 7. Decentralized Ad hoc network architecture.

E. SYSTEM MODEL

Figure 8 depicts the core technology components and core application components of the proposed method. Specifically, the core technical components mainly include consensus data transmission and reception, security services, data storage, network discovery, and message communication. The core application components can manage the group nodes. That is, the method adopts an asymmetric encryption mechanism to realize data encryption and decryption, signature verification, and authentication verification. This research provides security guarantees to protect the confidentiality, integrity, unforgeability, and privacy of data. For example, digital signatures provide authentication, data integrity, and data

confidentiality security services [55]. In addition, through the consensus mechanism, group nodes can verify data writing and other behaviors and reach a consensus trust establishment method.

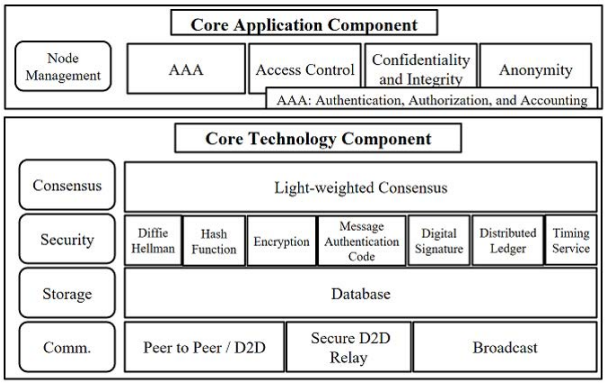


FIGURE 8. Blockchain-based technology of the proposed method.

F. PRELIMINARIES

This study leverages the salient features of cryptographic techniques and dynamic distributed consensus to provide a trusted and synchronized ledger for D2D communication.

1) BILINEAR PAIRING

$\mathbb{G}$ and $\mathbb{G}_T$ are two multiplicative cyclic groups of prime order q, where g and g_T are generators of $\mathbb{G}$ and $\mathbb{G}_T$ respectively. Mapping $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$ is called an acceptable bilinear mapping if the following attributes are satisfied.

- Bilinear: $\forall(x, y) \in \mathbb{Z}$, $e(g^x, g^y) = e(g, g)^{xy}$.
- Nondegenerate: If g generates $\mathbb{G}$ then $e(g, g)$ generates $\mathbb{G}_{\mathbb{T}}$.
- Computable: The group operations in $\mathbb{G}_1$, $\mathbb{G}_2$ and $\mathbb{G}_T$, as well as map e, are computable efficiently.

Definition 1 (Bilinear Parameter Generator Gen): A bilinear parameter generator Gen is a probabilistic algorithm. The algorithm takes the security parameter k as the input and outputs a tuple $(q, g, g_T, \mathbb{G}, \mathbb{G}_T, e)$, where $\mathbb{G}$ and $\mathbb{G}_T$ are two q factorial cyclic groups. g and g_T are two generators of $\mathbb{G}$ and $\mathbb{G}_T$, and $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$ is a recognized bilinear map.

2) CRYPTOGRAPHIC PRIMITIVES

Cryptography is one of the key techniques for realizing a secure distributed ledger [58].

- **Symmetric Ciphers**: For example, the widely used *Advanced Encryption Standard (AES)* is used for single-key or traditional encryption.
- **Asymmetric Ciphers**: Two related keys are used to perform supplementary operations such as encryption and decryption or signature generation and signature verification.
  1) *Public-key Cryptosystem*: The generation of such keys rely on an encryption algorithm based on mathematical problems to generate a one-way function. Using two keys has far-reaching implications for confidentiality, key distribution, and authentication [59].
  2) *Diffie Hellman key exchange*: The Diffie-Hellman algorithm depends on its effectiveness on the difficulty of computing *discrete logarithm*. For any integer b and the original root a of the prime p, a unique index i can be found. The index i is recalculated as the discrete logarithm of b for the base q, mod p.
  3) *Elliptic Curve Cryptography*: Elliptic curve encryption is based on computational hardness of the mathematical problem associated with elliptic curves. The well-known digital signature technology Elliptic Curve Digital Signature Algorithm (ECDSA) is based on elliptic curve cryptography.
  4) *Message Authentication Code*: The message authentication code function provides data integrity and data source authentication. When A has a message to send to B, it calculates the MAC based on message and key: $MAC = C(K, M)$, where M is input message, C is MAC function, K is shared secret key and MAC is message authentication code [55].
  5) *Digital signature*: A digital signature which is a public-key cryptography technique equipped with a pair of private and public keys (x, Y). A digital signature is composed of two main algorithms signature generation $Sign_x(.)$ and signature verification $Verify_Y(.)$:
     - *Signature generation* $Sign_x(.)$: Given a message m and a private key x, produces a digital signature on m as $Sign_m = Sign_x(m)$.
     - Signature verification $Verify_{pk}(.)$: Given the message m, a public key Y and digital signature $Sign_m$, either accepts or rejects the signature as being valid if $Verify_Y(m, Sign_m)$ outputs true.

3) DISTRIBUTED LEDGER TECHNOLOGY (DLT) AND MERKLE TREE

DLT is often used as a synonym for blockchain and refers to the distributed, decentralized ledger aspect of blockchain. In blockchain systems, the SHA256 hash function is $h : \{0, 1\}^* \rightarrow \{0, 1\}^{256}$ commonly used. For untrusted systems, consensus algorithms bring specific security attributes (integrity, agreement, and validity) [60]. The shared information is recorded in a trusted manner by collaboration between nodes in a distributed group using cryptographic techniques and distributed consensus algorithms.

The distributed ledger in the proposed method is on the basis of the structure of the Merkle tree (also commonly referred to as a hash tree). The Merkle tree provides efficient and secure verification of content that stores hash values and the ledger digest references the root node of the tree

data structure. The leaf nodes of a Merkle tree based on an unlimited one-time signature tree scheme are marked with values, such as pseudo identity *PID*. An example of a Merkle tree in a gateway group is shown in Figure 9. In addition, in the Merkle tree structure, the path length from any leaf to the root of a (balanced) binary tree with n leaves is determined using the $log_2 n$ approximation.

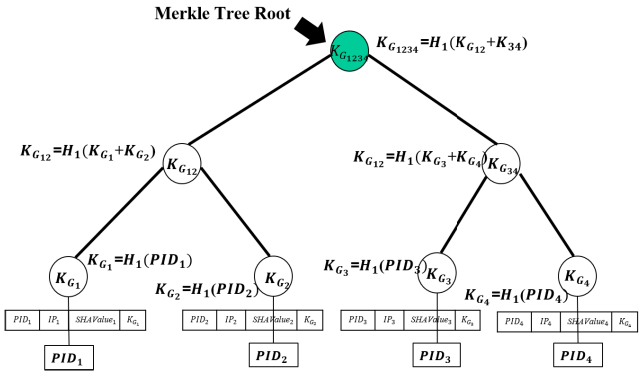


FIGURE 9. Example of Merkle tree.

IV. PROPOSED PROTOCOL

An authentication protocol based on hierarchical private group topology and distributed ledger technology was developed for D2D communication. A dynamic group management security mechanism can provide credential and dynamic membership management for secure IoT domains in groups.

A. SYSTEM INITIALIZATION

The entity descriptions, assumptions, system initialization phase, encryption keys, and hash function used in the groups are described below.

Entity descriptions are shown as follows.

- Local server: The local server acts as the group manager to manage the nodes in the gateway group *GW*. In addition, the real information of legitimate gateways and the UEs is stored therein. The local server can serve as a server or other electronic device with a powerful computing power.
- Gateway (GW): GWs act as gates from the UEs to a local server. GWs perform authentication operations for UE groups and can be deployed quickly through virtual machines. A gateway device may be a base station, a mobile edge computing platform, a roadside unit, or other computing capable electronic device, but it is not limited thereto.
- User equipment (UE): UEs are terminal devices in a network. A legitimate UE installs the D2D security application after registration with a local server. The personal device (user equipment) may be a mobile phone, a driving navigation device, a drone, or other electronic device having basic computing capabilities, but is not limited thereto.

