Permissioned Blockchain and Edge Computing Empowered Privacy-Preserving Smart Grid Networks

Keke Gai, *Member, IEEE*, Yulu Wu, Liehuang Zhu, *Member, IEEE*, Lei Xu, and Yan Zhang, *Senior Member, IEEE*

***Abstract*—The blooming trend of smart grid deployment is engaged by the evolution of the network technology, as the connected environment offers various alternatives for electrical data collections. Having diverse data sharing/transfer means is deemed an important aspect in enabling intelligent controls/governance in smart grid. However, security and privacy concerns also are introduced while flexible communication services are provided, such as energy depletion and infrastructure mapping attacks. This paper proposes a model permissioned blockchain edge model for smart grid network (PBEM-SGN) to address the two significant issues in smart grid, privacy protections, and energy security, by means of combining blockchain and edge computing techniques. We use group signatures and covert channel authorization techniques to guarantee users' validity. An optimal security-aware strategy is constructed by smart contracts running on the blockchain. Our experiments have evaluated the effectiveness of the proposed approach.**

***Index Terms*—Edge computing, permissioned blockchain, privacy-preserving, smart grid network (SGN).**

I. Introduction

SMART grid, as an emerging technology, has been driven by recent flourishing development of the network, in both wired and wireless environments. The interconnection-based setting makes energy service offerings varied, which involves a large number of distinct edge infrastructure or devices [1]–[3]. smart grid network (SGN) is a platform on which all electrical appliances, smart meters, and other energy-related embedded systems are joined [4]. Multiple electric sources and various user types can be configured to be network nodes in a typical SGN [5]–[7]. Thus, governance and optimization can be achieved by utilizing network features and service models.

Similar to other network-enabled solutions, vulnerabilities of networks also impact on the implementation of SGNs [8]. One of the recent power grid attacks took place in Ukraine in December 2015, which caused a large scope of energy users (more than 200 thousand) losing electricity supplies [9]. The consequence of the Ukrainian incident depicts that cyber attacks restrict the network's functionality in many application scenarios. Considering the service content, energy supply is the crucial attack objective, such as power theft and power loss. sleep deprivation torture (SDT) or battery exhaustion attack (BEA), for example, are classic malicious actions targeting at drying victims' energy [10], [11].

We observe that common threats mostly can be categorized into three layers, which include edge [2], [12], communication [13], and power plants [14] layers. Positioning threats are related to the complexity of launching attacks. The centralized datacenter generally is not a common attack objective, since launching an attack will be much easier at edge and communication layers than at a well protected cloud server. More specifically speaking, an edge layer mainly refers to those vulnerabilities in data collections, as sensors are routinely deployed at network edge. Attackers may reach those sensors without any difficulty so that physical abuse maybe applied. It is common for smart grid systems to keep a large amount of access points such as smart meters. Defending all potential adversaries at all access points is a remarkable tough job [15].

Next, when speaking to threats at the communication layer, attackers can fool central controllers or datacenter by tampering messages or disabling communication channels. A representative malicious activity is that attackers deploy a number of adversarial edge nodes (ENs) and launch a distributed denial of service (DDoS) attack [16]. A mix attack can further enhance the efficiency of adversaries [13], [17]. Consequence of the attack may not only take over control channels but also produce malicious commands.

Power plants are a special adversarial target existed in smart grid, even though the attacking complexity generally is high. A few types of power plants are even considered the critical infrastructure, such as nuclear power plants and hydroelectric power [18]. Many contemporary power plants are attached to a cloud setting due to the widely adopted central datacenter. The return of attacks will be excessive once the malicious action proceeded. Adversarial targets include a broad scope, from control systems to communications, from hardware to software.

Manuscript received October 2, 2018; revised January 31, 2019; accepted February 28, 2019. Date of publication March 11, 2019; date of current version October 8, 2019. This work was supported by the National Natural Science Foundation of China under Grant 61872041, Grant U1836212, and Grant 61871037. The work of K. Gai and L. Xu was supported in part by the Beijing Institute of Technology Research Fund Program for Young Scholars. *(Corresponding author: Liehuang Zhu.)*

K. Gai, Y. Wu, L. Zhu, and L. Xu are with the School of Computer Science and Technology, Beijing Institute of Technology, Beijing 100081, China (e-mail: gaikeke@bit.edu.cn; 2120171080@bit.edu.cn; liehuangz@bit.edu.cn; 6120180029@bit.edu.cn).

Y. Zhang is with the Department of Informatics, University of Oslo, Norway, and also with the Simula Metropolitan Center for Digital Engineering, Norway (e-mail: yanzhang@ieee.org).