Under the given assumptions, the local server and gateways are considered honest and sufficiently robust to provide correct source data and defend against attack. Moreover, UEs may consist of or be captured by some adversaries. Furthermore, the underlying encryption schemes are secure. Additionally, misbehaving nodes can be detected. For example, if a leaving node wants to illegally use a service or resource that is not available within its authority, the malicious node will not be able to grant permissions without an updated dynamic group key.

Algorithm 1 describes the five stages of system initialization of the gateway group performed by the local server. Note that the pre-shared key *preK* is used as the session key for the gateway group *GWG*. *preK* is prestored on each device's hardware encryption card, or produced and distributed by the local server through secure channels. In the same way, the UE group UEG_i is similar to *GWG*. For instance, the UE can establish a secure channel over SSL to ensure data integrity and confidentiality. The system parameters of UEG_i are issued by the UE group manager GW_i. UE_{ik_i} generates its public/private key pair by following the system setup phase of *GWG*.

The three stages related to the encryption key and hash function set of the group members are shown in Algorithm 2. Note that the process of triggering each dynamic group key update is based on joining the event or leaving event to *GWG* or UEG_i. In addition, the hash tree allows for efficient and secure verification of the contents of the data structure.

B. PROPOSED METHOD

After initializing the system, the proposed method focuses on how to share dynamic membership data between ad hoc network groups.

1) SOLUTION OVERVIEW

In the network security protocol, the requester and the group manager first perform an interactive mutual authentication phase. Subsequently, the group manager and other members enter the dynamically distributed consensus phase. If the network security system protocol is in a horizontal view, the same cross-group gateways belong to the same group, such as the *GWG*. Conversely, if the network security system protocol is in a vertical view, the gateway GW_i and multiple UEs are owned by the same group, such as UEG_i. Note that the process of the UE group is similar to that of the GW group except for the mutual authentication.

1) **Mutual authentication**: A secure process in which the requester and the group manager verifies each other's identities. If a node in *GWG* requests to join, the steps are described as follows. In addition, the leaving event is similar to the joining event. Figure 10 shows the state transitions of the proposed method. The requester sends the request and receives a response. On the other hand, the group manager processes the request and reaches a consensus with the group members.

Algorithm 1 System Initialization

Function `LocalServerIni`**:**

Generate and publish public key $Y_{localserver}$

Produce and distribute the pre-shared key *preK* via secure channels `/* as the session key for the gateway group` *GWG* `*/`

Function `SystemParameterGeneration`**:**

Initialization: bilinear mapping $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$ `/* 𝔾, 𝔾_T: multiplicative cyclic groups of prime order q, where g and g_T are generators of 𝔾 and 𝔾_T respectively. */`

Generate a tuple $(q, g, g_T, \mathbb{G}, \mathbb{G}_T, e)$ by running a bilinear parameter generator $Gen(k)$ `/* k: a given security parameter */`

Choose one secure symmetric encryption algorithm $Enc_{sk}(M)$ `/* M:msg, sk:secret key */`

Select two hash functions H_0 and H_1

`/*` $H_0 : \{0,1\}^* \rightarrow \mathbb{G}, H_1 : \{0,1\}^* \rightarrow \mathbb{Z}_q^*$ `*/`

Publish the system parameters of $GWG = (q, g, g_T, \mathbb{G}, \mathbb{G}_T, e, Enc_{sk}(M), H_0, H_1)$

Function `SystemSetup`**:**

The group manager of *GWG* ← the local server

while *not all the group managers of UE groups have been set* **do**

The group manager of $UEG_i \leftarrow GW_i$ GW_i randomly chooses its private key $x_i \in \mathbb{Z}_q^*$

GW_i calculates its public key by $Y_i = g^{x_i}$

Function `DigitalSignatureAlgoIni`**:**

The signature process is designed to be completed offline to reduce the data latency

The elliptic curve digital signature algorithm (ECDSA) selects private key x_i, message M, domain parameters, and public key Y_i as inputs

The output of ECDSA is $\sigma_i(M)$ signature

Function `PreRegistration`**:**

GW_i computes $Enc_{prek}(Y_i, GWID_i)$ and sends it to the local server for verification. `/* Similarly,` UE_{ik_i} `encrypts its public key` Y_{ik_i} `and real ID` RID_{ik_i} `and publishes` $Enc_{sk}(Y_{ik_i}, RID_{ik_i})$ `to the local server through` GW_i `*/`

Local server decrypts and verifies the encrypted data through *preK* or legitimate UE list

`/* If` UE_{ik_i} `is verified, the local server computes the pseudoidentity using` $PID_{ik_i} = H_1(RID_{ik_i})$ `*/`

The local server signs $(GWID_i, Y_i)$ by ECDSA using the private key $x_{localserver}$ and publishes it `/* Similarly,` (PID_{ik_i}, Y_{ik_i}) `in the authorized` UE_{ik_i} `*/`

Algorithm 2 Group Membership Setup

Function `DynamicGroupKey`**:**

Initialization:

The local server generates the initial temporary gateway group key *TGGI* or an initial temporary UE group key $TMGI_i$ for GW_i

TGGI and $TMGI_i$ are published by the local server to *GWG* and GW_i through secure channels, respectively

Dynamic Group Key Generation:

while *in GWG* **do**

if *a gateway node joins* **then**

The local server and group members update *TGGI* using $TGGI = H_1(TGGI)$

else

The local server chooses a random number $RAND_{localserver}$ and updates *TGGI* by $TGGI = H_0(TGGI, RAND_{localserver})$

The local server sends the updated *TGGI* to the remaining gateway group members

while *in* UEG_i **do**

if *an UE node joins* **then**

GW_i and group members update $TMGI_i$ by $TMGI_i = H_1(TMGI_i)$

else

GW_i selects a random number $RAND_{GW_i}$ and updates $TMGI_i$ by $TMGI_i = H_0(TMGI_i, RAND_{GW_i})$

GW_i publish the updated $TMGI_i$ to the remaining group members

Function `GroupMemberHashes`**:**

`/* is used for group consistency */`

while *is the local server* GW_0 *or group member in GWG* **do**

GW_0 and the group members produce a hardware-protected Merkle tree separately by $HK_{G_i} = H_1(GWID_i)$ `/*` HK_{G_i} `is located at leaf node */`

while *is the UE group manager* GW_i *(or* UE_0*) or group member in* UEG_i **do**

GW_i (or UE_0) and the group members generate a hardware-protected Merkle tree respectively by $HK_{U_{ik_i}} = H_1(PID_{ik_i})$ `/* Similar to` HK_{G_i} `*/`

Function `GroupMemberPrivateKey`**:**

`/* The group member private key is used for secure communication between the group manager and each group member */`

while *group member in GWG* **do**

$HK_{G_i}{}' = H_0(HK_{G_i}, RAND_{localserver})$

while *group member in* UEG_i **do**

$HK_{U_{ik_i}}{}' = H_0(HK_{U_{ik_i}}, RAND_{GW_i})$

- *Session Key Establishment.* A requester begins with a `ReqHello` message to a secure D2D network, followed by `AckHello` from the group manager. The two parties use a symmetric encryption key, such as a shared secret key, for the communication session.
- *Service Request.* The requester sends a `GrpMsgServiceRequest` message to the group manager. The requester creates message m and computes $m' = E(k, (m, Timestamp)_{sk_R})$ with its private key x_R and a secret shared key k. Then, the requester sends m and m' to the group manager.

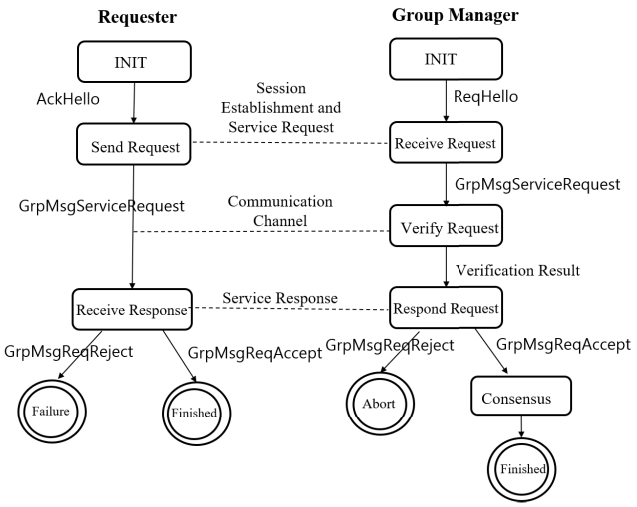


FIGURE 10. State transition: Mutual authentication.

- *Verify Request.* When the group manager receives a `GrpMsgServiceRequest` message, it validates the request through encryption and a digital signature. The group manager recovers $(m, Timestamp)_{x_R}$ using public key Y_R and the secret shared key k. Afterwards, the group manager verifies by checking if $(m, Timestamp)$ is equal to $D(k, m'_{Y_R})$. The group manager either sends `GrpMsgReqAccept` to accept the authentication request or `GrpMsgReqReject` to reject the request.
- *Service Response.* The group manager responds to the requester by using either `GrpMsgReqAccept` or `GrpMsgReqReject`. If a GrpMsgReqAccept is sent, then the group manager continues to complete the service. Otherwise, the group manager will abort the request.

2) **Dynamic Distributed Consensus**: When a new member joins or leaves a group, the group manager updates the dynamic group key. Note that the hierarchical topology has the advantages of being an effective group management and simple data fusion. Therefore, the proposed method with a hierarchical architecture helps address scalability and efficient resource utilization by reducing the communication load with the central authenticator. Furthermore, the proposed dynamic distributed consensus method is designed using cryptographic techniques to ensure consensus and trustworthiness in group management.

2) DYNAMIC GROUP KEY

An update of the dynamic group key is triggered by an event and is used to perform the consensus update process later. In addition, the dynamic group key is applied to *forward secrecy (FS)* and *backward secrecy (BS)* techniques.

Forward secrecy ensures that group members cannot decrypt previous group data sent before joining the group. This means that new users joining the conversation cannot access any old keys. On the other hand, *backward secrecy* ensures that members cannot decrypt data after leaving the group. The group members who leave the group will not have any access rights in the future.

The group manager then passes the consensus information to the group members. The detailed procedure is described in Section IV-B3.

3) DYNAMIC DISTRIBUTED CONSENSUS

The study proposes a dynamic distributed consistency solution to enable instant member changes across the entire participant network. In a traditional full blockchain, the node set is not known. However, the nodes of the proposed system are authorized. Each group member was responsible for distributed ledgers during each consensus period. Therefore, the group member information can be dynamically updated and maintained securely.

Figure 11 shows the state transition of the manager and members of the dynamic distributed consensus group. In the state diagram, the top label on the edge is the reason for the state transition (receive message). By contrast, the bottom label is a message sent due to a state transition. The detailed dynamic distributed consensus protocol includes the following phases.

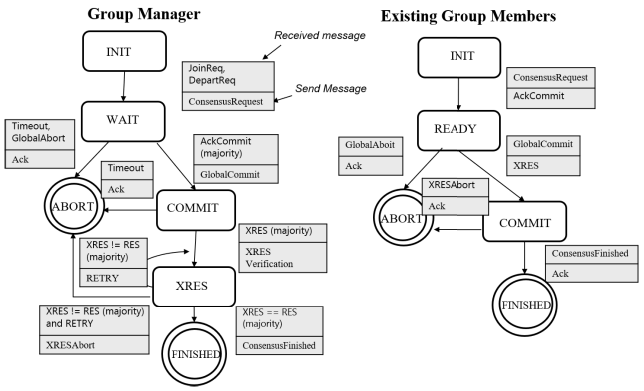


FIGURE 11. State transition: Dynamic distributed consensus.

1) **CONSENSUS REQUEST**: The group manager sends a `ConsensusReq` message to all group members.

The group manager generates a block log and proposal, and then multicasts the consensus information to all group members in the phase.

2) **ACK COMMIT**: When a group member receives a `ConsensusReq` message, it returns an `AckCommit` message to the group manager to tell the manager that it is ready to commit a portion of its consensus log locally. Otherwise, the group member returns the `ConsensusAbort` message.
3) **GLOBAL COMMIT**: When the group manager collects the majority response `AckCommit` from group members, and the group manager sends a `GlobalCommit` message to the group members. Otherwise, the group manager decides to abort the consensus transaction and multi-casts the `GlobalAbort` message.
4) **XRES RESPONSE**: Each group member that acknowledged for a commit waits for a reaction by the group manager. If a group member receives a `GlobalCommit` message, it commits to the consensus log locally. Afterwards, the group member responds to its Merkle tree root value `XRES` to the group manager for a consensus check. Otherwise, when a `GlobalAbort` message is received, the consensus transaction is locally aborted.
5) **CONSISTENCY CHECK**: When the group manager verifies that the majority of group members answer the Merkle Tree root value `XRES` successfully, the group manager sends a `ConsensusFinished` message to the group members. Otherwise, the group manager transmits `RETRY` message to the failure nodes. Subsequently, the group manager waits and updates the `XRES`.
   If the majority of group members answer correctly, the group manager sends a `ConsensusFinished` message to the group members. On the contrary, the group manager aborts the consensus transaction and multi-casts the `XRESAbort` message.

In addition, Algorithm 3 and 4 show the log replication process of the group manager and each group member, respectively, during the dynamic distributed consensus process. Additionally, Algorithm 5 describes the failover process when the group manager fails or crashes. The current group manager is selected based on the highest weight in the candidate group manager list or token ring rule. That is to say, the selection rules of the group manager are based on Token-Ring, computing workload, hardware support, capacity, etc.

4) SECURITY ANALYSIS

We now focus on the potential attacks and possible problems. An informal security analysis was performed on the following security features.

- *Authentication*: Entity authentication is implemented between several GWs (cross-group, such as *GWG*) in the same group or between GW and several UEs (vertical groups, such as UEG_i). When the GWs, GW and UEs exchange information, entity authentication is performed by checking membership. Typically, authentication in the D2D communication mode is implemented using the signature σ or message authentication code and the dynamic group key to verify the identity of the entity. In other words, the sender signs the group data before sending them to another device. Therefore, the recipient can verify the signer identity.
  In addition, the dynamic group key is only updated in the same group. In other words, non-members cannot authenticate and update the dynamic group key.
- *Data confidentiality*: During the transfer, the data is properly encrypted in transmission time and in the original text. The widest range of services protects all data transmitted between the sender and the receiver over a period of time.
- *Data integrity*: Message authentication code is used to provide message integrity and message verification. Digital signatures provide the source and integrity of data units to prevent forgery. In other words, the correctness and permission of the data are protected by the signature σ. Raw data from a sender is expected by verifying the original σ. In other words, the connection-oriented integrity service handles the message flow, ensuring that the message is received when the message is sent without modification, copying, inserting, reordering, or replaying. Therefore, the integrity and authority of the data is guaranteed.
  In the absence of a key (such as a session key and the dynamic group key), the eavesdropper cannot decrypt the ciphertext. In other words, a session key is shared between a sender and a receiver. In addition, the dynamic group key is updated in the same group.
  For example, the session uses an integer multiplicative set of modulo p, where p is a prime number and g is the original root modulus p. Both parties agree on the algorithm parameters p and g. The parties generate their private keys, named a, b, and c. During the session, an eavesdropper obtains two key hints g^b and g^c, and in the absence of b or c.
  However, during the session, the eavesdropper cannot export the shared key g^{bc} under the discrete log problem (DLP) assumption during the session.
  Moreover, if the eavesdropper is not a member of the group, the eavesdropper cannot acquire the updated dynamic group key. Therefore, the eavesdropping attack is resisted and the confidentiality of the data is guaranteed.
- *Non-repudiation*: Mutual authentication procedure and dynamic consensus procedure are undeniable to both the sender and the receiver. A sender's signature σ does not give the entity an opportunity to reject the transfer event. In other words, digital signatures provide protection that prevents the entities participating in the communication from participating in all or part of the communication.