Digital Object Identifier 10.1109/JIOT.2019.2904303

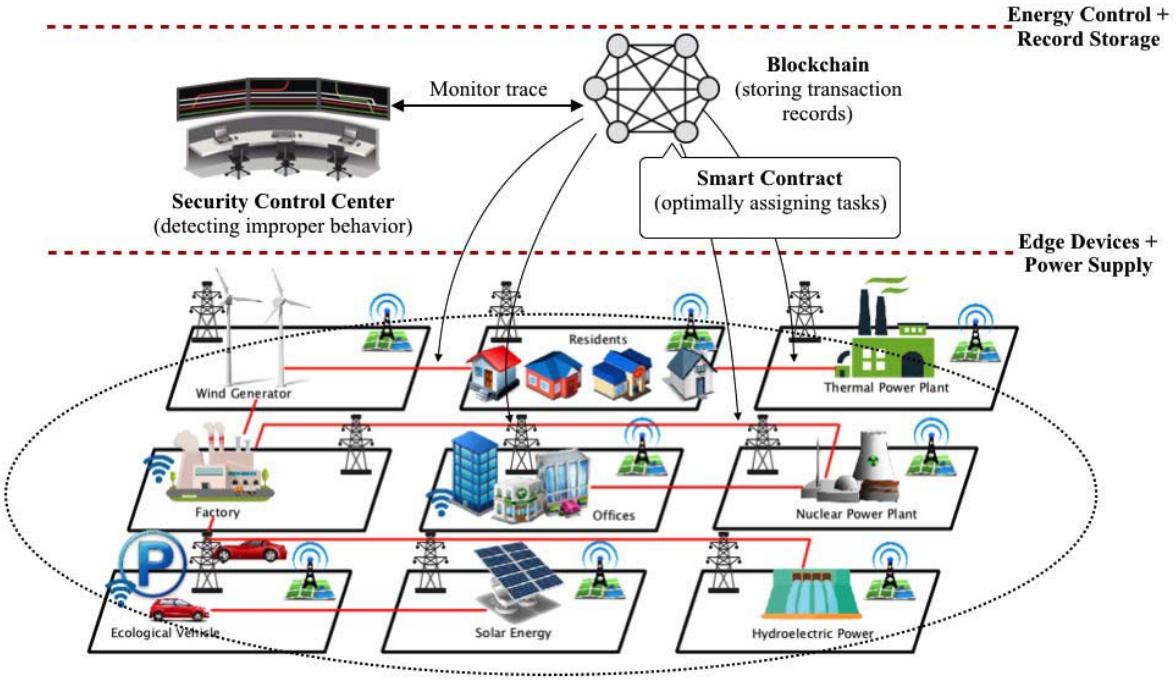


Fig. 1. High level architecture of privacy-preserving smart grid using blockchain-based edge computing.

In this paper, we aim to design an adoptable energy management approach for SGN. In essence, the proposed approach will address a few primary issues in the three layers above, although it does not cover all vulnerabilities. A brief description about our design objectives are presented here.

1) Ensures that energy usage is traceable and transparent without leaking end users' identities, so that improper behaviors of energy consumptions are detectable.
2) Provides optimal business logic for energy governance when a latency tolerance is settled.
3) Empowers controllers to evaluate the identity of the edge device in SGNs.

Fig. 1 illustrates a high level architecture of the proposed approach, called permissioned blockchain edge model for SGN (PBEM-SGN). Our approach introduces permissioned blockchain into the scenario of SGN. Edge computing is a deemed deployment method of network nodes for constructing a blockchain system. At the layer of edge devices and power supply, smart devices and power supply facilities compose smart grid generating electricity trading transactions. At the layer of energy control and record storage, security control center monitors transactions of electricity trading and detects improper behavior of smart grid. While smart contracts in blockchain optimally assign tasks to edge devices and storing electricity trading transaction records in blocks to ensure a secure and trustworthy trading environment. As a decentralized storage technique, blockchain offers a core value for recording the approved transactions in the tamper proof storage. Combining with edge computing, blockchain's values are broadened to a wide range of services from pure data storage, such as device configuration and governance, sensor data storage and management, and multiaccess payments. The convergence of blockchain and edge computing not only strengthens the existing benefits from two sides but also creates new services with additional values. This is the primary standpoint of designing the edge empowered blockchain system in SGN.

Main contributions of this paper are summarized in the followings.

1) This paper proposes a novel solution to designing a traceable energy governance in SGN. Transparent operations in the proposed model assist in detecting improper energy usage behavior in order to reduce/avoid energy-related attacks. We present a method of combining permissioned blockchain with edge computing to govern and map network nodes of SGN.
2) This paper utilizes smart contract attached to the permissioned blockchain system to achieve optimal energy resource management. We configure a few super nodes (SNs) in blockchain system to be responsible for organizing resource allocations and use the voting functionality of blockchain to validate the users' identities. Three crucial elements are considered in the decision-making that include execution time, security performance, and energy consumption.

We organize the rest of this paper by the following order. A review on related work is provided in Section II. Next, we present the model design and core proposed algorithms in Sections III and IV, respectively. Moreover, evaluation results as well as analysis are given in Section V. Finally, Section VI draws a conclusion of this paper.

II. Related Work

Deploying permissioned blockchain systems with edge computing was a novel alternative for strengthening SGN's

functionality and security. Observable benefits of using edge computing mostly derived from the reduced overall network traffics and enhanced edge resource governance [19]–[21]. As a reinforcing party, blockchain technique further avoided a few critical security and privacy concerns due to its decentralized features [22]. One crucial fact is that blockchain technique provided edge computing with a decentralized storage, such that a traceable and tamper-resistant could be attained. In this section, we had summarized and reviewed relevant work in order to provide theoretical supports for the proposed work.