Therefore, non-repudiation prevents messages that the sender or receiver refuses to transmit.

V. EVALUATION

The method was tested using the well-known network simulator OMNET++ and was implemented in an actual experimental environment. The experimental tests were passed by a trusted third party and are divided into three types: functional tests, performance tests, and security tests.

To begin with, functional tests included mutual authentication: authentication (identity verification), mutual authentication: integrity, mutual authentication: non-repudiation, and mutual authentication: confidentiality. In addition, performance tests were performed on the transmission delay of mutual authentication and consensus information. Note that the security tests include violations of authorization, masquerade and forgery, eavesdropping, and modification attacks.

A. SIMULATION

The ad hoc mode and IEEE 802.11 infrastructure were implemented in the INET framework version 3.2.4 of OMNeT++ simulator version 4.6 [61]. Among them, OMNeT++ is an open source network simulator widely used in academia. The realization of the Wi-Fi Direct function for D2D communication in the work is based on [62] to negotiate the group manager and group members with secure communication. Figure 12 shows the simulation performance analysis between the centralized and hierarchical topologies. The results demonstrated that the hierarchical topology had advantages in terms of effective group management and simple data fusion.

In this section, Wi-Fi Direct is realized in five cases. In the test scenario, the group manager is assigned at the beginning of the simulation. As a result, other nodes detected the existing group manager and joined or left the group. In the test cases, there are a different topology, in which there is a group manager with group members 10, 50, 100, and 300. The Wi-Fi Direct function was implemented as a management module to promote its use, customization, and integration in OMNeT++.

B. EXPERIMENTAL ENVIRONMENT SETTING

In the experiment, these devices are installed and different nodes are run for the gateway group on the Cubieboard4 CC-A80 platform. In addition, smart mobile devices as UEs are used for unrestricted user brand Android OS 6.0 or higher user device groups. CC-A80 has 2GB DDR3 memory, onboard VGA display port, 100M/1000M RJ45, dual-band WIFI, onboard Bluetooth, supports lithium battery and RTC battery, four USB ports and 1 USB 3.0 OTG port. The implemented experimental environment is performed on D2D communication using WiFi Direct technology.

Let GW_B be the group manager of the gateway group, as shown in Figure 13. If the new gateway GW_C wanted to

Algorithm 3 Log Replication: At Group Manager

Data: `JoinReq` or `DepartReq`, `AckCommit`, `XRES`
Result: `ConsensusReq`, `GlobalCommit`, `ConsensusFinished`
Initialization: `acknowledgment`, `XRES` $\leftarrow 0$

```
write StartConsensus to local log and multicast ConsensusReq to all group members ; /* Phase 1 */
while not the majority acknowledgment collected do
    wait for incoming acknowledgment ;   /* Phase3 */
    if timeout then
        write GlobalAbort to local log;
        multicast GlobalAbort to all group members and exit;
    end
    record acknowledgment;
end
if the majority group members sent AckCommit then
    write GlobalCommit to local log and multicast GlobalCommit to all group members
else
    write GlobalAbort to local log and multicast GlobalAbort to all group members;
end
while not the majority XRES have been collected do
    wait for any incoming XRES ;        /* Phase 5 */
    if timeout then
        write GlobalAbort to local log;
        multicast GlobalAbort to all group members and exit;
    end
    record XRES ;
end
if the majority group members sent XRES then
    retrieve RES from its MerkleTreeRoot ;
    if the majority group members answer (XRES == RES) then
        write ConsensusFinished to local log;
        multicast ConsensusFinished
    else
        ask the group members for RETRY and wait for any incoming XRES;
        record and update XRES ;
        if not the majority answer (XRES == RES) and timeout then
            write XRESAbort to local log;
            multicast XRESAbort to all group members and exit;
        else
            write ConsensusFinished to local;
            multicast ConsensusFinished to all group members;
        end
    end
else
    write GlobalAbort to local log;
    multicast GlobalAbort to all group members;
end
```

Algorithm 4 LogSync: For Each Group Member

Data: ConsensusReq
GlobalCommit
Result: AckCommit
XRES ; /* XRES: Respond MerkleTree root for consistency check */
Initialization: DECISION $\leftarrow \phi$
AckCommit $\leftarrow \phi$
RESULT $\leftarrow \phi$

write INIT to local log ; /* Make transition to abort safely */
wait for ConsensusReq from the group manager;
if *timeout* **then**
  write ConsensusAbort to local log;
  exit ; /* Abort the procedure LogSync consensus transaction */
end
if *group member sends an acknowledgment* **then**
  write AckCommit to local log;
  send AckCommit to the group manager ; /* Phase 2: ACK to the group manager by the existing group members */
  wait for DECISION from the group manager;
  if *timeout* **then**
    ask the group manager for DECISION;
  end
  write DECISION to local log;
  if *(DECISION == GlobalCommit)* **then**
    write GlobalCommit to local log ; /* Update the merkle Tree */
    respond XRES = its MerkleTreeRoot to the group manager ; /* Phase 4: Respond for consistency check by the group members */
    wait for RESULT from the group manager;
    if *timeout* **then**
      ask the group manager for RESULT;
    end
    write RESULT to local log;
    if *(RESULT == RETRY)* **then**
      re-calcuate XRES;
      respond XRES to the group manager ;
      /* RETRY: Response to XRES correction */
    else if *(RESULT == ConsensusFinished)* **then**
      write ConsensusFinished to local log;
    else
      write XRESAbort to local log;
  else
    write GlobalAbort to local log;
  end
else
  write ConsensusAbort to local log;
  send ConsensusAbort to the group manager;
end

Algorithm 5 Failover: Group Manager Selection

Data: Trigger Event TEvent ; /* The current group manager is busy loading or leaving or crashing */
RULE ; /* The group manager selection rule */
Result: Group manager candidate GMC(N)
Initialization: Select the initial group manager candidate list through the local server

write INIT to the local log ; /* Initial group manager candidate list */
wait for TEvent ;
if *timeout* **then**
  write FailoverAbort to local log;
  exit ; /* Abort procedure, Failover */
end
while *TEvent* **do**
  if *RULE == 1* **then**
    Calculate the group manager candidates according to specified features ; /* RULE 1: computing workload, hardware support, capacity, etc.. */
    Sort the group manager candidates and select the first ranking node GMC(N) ;
  else
    Select new group manager candidate GMC(N) using the Token Ring method ; /* RULE 2: Token-Ring */
  end
  GM = GMC(N) ; /* Current group manager is updated */
  update the list of new group manager candidates;
end
regularly monitor the data and energy of each group member;

join, the security group management mechanism is activated when GW_C communicates with the D2D network. GW_C then authenticated with the group manager GW_B. After that, the group manager GW_B and the existing team member GW_A performed a consensus check. Note that an unauthorized node would be forced to disconnect from the network.

C. EVALUATION

1) PERFORMANCE TEST

The experimental topology and environment for the performance test are shown in Figure 14. The Remote Manager web-based page displays the gateway connection status of the D2D network. There are five of these devices used to test the security protocols proposed in D2D communication. Figure 15 describes the performance of joining the group of five devices, including the request process and the consensus process. In addition, Figure 16 demonstrates the performance of leaving the group of five devices. Note that the practical

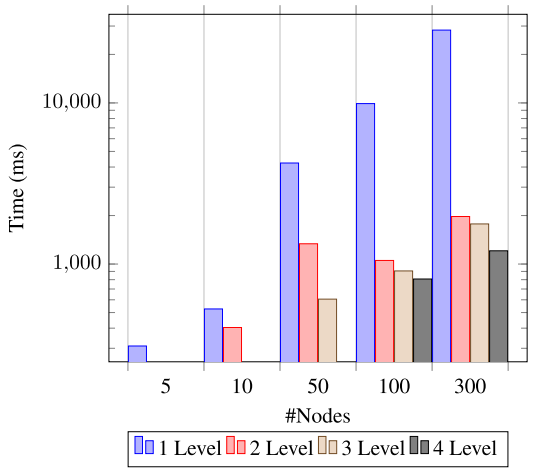


FIGURE 12. Simulation performance analysis.