The deployment of multiaccess techniques could solve the complicated authorizations caused by diverse network participants. This directs the edge system to become a larger setting. Users could be categorized into a few groups such that different access strategies were applied, such as attribute-based and policy-based access controls. Lin *et al.* [23] presented a novel access control method to protect data privacy on the cloud. Seol *et al.* [24] proposed an EHR model-based cloud accepting attribute-based access control to guarantee patients' privacy. Liu *et al.* [25] proposed a role-based access control model for certain users' permission assigning request based on certain role to guarantee security of data sharing in manufacturing Internet of Things. Chatterjee *et al.* [26] supported identity authentication for users and provided certain service for authorized users based on their access control model protecting secure information exchange in a telecare medicine information system. Using a semantic-based technique could even achieve pro-active access control in a specific application scenario [27]. Zhang *et al.* [28] explored the reduction of the complexity by providing predictive offloading. The optimization took place at the decision-making sector.

Edge computing is a layer that consists of a group of sensors for data collections, some of which could offer computation capabilities [29]. Liu *et al.* [30] attempted to utilize the benefits of edge computing for mitigating the computation workload of blockchain, e.g., computation offloading and content caching. This paper had measured the performance of blockchain systems when an edge-enabled computation was available. Findings showed that utilizing idle edge computing resources could effectively increase the efficiency of blockchain implementations. Other studies [31], [32] had retrieved similar conclusions, which argued that offloading workload by edge computing was superior when the allocation/governance strategy was effective. Using edge-enabled applications could mostly match the requirement of real-time services.

Due to the characteristic of decentralization, edge computing was instinctively suitable for combining with distributed techniques [33]. Besides, security and trustworthy of blockchain is vital to many researches solving many security problems. Li *et al.* [34] used consortium blockchain technique to propose energy blockchain, a secure energy trading system, addressing security problems and avoiding the use of trusted mediations. Kang *et al.* [35] accepted consortium blockchain to ensure electricity transactions security. Various edge devices could be formulated into network nodes in a blockchain system. For example, Sharma *et al.* [36] had proved that applying blockchain in fog/edge computing could realize secure energy transactions because of the advantage of blockchain techniques in preserving privacy. Kang *et al.* [37] combined edge computing technique with vehicular network to provide huge computing and storage resources and accepted consortium blockchain to guarantee secure data storage and sharing among vehicular edge computing networks. The study had examined the flexibility of smart contract in resource management. Xiong *et al.* [38] have investigated the technical fusion of blockchain and edge computing in a dimension of mobile networks. Their findings pointed out that fusing two techniques could achieve a seamless integration for establishing a whole system. Another study [39] provided a use case of integrating edge computing and blockchain in enabling secure electric vehicles cloud. In this case, two types blockchain coins, blockchain-inspired data coins and energy coins, were proposed. The proof of work was assessed by both data contribution frequency and energy contribution amount.

SGN is considered a communication platform in which hosts either centralized or decentralized applications by using multiple networking interconnections. Power nodes are deployed in this networking system such that the manner of reaching connections can be built up not only by linking facilities with servers but also by fastening mobile users [40]. With the inherent feature of mobile applications, SGN gets more involvements from enabling untrusted authorities in the communications, which include the untrusted wireless network systems. Recent studies [41] found that centralized computation (cloud datacenter) was no longer an optimal alternative for some smart grid applications, since diverse electric devices and wide geographic electric embedded systems could result in energy wastes and latency time.

Compare with other types of networks, SGN has a few distinctive characteristics, according to the prior researches [42]. First, SMG is an intelligent network deployed in a dynamic networking environment. The mobility of the nodes in the network depends on humans' mobilities that are usually related to users' behaviors and the relative static networking environment. Other network types may have different focuses.

From the perspective of the network infrastructure, SGN was highly tied to wireless communications instead of the infrastructure deployment. The focused research issue usually addressed the application level in which the functionality was generally delivered by user-centric solutions in the wireless environment. The security concerns of using SGN also were associated with the application level. The risks existed in a variety of SGN communications when mobile users interconnect with each other, such as peer-to-peer, user-chain, and centralized connections [43]. This paper addressed one of the important aspects of SGN applications, which was to prevent the wireless communications from spoofing attacks. All participant network nodes would be identified by a two-layer authorization method.

Previous researches also have done a variety of explorations in SGN security. *K anonymity technique* has been proved as a type of communication protocols providing a low level security by requiring a weak anonymity [44]. Therefore,

many researches had focused on executing pseudonym techniques (PT) by introducing the trusted authorities to protect the users' identifies [45]. Two common methods of executing PT are *mapping functions* and *grouping signatures*, which are designed to against sybil attacks [46]. However, these approaches did not consider protecting the social network systems from the perspective of the wireless communications. This paper had a wide scope covering most types of networks.

In addition, prediction methods were considered as an effective approach for increasing the security level by weighting the input data [47]. This technique is based on examining a series of data to learn the hidden rules of the observed states in the system. For example, using coefficient parameters is an option to detect unusual behaviors. Meanwhile, cryptographic techniques were also explored by the prior researches, such as bilinear pairing [48] and identity-based aggregate signature [49]. Nevertheless, very limited researches concerned the risk of tampering records, even through privacy issues were addressed.

In summary, we found that contemporary SGN system was facing threats from three aspects, including edge malicious activities, communication interferences, and datacenter attacks. A crucial feature of current SGN was a decentralized system setting. Even though many previous studies had made progress in securing SGN, it still showed that single technique could hardly cover all risks. Decentralized deployment made intrusion detections dramatically hard due to a vast of access points. Therefore, it should be a suitable occasion to develop an approach that could cover both decentralized computing and tamper-resistant without lowering down privacy-preserving capability and SGN's functionality. According to our survey, edge computing was a fitting solution to avoiding energy waste and latency time as well as a decentralized-friendly system setting. Combining with blockchain system could further increase the capability of edge computing from preserving privacy, tamper-resistant, and transparent transactions aspects. The next section will present the design of the proposed model.

III. Proposed Model

A. Model Design

In our proposed model, the core component is supported by deploying a permissioned blockchain system. The primary motivation of adopting blockchain technique is to preserve privacy. All participant nodes (users) in a blockchain system can be identified by pseudo names, so that a direct connection between privacy and user identities will be avoided.

Moreover, three crucial layers are involved in the proposed blockchain system, which include ENs, SNs, and smart contract layers. The three-layer blockchain setting is designed to guarantee the correctness and trustworthiness for both involved network nodes and voting results. Fig. 2 presents main activities taken place at each layer in the proposed blockchain system. Three participants concerning permissioned nodes, include SNs, ENs, and smart contract servers (SCSs).

In our model, an SN is defined as a class of network nodes for validating those ENs that participate in voting energy transactions. Compare with voting nodes in traditional blockchain system, an SN is a type of the special-purpose node that is constructed by offering additional authorization(s). Two methods are involved in the process of validating ENs, which are identity authorization and covert channel authorization (CCA). A detailed presentation about SN operations in our model will be described in Section III-C.

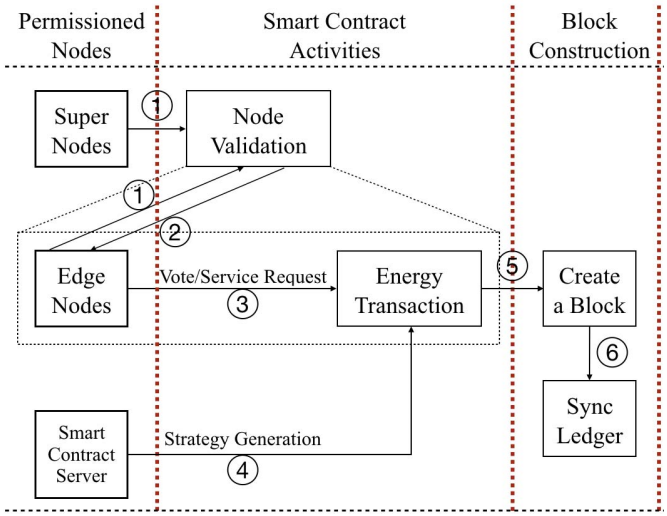


Fig. 2. Layered activities in the proposed permissioned blockchain system.

In addition, an EN is similar to a node in a classic blockchain system, which can be either a regular user or a voter. Considering the smart grid application scenario, ENs cover a wide scope of equipment deployed at the network edge, e.g., smart meter, energy measurement equipment, power sensor, etc. On one side, SNs need to validate an EN's identity when it asks for participating in a voting action. The validation is designed to again 51% attack by assuring most participants are "good men." On the other side, a user type of ENs requires energy service requests, such as processing energy transactions between battery bank and users. Transparent service records stored on the blockchain will be used for alarm harmful activities as well.

By using edge computing technique, ENs deal with edge computing problems like allocating power to electricity users in smart grid managed by ENs which will decrease the computation burden of this system. Smart contracts in blockchain need to assign tasks to ENs in an optimal way and using the characteristics of blockchain, privacy and security, to guarantee the privacy of ENs' identities and be easy to review the blockchain ledger to get the corresponding parameters which are set to select optimal node avoiding the wrong node selection and making the selected node more credible.

The SCS is a blockchain server in which smart contract is implemented. Our model considers the functionality of implementing blockchain technique such that an optimal strategy-making for energy-allocations will be made by smart contract. We design a dynamic programming to produce an optimal solution to energy resource allocation while concerning three elements, including energy consumption, latency time, and communication security. The core algorithm will be presented in Section IV.

B. Threat Model

Due to the power demand of users in smart grid, each EN needs to request power allocation from SN to satisfy their power demand. So it causes that some attackers pretend to be some valid ENs to illegally distribute power to users bringing additional financial burden to users, etc. So, in our PBEM-SGN model, we define a threat model resisting illegally distributing electricity model (RIDEM). In this model, we assume that attackers can achieve valid ENs' group signatures to assume to be a valid EN to allocate power to users.

C. Permissioned Blockchain System

A permissioned blockchain refers to a blockchain system using an authorization layer that determines the scope of users or voters and allows the target group scope to have the access to the system. Our model selects a permissioned blockchain as the data storage platform due to two principal reasons. First, as we mentioned in Section III-A, privacy-preserving ability offered by blockchain is a fundamental consideration in our model design. Second, a permissionless blockchain, based on our observation, is unsuitable for a smart grid energy trading scenario, since user groups (energy buyers and sellers) are mostly internal entities connected by SGNs with a relatively stable identity status. Having an open and public blockchain will increase threats from external attacks without a notable function enhancement.

An authorization layer is one of cores for permissioned blockchain comparing with other types of blockchain. We use both traditional access controls and a special access method, CCA, in our model. SNs are responsible for authorizing ENs. Traditional access control methods are applied in authorization, e.g., attribute-based authorization. This section mainly describes the additional authorization method (CCA). Fig. 3(a) exhibits an operation diagram of our proposed CCA.