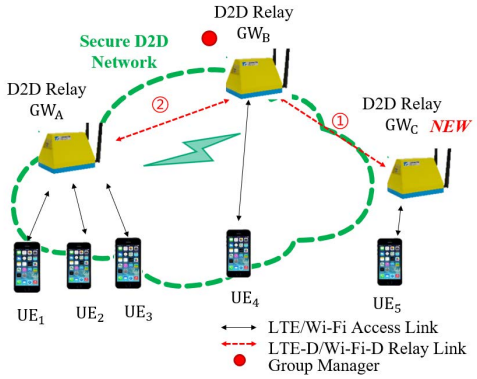
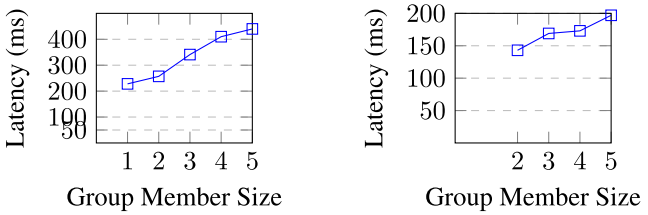


FIGURE 13. Experiment setting.

results have higher delay than the simulation results and may be affected by propagation delay, transmission delay, nodal processing delay, and device performance.

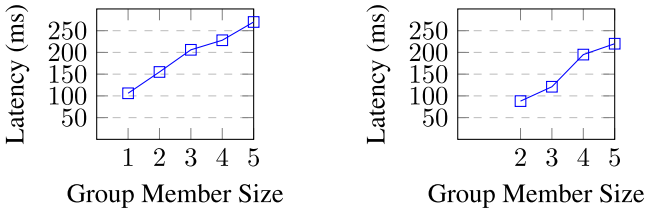
(a) Topology for Testing (b) Experiment Devices

FIGURE 14. Experimental topology and devices.



(a) Joining Request (b) Consensus for Joining

FIGURE 15. Joining procedure.



(a) Leaving Request (b) Consensus for Leaving

FIGURE 16. Leaving procedure.

2) COMPARISON

Table 4 presents the security features of the proposed protocol and existing blockchain-based group key agreement protocols. Let denote security properties such as S1: "Replay Attack Resistance", S2: "Forgery Attack Resistance", S3: "Forward Secrecy", S4: "Backward Secrecy", S5: "Mutual Authentication" and S6: "Anonymity". However, existing schemes [63], [64], [65] do not consider or guarantee safety features. In [63], Mandal *et al.* proposed a certificateless group key agreement framework to solve the key escrow problem using cryptographic signatures. Nevertheless, this protocol does not support privacy protection. Tan and Chung [64] designed a blockchain-based certificateless authentication scheme using consortium blockchains to manage authentication and group key distribution in VANET. However, their protocol does not take into account the backward secrecy requirement of the group key. Baga *et al.* [65] introduced a blockchain-based secure communication privacy protection batch authentication scheme for VANET. However, the batch verification scheme do not consider whether there are invalid signatures in batch signatures. Furthermore, their protocol does not support forward secrecy and backward secrecy for group keys. Therefore, according to the Table 4, the proposed protocol based on hierarchical topology and distributed ledger technology can provide higher security compared to existing protocols. The overhead of the proposed method can be reduced from the hierarchical structure. Furthermore, the proposed dynamic group management and dynamic distributed consensus scheme can provide higher security.

TABLE 4. Blockchain-based group key protocol comparison.

Protocol	S1	S2	S3	S4	S5	S6
Mandal *et al.* [63]	✓	✓	✓	✓	✓	✗
Tan and Chung [64]	✓	✓	✓	✗	✓	✓
Bagga *et al.* [65]	✓	✓	✗	✗	✓	✓
Proposed Work	✓	✓	✓	✓	✓	✓

Table 5 describes a comparison of group key management protocols. A hierarchical topology with N nodes allows a communication distance of $O(logN)$ between the group manager and group members through the fabric, while a KDC

(Key Distribution Center) based topology would generate $O(N)$. Note that [66] proposed an authenticated asymmetric group key agreement protocol to support node dynamics. However, if the manager leaves the system, the system must be reinitialized, which results in a lot of communication and computational overhead. For our distributed ledger-based architecture and failover support, if a node leaves the system, the communication and computation complexity is $O(logN)$ and $O(N)$, respectively. Therefore, the proposed hierarchical scheme is more efficient and flexible.

TABLE 5. Performance comparison.

Scheme	Computational Cost		Communication Cost	
	Join	Leave	Join	Leave
Zhang *et al.* [66]	$O(N)$	$O(N)$	$O(N)$	$O(N)$
Chen *et al.* [67]	$O(N)$	$O(N)$	$O(N)$	$O(N)$
Tseng *et al.* [68]	$O(N)$	$O(N)$	$O(N)$	$O(N)$
Proposed	$O(N)$	$O(N)$	$O(logN)$	$O(logN)$

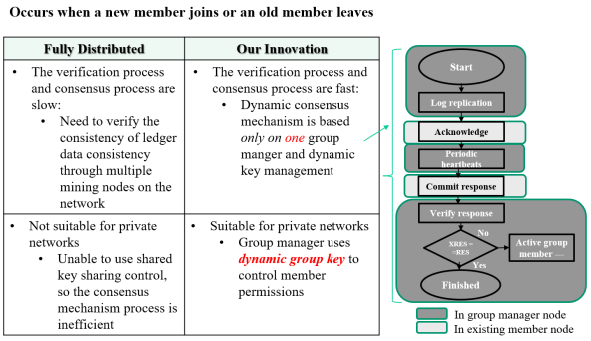


FIGURE 17. Consensus Comparison with Fully Distributed System.

VI. DISCUSSION AND FUTURE DIRECTION

A. DISCUSSION

A **fast and reliable dynamic group management** security mechanism allows the authentication and membership update time to be less than 1 second.

- *Dynamic group management.* When a new member joins or one of the old members leaves a group, the dynamic group key is updated. After a group manager and group members update the dynamic group keys, a dynamically distributed consensus process for the group manager and group members is performed.
- *Dynamic distributed consensus*. The proposed dynamic distributed consensus method was designed using cryptographic techniques to ensure consensus and trustworthiness in group management. The consensus algorithm specifies a set of rules and procedures that all the participating nodes should follow. A group manager is responsible for managing group membership through a dynamic group key and dynamic distributed consensus mechanisms. Figure 17 shows a comparison with a fully distributed system. For example, the Bitcoin network utilizes proof-of-work consensus and has a high latency of approximately 10 minutes, making it ineffective for ad hoc networks. In other words, the PoW relies on machines to perform mathematical operations to obtain bookkeeping rights. At the same time, each time a consensus is reached, the entire network must to participate in the calculation, and the performance efficiency is relatively low.

A hierarchical topology with N nodes allows the average distance between the group manager and group members to be $O(logN)$ through a structure, whereas a fully distributed topology would generate $O(N)$. In other words, a rights-based **private decentralized network** that limits the group member size, so trust is not lost in a decentralized network.

- *Trust communication.* The private decentralized network map of the honest nodes is well collected, and the communication channels between the honest nodes are synchronized.
- *No Intrinsic incentive.* The private decentralized network can be implemented without an intrinsic token to provide an economic incentive as a fully distributed blockchain system.