As a cryptographic technique, covert channel attack is a type of adversarial approach that uses latency time to leak critical information. In our model, we utilize the thought of the covert channel communication to design an authorization method using latency time difference for validating identities. CCA can be deployed at an embedded system attached to ENs in an SGN, e.g., smart meters. The method of our CCA originally derive from an attack method [50]. In line with Fig. 3(a), there are mainly two parts in CCA, encoding message (EN sends ID; add a covert channel) and decoding operations (SN validates covert information, responds to EN). We present a phase-by-phase description in the following.

Phase I (Encoding Message): The process starts with sending a public ID from an EN to SNs. A public ID is a piece of information that is not deemed privacy; thus, it can be a random number for show-identity purpose only. We intend to attach covert information to this public message and only SNs have the decoding capability. The operation of encoding messages is implemented in a "Block Box," which assumes it is not reachable for attackers.

Assume that the covert message is a binary sequence ($\mathbf{E} = \{E_i\}$), so that it is an input of the encoding message phase.

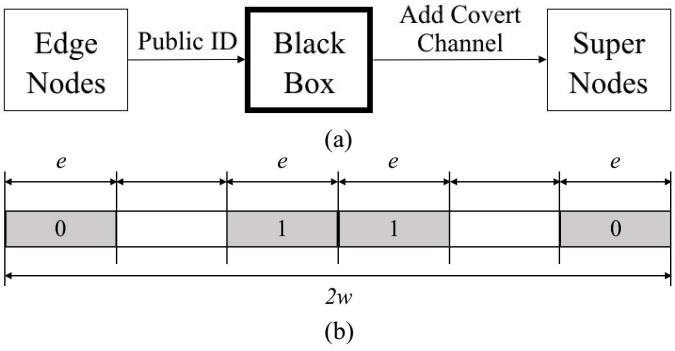


Fig. 3. CCAs. (a) Diagram of CCA. (b) Illustration of time window configuration for covert channel decoding.

We configure a discrete time set $\{t_i\}$ as an original time set. The goal is to create a noised new set $\{\mathcal{T}_i\}$ by adding latency time. Let the set $\{\Delta t_i\}$ be the noise set; thus, there exists $\mathcal{T}_i = t_i + \Delta t_i$. In addition, let a w be a parameter of the *time window*. From the perspective of the receiver (SN), the time interval of data arrivals is denoted by $\Delta\mathcal{T}_i$, where $\Delta\mathcal{T}_i = \mathcal{T}_i - \mathcal{T}_{i-1}$. The operation principle of encoding is shown in

$$\Delta\mathcal{T}_i \bmod w = \begin{cases} 0, & \text{if } E_i = 0 \\ \lfloor w/2 \rfloor, & \text{if } E_i = 1. \end{cases} \tag{1}$$

The effectiveness of this covert channel methods has been proved in the prior work [50] (as an attack method). In order to increase the complexity of detecting covert channel communications, we further introduce a time-based mapping function $m(x)$ that is cognized by both SNs and ENs. The attack complexity is increased as original time gaps are dynamically changed such that detecting change rules is hard. Assume that there exists a time scope S_t. The set $\{t_i\}$ can be created by implementing function $m(x)$ at a specific time slot t, denoted by (2). The value scope of each element in this set shall be not shorter than basic communication time that is retrieved from historical observations

$$\{t_i\} \leftarrow m(S_t), \text{ where } t \in S_t. \tag{2}$$

Similarly, another mapping function is added to construct a dynamic time window values. Mathematical expression is shown in

$$w \leftarrow W(S_t), \text{ where } t \in S_t. \tag{3}$$

A simple example is presented in here. Given a value to m, where $m = 10$ ms. Public information (open identity) is a "TestNo1." Our goal is to use covert channel communications to deliver a hidden message for proving EN's legal identity. Assume that at a specific time slot, the identity of an EN is "1001001," as the same length as the public information, so that the input sequence is $\mathbf{E} = \{1, 0, 0, 1, 0, 0, 1\}$. A set $\{t_i\}$ is created by implementing function $m(x)$, where $\{t_i\} = \{121, 75, 60, 151, 123, 121, 92\}$. Applying (1) will have a noised set $\{\mathcal{T}_i\}$, where $\{\mathcal{T}_i\} = \{126, 80, 60, 155, 130, 130, 95\}$, which determines time intervals between letters in TestNo1. A noise set then will be retrieved: $\Delta t_i = \{5, 5, 0, 4, 7, 9, 3\}$.

Phase I (Decoding Operation): The other important phase of CCA is to decode the covert information. A decoding operation is triggered when an SN receives the public information.

In practice, there maybe an error scope due to impacts made by the instant communication quality. An e is configured to be a tolerance parameter. We use a $\Delta T_i'$ to denote the actual time intervals. Fig. 3(b) a time window configuration with a tolerance parameter e for covert channel decoding operations. An observation suggests that the value scope of an e is [0, w/4]. Mathematical expression of decoding operations is shown in

$$\Delta T_i' \bmod w = \begin{cases} E_i = 0, & \text{if } -e < \Delta T_i' \leqslant e \\ E_i = 1, & \text{if } w/2 - e \leqslant \Delta T_i' < w/2 + e. \end{cases} \tag{4}$$

For example, align with the same example given in Phase I, assume that the an SN receives actual time gap set is {126, 81, 61.5, 156, 130.5, 131.5, 96.9}. Let an e be a 2. We apply (4) on this set so that the following set will be retrieved by implementing a mod operation, {6, 1, 1.5, 6, 0.5, 1.5, 6.9}. The covert information can be obtained, {1, 0, 0, 1, 0, 0, 1}. An SN will mark a request as an *illegal* when the hidden identity code is incorrect or there is no covert channel communication detected.

D. Edge-Enabled Energy Optimization Sector

In this section, we introduce an optimal solution to optimizing energy transactions in SGN. The proposed problem is called an instant energy transaction optimization (IETO) problem and its definition is given in Definition 1.

Definition 1 (IETO Problem): Inputs mainly include an input task; a set of edge computation nodes $\{N^j\}$; for each node N^j, it has an estimated time consumption $t_{i,e}^m$ and an estimated energy cost $E_{i,e}^m$; a timing constraint T_c.

Output will be a label of the computation node for allocating the input task.

The proposed problem is to find out an optimized task allocation solution that considers real-time states, energy-saving, and matching the given timing constraint.

In the definition, we present a number of parameters to abstract the optimization process. In the input, assume that the system has the nth incoming energy requests and our model will allocate the input task to one of the available power suppliers. The input task is denoted by T^n; the number of the total available power suppliers is M. For a power supplier, it offers an estimated energy delivery cost ($E_{i,e}^m$) and an estimated time consumption ($t_{i,e}^m$) for the task T^n. $E_{i,e}^m$ denotes the mth power supplier has an estimated energy waste for the ith state; t_e^m denotes the mth power supplier's estimated time at the ith state.

In order to reduce the impact of the estimated error, we introduce the thought of reinforcement learning (RL) to our model. An adjustment parameter is created to make up the estimated time length, which is denoted by γ. We formulate the concept of the state for describing an EN's general performance of accomplishing tasks. In another presentation, we adjust the estimated time consumption by using the EN's prior performance. For the nth state, we obtain the value of γ by implementing

$$\gamma_n = \gamma_{n-2} + \frac{1}{n} R_{n-1}. \tag{5}$$

The equation depicts the acquisition of the γ value when allocating the task at the nth state. Meanwhile, our model creates another parameter to represent the error for the purpose of the correctness adjustment, denoted by R. Thus, the value of γ is a result of iterations. Equation (6) shows the equation of acquiring R, where t_r denotes the real time consumption and t_e denotes the estimated time consumption. t_r is obtained from calculating the difference value between the service response arrival time and the service request arrival time

$$R = \frac{t_r - t_e}{t_e}. \tag{6}$$

Therefore, the value of R_{n-1} can be obtained by the following formulation:

$$R_n = \frac{t_{r_{n-1}} - t_{e_{n-1}}}{t_{e_{n-1}}}.$$

Moreover, our model also considers the energy cost. In order to concurrently measure both energy cost and time consumption, we examine the efficiency to obtain the optimal allocation, as shown in (7). The efficiency parameter is denoted by $\mathbb{E}$

$$\mathbb{E} = \frac{E}{t_a} = \frac{E}{t_e + \gamma}. \tag{7}$$

In the equation, t_a refers to an adjusted time value that derives from $(t_e + \gamma)$. In line with (1) and (2), we present the acquisition equation of $\mathbb{E}$ in

$$\mathbb{E}_n = \frac{E_{i,e_n}^m}{t_{i,e_n}^m + \gamma_n}. \tag{8}$$

Therefore, the objective of IETO problem can be presented in the following mathematic expression, (9). Assume that the selected edge source N^s and its estimated energy cost is $\mathbb{E}^s$

$$\mathbb{E}^s = \mathrm{MIN}\left[E_{i,e_n}^m / \left(t_{i,e_n}^m + \gamma_n\right)\right]$$

while

$$T_c \geqslant t_a. \tag{9}$$

The objective function of the proposed optimization problem is presented in the next section.

Align with the problem definition above, we provide an objective function in

Assume that there exist functions $f\left(N^j\right)$ and $g(x)$:

$$f\left(N^j\right) = E_{i,e_n}^m / \left(t_{i,e_n}^m + \gamma_n\right)$$

$$g\left(f\left(N^j\right)\right) = t^e + \gamma_n^m$$

The objective is to find out a binary function $s(j)$

$$\text{have } f\left(N^a\right)_{s(i)=1} \leqslant f\left(N^j\right)_{s(i)=1}$$

$$\text{and } g\left(f\left(N^j\right)\right) \leqslant T_c. \tag{10}$$

The binary function $s(i)$ is a selection function, by which EN is selected when $s(i) = 1$. Thus, the expression $f(N^a)_{s(i)=1} \leqslant f(N^j)_{s(i)=1}$ means the minimum energy cost when a certain

EN is selected. $g(f(N^j)) \leqslant T_c$ means the execution time is no longer than the timing constraint.

One of the roles of the smart contract, in our model, is to discriminate performance between ENs, based on the ENs' estimated values and historical records. It is an executor for allocating tasks. In addition, an ART is a table for storing iteration results for the purpose of adjusting estimated time. There is generally a gap between the estimated time length and real-world performance due to various hardware or network conditions. This table is designed for reducing the impact caused by the gap. Finally, blockchain system consists of a chain of blocks. Our model designs functionalities for blocks for the purpose of the traceability and adjustments.

There are mainly four phases in our model, which are preparation, task allocation, allocation block creation, and performance block creation (PBC). The description of each phase is given in the followings.