In summary, the proposed method must trade off scalability and performance efficiency to address high throughput or security issues with a large number of nodes. Therefore, we propose a layered decentralized architecture with distributed ledger technology for dynamic group management to reduce overhead and enhance security. For example, the space overhead of the proposed method can be reduced from the hierarchical structure. In other words, we only need to maintain the hash of the backup copy. Because we only need to compute the root of the Merkle tree for verification. Therefore, the storage overhead of the hash value is low. However, the proposed method is still a proof of concept and has limitations in future work. First, further governance and regulation are required. In addition, another disadvantage of the study is that it does not formally focus on the usual honest or malicious devices, which may be unrealistic in anonymity and public systems. Furthermore, the study has some limitations, namely, the group key update and session key exchanged in each message introduced network management traffic. Therefore, this study is suitable for group management within a limited range due to the propagation delay caused by the distance from the transmitter to the receiver, or the transmission delay caused by the wireless network protocol used. In a normal blockchain system, if the node size scales to hundreds or thousands, the key bottleneck for scalability is: performance drops significantly. In other words, the more nodes participating in validation, the longer the latency will generally result, which reduces throughput. Additionally, malicious clients masquerading as legitimate clients pose a higher security threat to larger participant sizes.

B. FUTURE DIRECTION

1) DEEP LEARNING LEDGER

The remote location of the cloud makes the current Artificial intelligence (AI) algorithms ineffective or inefficient

for time-critical applications. Currently, the emergence of distributed ledger technology makes it more likely to bring AI to the edge. Therefore, combining deep learning and distributed ledger techniques helps provide an advanced computing infrastructure [69], [70], [71]. For example, combining blockchain and deep recurrent neural networks for edge computing fleet identification.

2) KEYLESS SIGNATURE INFRASTRUCTURE

The keyless signature infrastructure (KSI) is a globally distributed system for providing server support and timestamp digital signature services. Keyless signatures are efficient and are an alternative to traditional public key infrastructure. Accordingly, KSI can solve the long-term validity of digital signatures in a private ad hoc network [69]. For example, fleet nodes can use KSI to attest to the registration time of group management.

3) QUANTUM INFORMATION

Quantum teleportation and blind quantum computing are powerful tools [69], [72]. Therefore, to better transmit quantum information through vehicular networks, it is necessary to explore the related quantum vehicular network communication technologies. For example, quantum key distribution technology has been applied for identity registration and authentication in VANET.

VII. CONCLUSION

Owing to frequent changes in the number of nodes in wireless ad hoc networks, this study proposes a distributed dynamic security authentication adjustment method to solve the limitations of centralized and fully distributed networks. The distributed ledger-based architecture for dynamic group management does not affect the security and performance. This study leverages the salient features of cryptographic techniques and dynamic distributed consensus to provide a trusted and synchronized ledger for D2D communication. The dynamic membership extension of the consensus algorithm with dynamic group keys allows a set of nodes in the group to change over time. Additionally, the proposed method can facilitate the transmission of direct communication data without a centralized database, thereby reducing the chance of a single point of failure.

Moreover, the research was tested using the well-known network simulator OMNET++ and was implemented in an experimental environment. In addition, the experimental tests were conducted by a trusted third party. Therefore, this study provides appropriate procedures for the group management of customers and public fleets (such as drones).

ACKNOWLEDGMENT

The authors would like to thank their close partner Wistron NeWeb Corporation and also would like to thank the Third Party Institute for Information Industry (III) for the security test.

REFERENCES

[1] N. Wang, W. Li, A. Alipour-Fanid, L. Jiao, M. Dabaghchian, and K. Zeng, "Pilot contamination attack detection for 5G mmWave grant-free IoT networks," *IEEE Trans. Inf. Forensics Security*, vol. 16, pp. 658–670, 2021.
[2] M. A. Ferrag, L. Maglars, A. Argyriou, D. Kosmanos, and H. Janicke, "Security for 4G and 5G cellular networks: A survey of existing authentication and privacy-preserving schemes," *J. Netw. Comput. Appl.*, vol. 101, pp. 55–82, Jan. 2018.
[3] S. Mumtaz, A. Al-Dulaimi, V. Frascolla, S. A. Hassan, and O. A. Dobre, "Guest editorial special issue on 5G and beyond—Mobile technologies and applications for IoT," *IEEE Internet Things J.*, vol. 6, no. 1, pp. 203–206, Feb. 2019.
[4] M. P. R. S. Kiran and P. Rajalakshmi, "Saturated throughput analysis of IEEE 802.11 AD EDCA for high data rate 5G-IoT applications," *IEEE Trans. Veh. Technol.*, vol. 68, no. 5, pp. 4774–4785, May 2019.
[5] R. Abozariba, M. K. Naeem, M. Patwary, M. Seyedebrahimi, and P. Bull, "NOMA-based resource allocation and mobility enhancement framework for IoT in next generation cellular networks," *IEEE Access*, vol. 7, pp. 29158–29172, 2019.
[6] F. Jameel, M. A. Javed, S. Zeadally, and R. Jäntti, "Efficient mining cluster selection for blockchain-based cellular V2X communications," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 7, pp. 1–9, Jul. 2020.
[7] F. Jameel, W. U. Khan, N. Kumar, and R. Jäntti, "Efficient power-splitting and resource allocation for cellular V2X communications," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 6, pp. 1–10, Jun. 2020.
[8] S. Liu, Y. Wu, L. Li, X. Liu, and W. Xu, "A two-stage energy-efficient approach for joint power control and channel allocation in D2D communication," *IEEE Access*, vol. 7, pp. 16940–16951, 2019.
[9] K. Doppler, M. Rinne, C. Wijting, C. B. Ribeiro, and K. Hugl, "Device-to-device communication as an underlay to LTE-advanced networks," *IEEE Commun. Mag.*, vol. 47, no. 12, pp. 42–49, Dec. 2009.
[10] P. Gandotra and R. K. Jha, "Device-to-device communication in cellular networks: A survey," *J. Netw. Comput. Appl.*, vol. 71, pp. 99–117, Aug. 2016.
[11] L. Zhou and Z. J. Haas, "Securing Ad hoc networks," *IEEE Netw.*, vol. 13, no. 6, pp. 24–30, Nov. 1999.
[12] L. Ferretti, M. Marchetti, and M. Colajanni, "Fog-based secure communications for low-power IoT devices," *ACM Trans. Internet Technol.*, vol. 19, no. 2, pp. 1–21, May 2019.
[13] G. He, W. Su, S. Gao, J. Yue, and S. K. Das, "ROAchain: Securing route origin authorization with blockchain for inter-domain routing," *IEEE Trans. Netw. Service Manag.*, vol. 18, no. 2, pp. 1–16, Jun. 2020.
[14] W. Hao, J. Zeng, X. Dai, J. Xiao, Q.-S. Hua, H. Chen, K.-C. Li, and H. Jin, "Towards a trust-enhanced blockchain P2P topology for enabling fast and reliable broadcast," *IEEE Trans. Netw. Service Manage.*, vol. 17, no. 2, pp. 904–917, Jun. 2020.
[15] X. Li, T. Liu, M. S. Obaidat, F. Wu, and P. Vijayakumar, "A lightweight privacy-preserving authentication protocol for VANETs," *IEEE Syst. J.*, vol. 14, no. 3, pp. 3547–3557, May 2020.
[16] S. H. Islam, M. S. Obaidat, P. Vijayakumar, E. Abdulhay, F. Li, and M. K. C. Reddy, "A robust and efficient password-based conditional privacy preserving authentication and group-key agreement protocol for VANETs," *Future Gener. Comput. Syst.*, vol. 84, pp. 216–227, Jul. 2018.
[17] G. Hatzivasilis, O. Soultatos, S. Ioannidis, G. Spanoudakis, V. Katos, and G. Demetriou, "MobileTrust: Secure knowledge integration in VANETs," *ACM Trans. Cyber-Phys. Syst.*, vol. 4, no. 3, pp. 1–25, Jul. 2020.
[18] S. Tangade, S. S. Manvi, and P. Lorenz, "Trust management scheme based on hybrid cryptography for secure communications in VANETs," *IEEE Trans. Veh. Technol.*, vol. 69, no. 5, pp. 5232–5243, May 2020.
[19] M. Kamal, G. Srivastava, and M. Tariq, "Blockchain-based lightweight and secured V2V communication in the internet of vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 7, pp. 3997–4004, Jul. 2020.
[20] I. S. Association *et al.*, *IEEE Trial-Use Standard for Wireless Access in Vehicular Environments-Security Services for Applications and Management Messages*, IEEE Standard 1609.2-2006, 2006.
[21] F. Jameel, Z. Hamid, F. Jabeen, S. Zeadally, and M. A. Javed, "A survey of device-to-device communications: Research issues and challenges," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 3, pp. 2133–2168, 3rd Quart., 2018.
[22] F. Tariq, M. Anwar, A. R. Janjua, M. H. Khan, A. U. Khan, and N. Javaid, "Blockchain in WSNs, VANets, IoTs and healthcare: A survey," in *Proc. Workshops Int. Conf. Adv. Inf. Netw. Appl.*, vol. 1150. Cham, Switzerland: Springer, 2020, pp. 267–279.