Phase I (Preparation): This phase assumes the duty of the information retrieval. All ENs are noticed when a new task is inputted. Task header includes basic information for ENs to estimate costs, such as task type, task size, and request arrival time. Each EN sends two estimated values to the SCS of blockchain based on the header information, which include the estimated energy cost and time length.

Phase II (Energy Transaction Allocation): This phase bears the responsibility for making decisions on task allocations, which relies on both ENs' estimated values and their prior performance. It is a crucial process in which the optimization strategy is made. We consider both energy cost and time consumption so that the energy cost in a unit time is measured for assessment.

The first step of task allocation is that the SCS recalls an ART in order to make adjustments on the estimated time lengths sent from ENs. Our model does not adjust the estimated energy costs. In addition, in order to obtain an optimal solution, we implement a greedy algorithm to assess $\mathbb{E}$s of all ENs. The EN with a minimum $\mathbb{E}$ value will be selected for the task allocation.

Phase III [Energy Transaction Block Creation (ETBC)]: This phase undertakes to create a chained block for storing information that participates in the decision-making of the task allocation. Main job at this phase is to construct a block based on the allocation strategy. The block packs up the decision of the allocated EN as well as its estimated time consumption. The reason for storing the information is twofold: 1) making the allocation traceable and 2) recording data for the requirement of the future adjustment. Concurrently, a task is allocated to the selected EN according to the allocation strategy.

Phase IV (Performance Block Creation): At this phase, our model creates another block for assessing the gap between the real-time and estimated data. The SCS will receive a service response once the task is completed. A new block will be created for packing up the parameters collected from the real performance, including values of $\mathbb{R}$ and γ. The reason for storing real-time data is to support adjustments for future allocations.

IV. Proposed Algorithms

A. Edge Nodes Identity Validation Algorithm

User identity validation [edge nodes identity validation (ENIV)] algorithm is designed for validating ENs' identities by an SN using CCA. Before EN's identity validation, it must register to the system using group signature algorithm [51] to ensure their credibility. So, we first state the initialization algorithm of system using group signature algorithm. The initialization algorithm consists of five parts.

SystemSetup: First, given a security parameter $k \in \mathbb{Z}^+$, generate a k bit prime number p and three multiplicative cyclic groups G_1, G_2, G_T of the same prime order p. Let $g_1 \in G_1$ and $g_2 \in G_2$ be two generators for G_1 and G_2, respectively. ψ is a computable isomorphism from G_2 to G_1, where $\psi(g_2) = g_1$ and a pairing relationship $\hat{e} : G_1 \times G_2 \to G_T$ be a bilinear map which is generated by these three multiplicative cyclic groups G_1, G_2 and G_T. Also, employ $H : \{0, 1\}^* \to \mathbb{Z}_p$ in our algorithm.

KeyGen: Take n the number of edge devices as an input of our signature algorithm to generate keys when edge devices send their public IDs, a kind of physical signals, to SN to guarantee the validity of their identities. Set a random number $h \in G_1 \backslash \{1_{G_1}\}$ and two random numbers $\xi_1, \xi_2 \in \mathbb{Z}_p^*$, and set $u, v \in G_1$ satisfying $u_{\xi_1} = v_{\xi_2} = h$. Select a random number $\tau \in \mathbb{Z}_p^*$ and set $w = g_2^{\tau}$. For each EN i, $i \in n$, generate an SDH tuple (A_i, x_i), where the random number $x_i \in \mathbb{Z}_p^*$, $A_i \in G_1$ and ${A_i}^{\tau+x} = g_1$. So, the group public is $gpk = (g_1, g_2, h, u, v, w)$, and SN's (the group manage to trace signatures) private key is $gmsk = (\xi_1, \xi_2)$. Each EN's private key is the SDH tuple $gsk[i] = (A_i, x_i)$.

Sign: Given the group public key $gpk = (g_1, g_2, h, u, v, w)$, an EN's key $gsk[i] = (A_i, x_i)$ and a message M, where $M \in \{0, 1\}^*$. Generate two random numbers $\alpha, \beta \in \mathbb{Z}_p$ and compute a series of values: $T_1 = u^{\alpha}$, $T_2 = v^{\beta}$, $T_3 = A_i h^{\alpha+\beta}$, $\delta_1 = x\alpha$ and $\delta_2 = x\beta$. Select a series of random numbers $r_{\alpha}, r_{\beta}, r_x, r_{\delta_1}, r_{\delta_2} \in \mathbb{Z}_p$ to achieve $R_1 = u^{r_{\alpha}}$, $R_2 = v^{r_{\beta}}$, $R_3 = e(T_3, g_2)^{r_x} \cdot e(h, w)^{-r_{\alpha}-r_{\beta}} \cdot e(h, g_2)^{-r_{\delta_1}-r_{\delta_2}}$, $R_4 = T_1^{r_x} \cdot u^{-r_{\delta_1}}$ and $R_5 = T_2^{r_x} \cdot v^{-r_{\delta_2}}$. Obtain a challenge $c = H(M, T_1, T_2, T_3, R_1, R_2, R_3, R_4, R_5) \in \mathbb{Z}_p$, where $H : \{0, 1\}^* \to \mathbb{Z}_p$. Then compute $s_{\alpha} = r_{\alpha} + c\alpha$, $s_{\beta} = r_{\beta} + c\beta$, $s_x = r_x + cx$, $s_{\delta_1} = r_{\delta_1} + c\delta_1$ and $s_{\delta_2} = r_{\delta_2} + c\delta_2$. So, the signature $\sigma_i = (T_1, T_2, T_3, c, s_{\alpha}, s_{\beta}, s_x, s_{\delta_1}, s_{\delta_2})$.