[23] H. Chai, S. Leng, J. He, K. Zhang, and B. Cheng, "CyberChain: Cybertwin empowered blockchain for lightweight and privacy-preserving authentication in Internet of Vehicles," *IEEE Trans. Veh. Technol.*, vol. 71, no. 5, pp. 4620–4631, May 2022.

[24] B. Leiding, P. Memarmoshrefi, and D. Hogrefe, "Self-managed and blockchain-based vehicular Ad-hoc networks," in *Proc. ACM Int. Joint Conf. Pervasive Ubiquitous Comput., Adjunct*, Sep. 2016, pp. 137–140.

[25] N. Lasla, M. Younis, W. Znaidi, and D. Ben Arbia, "Efficient distributed admission and revocation using blockchain for cooperative ITS," in *Proc. 9th IFIP Int. Conf. New Technol., Mobility Secur. (NTMS)*, Feb. 2018, pp. 1–5.

[26] M. Singh and S. Kim, "Branch based blockchain technology in intelligent vehicle," *Comput. Netw.*, vol. 145, pp. 219–231, Nov. 2018.

[27] J. Li, T. Liu, D. Niyato, P. Wang, J. Li, and Z. Han, "Contract-theoretic pricing for security deposits in sharded blockchain with Internet of Things (IoT)," *IEEE Internet Things J.*, vol. 8, no. 12, pp. 10052–10070, Jun. 2021.

[28] X. Wang, S. Garg, H. Lin, M. J. Piran, J. Hu, and M. S. Hossain, "Enabling secure authentication in industrial IoT with transfer learning empowered blockchain," *IEEE Trans. Ind. Informat.*, vol. 17, no. 11, pp. 7725–7733, Nov. 2021.

[29] L. Feng, H. Zhang, L. Lou, and Y. Chen, "A blockchain-based collocation storage architecture for data security process platform of WSN," in *Proc. IEEE 22nd Int. Conf. Comput. Supported Cooperat. Work Design (CSCWD)*, May 2018, pp. 75–80.

[30] M. R. Abdmeziem, D. Tandjaoui, and I. Romdhani, "A decentralized batch-based group key management protocol for mobile Internet of Things (DBGK)," in *Proc. IEEE Int. Conf. Comput. Inf. Technol., Ubiquitous Comput. Commun., Dependable, Autonomic Secure Comput., Pervasive Intell. Comput.*, Oct. 2015, pp. 1109–1117.

[31] H. T. T. Truong, M. Almeida, G. Karame, and C. Soriente, "Towards secure and decentralized sharing of IoT data," in *Proc. IEEE Int. Conf. Blockchain (Blockchain)*, Jul. 2019, pp. 176–183.

[32] L. Xu, L. Chen, Z. Gao, X. Fan, T. Suh, and W. Shi, "DIoTA: Decentralized-ledger-based framework for data authenticity protection in IoT systems," *IEEE Netw.*, vol. 34, no. 1, pp. 38–46, Jan. 2020.

[33] L. Veltri, S. Cirani, S. Busanelli, and G. Ferrari, "A novel batch-based group key management protocol applied to the Internet of Things," *Ad Hoc Netw.*, vol. 11, no. 8, pp. 2724–2737, Nov. 2013.

[34] E. Abd-Elrahman, H. Ibn-khedher, and H. Afifi, "D2D group communications security," in *Proc. Int. Conf. Protocol Eng. (ICPE) Int. Conf. New Technol. Distrib. Syst. (NTDS)*, Jul. 2015, pp. 1–6.

[35] M. Dammak, S.-M. Senouci, M. A. Messous, M. H. Elhdhili, and C. Gransart, "Decentralized lightweight group key management for dynamic access control in IoT environments," *IEEE Trans. Netw. Service Manage.*, vol. 17, no. 3, pp. 1742–1757, Sep. 2020.

[36] Z. Lu, W. Liu, Q. Wang, G. Qu, and Z. Liu, "A privacy-preserving trust model based on blockchain for VANETs," *IEEE Access*, vol. 6, pp. 45655–45664, 2018.

[37] V. O. Nyangaresi, A. J. Rodrigues, and N. K. Taha, "Mutual authentication protocol for secure VANET data exchanges," in *Proc. Int. Conf. Future Access Enablers Ubiquitous Intell. Infrastructures*. Cham, Switzerland: Springer, 2021, pp. 58–76.

[38] A. Zhang, L. Wang, X. Ye, and X. Lin, "Light-weight and robust security-aware D2D-assist data transmission protocol for mobile-health systems," *IEEE Trans. Inf. Forensics Security*, vol. 12, no. 3, pp. 662–675, Mar. 2016.

[39] H. Xiong, "Cost-effective scalable and anonymous certificateless remote authentication protocol," *IEEE Trans. Inf. Forensics Security*, vol. 9, no. 12, pp. 2327–2339, Dec. 2014.

[40] J. Liu, Z. Zhang, X. Chen, and K. S. Kwak, "Certificateless remote anonymous authentication schemes for wirelessbody area networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 2, pp. 332–342, Feb. 2013.

[41] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," SSRN, Rochester, NY, USA, Tech. Rep. SSRN 3440802, 2019.

[42] G. Wood, "Ethereum: A secure decentralised generalised transaction ledger," *Ethereum Project Yellow Paper*, vol. 151, pp. 1–32, Apr. 2017.

[43] A. Gervais, H. Ritzdorf, G. O. Karame, and S. Capkun, "Tampering with the delivery of blocks and transactions in bitcoin," in *Proc. 22nd ACM SIGSAC Conf. Comput. Commun. Secur.*, Oct. 2015, pp. 692–705.

[44] J. Grover, "Security of vehicular Ad hoc networks using blockchain: A comprehensive review," *Veh. Commun.*, vol. 34, Apr. 2022, Art. no. 100458.

[45] H. Wang, T. Liu, B. Kim, C.-W. Lin, S. Shiraishi, J. Xie, and Z. Han, "Architectural design alternatives based on cloud/edge/fog computing for connected vehicles," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 4, pp. 2349–2377, 4th Quart., 2020.

[46] G. Karagiannis, O. Altintas, E. Ekici, G. Heijenk, B. Jarupan, K. Lin, and T. Weil, "Vehicular networking: A survey and tutorial on requirements, architectures, challenges, standards and solutions," *IEEE Commun. Surveys Tuts.*, vol. 13, no. 4, pp. 584–616, Nov. 2011.

[47] S. Kim, "Impacts of mobility on performance of blockchain in VANET," *IEEE Access*, vol. 7, pp. 68646–68655, 2019.

[48] Q. Zhang, L. Cheng, and R. Boutaba, "Cloud computing: State-of-the-art and research challenges," *J. Internet Services Appl.*, vol. 1, no. 1, pp. 7–18, May 2010.

[49] H. Khayyam, J. Abawajy, B. Javadi, A. Goscinski, A. Stojcevski, and A. Bab-Hadiashar, "Intelligent battery energy management and control for vehicle-to-grid via cloud computing network," *Appl. Energy*, vol. 111, pp. 971–981, Nov. 2013.

[50] J. Fan, W. Yang, Z. Liu, J. Kang, D. Niyato, K.-Y. Lam, and H. Du, "Cybersecurity challenges of IoT-enabled smart cities: A survey," 2022, *arXiv:2202.05023*.

[51] S. Park, A. Kwon, G. Fuchsbauer, P. Gaži, J. Alwen, and K. Pietrzak, "Spacemint: A cryptocurrency based on proofs of space," in *Proc. Int. Conf. Financial Cryptogr. Data Secur.* Berlin, Germany: Springer, 2018, pp. 480–499.

[52] Y. Gilad, R. Hemo, S. Micali, G. Vlachos, and N. Zeldovich, "Algorand: Scaling byzantine agreements for cryptocurrencies," in *Proc. 26th Symp. Operating Syst. Princ.*, Oct. 2017, pp. 51–68.

[53] E. Kokoris-Kogias, P. Jovanovic, L. Gasser, N. Gailly, E. Syta, and B. Ford, "OmniLedger: A secure, scale-out, decentralized ledger via sharding," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2018, pp. 583–598.

[54] A. Zhang, J. Chen, R. Q. Hu, and Y. Qian, "SeDS: Secure data sharing strategy for D2D communication in LTE-advanced networks," *IEEE Trans. Veh. Technol.*, vol. 65, no. 4, pp. 2659–2672, Apr. 2016.

[55] W. Stallings, *Cryptography and Network Security: Principles and Practice*, 7th ed. London, U.K.: Pearson, 2017.

[56] *Security Architecture for Open Systems Interconnection (OSI) for CCITT Applications*, T. S. S. International Telecommunication Union, Geneva, Switzerland, 1991.

[57] A. Ahmad, "Blockchain-driven secure and transparent audit logs," M.S. thesis, Univ. Central Florida, Orlando, FL, USA, 2019.

[58] L. D. K. Chuen and L. Linda, *Inclusive FinTech: Blockchain, Cryptocurrency and ICO*. Singapore: World Scientific, 2018.

[59] D. Aggarwal, A. Joux, A. Prakash, and M. Santha, "A new public-key cryptosystem via Mersenne numbers," in *Proc. Annual Int. Cryptol. Conf.* Springer, 2018, pp. 459–482.

[60] M. Kleppmann, *Designing Data-Intensive Applications: The Big Ideas Behind Reliable, Scalable, Maintainable Systems*. Sebastopol, CA, USA: O'Reilly Media, 2017.

[61] OMNeT++, *INET Framework Manual—Chapter 9: The 802.11 Model*. Singapore: World Scientific, 2016.

[62] S. Iskounen, T. M. T. Nguyen, and S. Monnet, "WiFi-direct simulation for INET in OMNeT++," 2016, *arXiv:1609.04604*.

[63] S. Mandal, S. Mohanty, and B. Majhi, "CL-AGKA: Certificateless authenticated group key agreement protocol for mobile networks," *Wireless Netw.*, vol. 26, no. 4, pp. 3011–3031, May 2020.

[64] H. Tan and I. Chung, "Secure authentication and key management with blockchain in vanets," *IEEE Access*, vol. 8, pp. 2482–2498, 2019.

[65] P. Bagga, A. K. Sutrala, A. K. Das, and P. Vijayakumar, "Blockchain-based batch authentication protocol for Internet of Vehicles," *J. Syst. Archit.*, vol. 113, Feb. 2021, Art. no. 101877.

[66] L. Zhang, Q. Wu, J. Domingo-Ferrer, B. Qin, and Z. Dong, "Round-efficient and sender-unrestricted dynamic group key agreement protocol for secure group communications," *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 11, pp. 2352–2364, Nov. 2015.

[67] T. Chen, L. Zhang, K.-K.-R. Choo, R. Zhang, and X. Meng, "Blockchain-based key management scheme in fog-enabled IoT systems," *IEEE Internet Things J.*, vol. 8, no. 13, pp. 10766–10778, Jul. 2021.

[68] Y.-M. Tseng, C.-C. Yang, and D.-R. Liao, "A secure group communication protocol for Ad hoc wireless networks," in *Advances in Wireless Ad Hoc and Sensor Networks* (Signals and Communication Technology). New York, NY, USA: Springer, pp. 102–130, 2007. [Online]. Available: https://shorturl.at/cetu0

[69] P. Raj and G. C. Deka, *Blockchain Technology: Platforms, Tools and Use Cases*, vol. 111. New York, NY, USA: Academic Press, 2018.
[70] R. Yang, F. R. Yu, P. Si, Z. Yang, and Y. Zhang, "Integrated blockchain and edge computing systems: A survey, some research issues and challenges," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 2, pp. 1508–1532, 2nd Quart., 2019.
[71] O. Alkadi, N. Moustafa, B. Turnbull, and K.-K.-R. Choo, "A deep blockchain framework-enabled collaborative intrusion detection for protecting IoT and cloud networks," *IEEE Internet Things J.*, vol. 8, no. 12, pp. 9463–9472, Jun. 2021.
[72] H. T. Le, H. T. T. Pham, H.-C. Le, and N. T. Dang, "Satellite quantum key distribution for vehicular visible light communication networks," in *Proc. IEEE 8th Int. Conf. Commun. Electron. (ICCE)*, Jan. 2021, pp. 45–50.

SHU-PING LU received the M.S. degree in electrical engineering and computer engineering from the National Taiwan University of Science and Technology, Taipei. She is currently pursuing the Ph.D. degree with the Department of Electrical Engineering, National Taiwan University, Taiwan. She works at the Hsinchu Industrial Technology Research Institute, Taiwan. Her research interests include network security, mobile and wireless networks, computer networks, service systems, and imaging processing.

CHIN-LAUNG LEI received the B.S. degree in electrical engineering from the National Taiwan University, Taipei, in 1980, and the Ph.D. degree in computer science from The University of Texas at Austin, in 1986. From 1986 to 1988, he was an Assistant Professor with the Computer and Information Science Department, The Ohio State University, Columbus. In 1988, he joined as a Faculty Member of the Department of Electrical Engineering, National Taiwan University, where he is currently a Professor. He has published more than 250 technical articles in scientific journals and conference proceedings. His current research interests include network security, cloud computing, the Internet of Things, and big data analytics. He is a Co-Winner of the first IEEE LICS Test of Time Award.

CHENG-YUN HO received the Ph.D. degree in computer science and engineering from the National Chiao Tung University, Taiwan, in 2015. He is currently a Senior Engineer with the Information and Communications Research Laboratories, Industrial Technology Research Institute, Hsinchu City, Taiwan. His research interests include P2P networking, computer networks, network protocols, and mobile and wireless networks.

SHY-SHANG HWANG received the B.S. degree in electronic engineering from Chung Yuang University and the M.S. degree in electrical engineering from the National Cheng Kung University, Taiwan. He is currently with the Information and Communications Research Laboratories, Industrial Technology Research Institute, Taiwan. His research interests include multimedia, distributed computation, computer graphics, user-interface design, image processing, computer networks, and mobile and wireless networks.

HSIN-CHEN CHEN received the M.S. degree in electrical engineering and computer engineering from the National Taiwan University of Science and Technology, Taipei, Taiwan. He worked at the Hsinchu Industrial Technology Research Institute, Taiwan. He is currently joining Phison Electronics Corporation. His research interests include cloud data at centers, computer networks, and mobile and wireless networks.