Verify: Given the group public key $gpk = (g_1, g_2, h, u, v, w)$, a message M and a relative secret signature σ_i, compute $\tilde{R_1} = (u^{s_{\alpha}}/T_1^c)$, $\tilde{R_2} = (v^{s_{\beta}}/T_2^c)$, $\tilde{R_3} = e(T_3, g_2)^{s_x} \cdot e(h, w)^{-s_{\alpha}-s_{\beta}} \cdot e(h, g_2)^{-s_{\delta_1}-s_{\delta_2}} \cdot (e(T_3, w)/e(g_1, g_2))^c$, $\tilde{R_4} = (T_1^{s_x}/u^{s_{\delta_1}})$, and $\tilde{R_5} = [T_2^{s_x}/(v^{s_{\delta_2}})]$. Then check EN's signature σ_i by comparing the value of $H(M, T_1, T_2, T_3, \tilde{R_1}, \tilde{R_2}, \tilde{R_3}, \tilde{R_4}, \tilde{R_5})$ with c. Only the value equals to c, the signature σ_i is valid.

Open: This part is used to trace a signature of a signer to verify his/her identity. Given the group public key $gpk = (g_1, g_2, h, u, v, w)$, a message M, the group manager SN's private key $gmsk = (\delta_1, \delta_2)$ and the secret signature σ_i. First, verify that σ_i is a valid signature on the message M. Then recover the EN's A_i, where $A_i = [T_3/(T_1^{\delta_1} \cdot T_2^{\delta_2})]$ by considering T_1, T_2, T_3 as a linear encryption. SN can recover ENs'

Algorithm 1 ENIV Algorithm

Require:
The group public key $gpk = (g_1, g_2, h, u, v, w)$;
A secret signature $\sigma_i = (T_1, T_2, T_3, c, s_\alpha, s_\beta, s_x, s_{\delta_1}, s_{\delta_2})$ of an EN;
A message M from an EN;
A binary sequence ($E = E_i$) (a public ID) from an EN;
A discrete time set t_i;
A time window w;
A binary sequence ($E = E_i'$) stored in SN;
Tolerance parameter e =[0, w/4];

Ensure:
A EN's identity validation result A

EN sends its signature σ_i and a message M to SN
SN computes $\tilde{R_1} = \frac{u^{s_\alpha}}{T_1^c}$, $\tilde{R_2} = \frac{v^{s_\beta}}{T_2^c}$, $\tilde{R_3} = e(T_3, g_2)^{s_x} \cdot e(h, w)^{-s_\alpha - s_\beta} \cdot e(h, g_2)^{-s_{\delta_1} - s_{\delta_2}} \cdot (\frac{e(T_3, w)}{e(g_1, g_2)})^c$, $\tilde{R_4} = \frac{T_1^{s_x}}{u^{s_{\delta_1}}}$ and $\tilde{R_5} = \frac{T_2^{s_x}}{v^{s_{\delta_2}}}$
if $c = H(M, T_1, T_2, T_3, \tilde{R_1}, \tilde{R_2}, \tilde{R_3}, \tilde{R_4}, \tilde{R_5})$ **then**
    the signature σ_i is valid
    if the binary sequence $E_i = 0$ **then**
        /*encoding phase*/
        time interval of data arrivals $\Delta\mathcal{T}_i$ mod time window $w = 0$
    else
        time interval of data arrivals $\Delta\mathcal{T}_i$ mod time window $w = \lfloor w/2 \rfloor$
    end if
    if time interval of data arrivals $\Delta\mathcal{T}_i = 0$ **then**
        the noised new set $\mathcal{T}_i$ = (discrete time t_i - discrete time set $\{t_i\}$ mod time-based mapping function $m(x)$) + time interval of data arrivals $\Delta\mathcal{T}_i$
    else
        the noised new set $\mathcal{T}_i$ = (discrete time t_i - discrete time set $\{t_i\}$ mod time-based mapping function $m(x)$) + time interval of data arrivals $\Delta\mathcal{T}_i$ + time-based mapping function $m(x)$
    end if
    the noise set Δt_i = noised new set $\mathcal{T}_i$ - discrete time t_i
    if $-e < \Delta\mathcal{T}_i' \leqslant e$ **then**
        /*decoding phase*/
        actual time intervals $\Delta\mathcal{T}_i'$ mod $w = E_i = 0$
    else
        actual time intervals$\Delta\mathcal{T}_i'$ mod $w = E_i = 1$
    end if
    if $E_i = E_i'$ **then**
        EN is a valid node, EN's identity validation result $A = T$
    else
        EN is an invalid node, EN's identity validation result $A = F$
    end if
else
    EN is an invalid node, EN's identity validation result $A = F$
end if
return A EN's identity validation result A

identities from their signatures when given A_i of ENs' private keys.

Algorithm 1 shows pseudo codes of ENIV algorithm. Main inputs of this algorithm consists of the group public $gpk = (g_1, g_2, h, u, v, w)$, an EN'S secret signature $\sigma_i = (T_1, T_2, T_3, c, s_\alpha, s_\beta, s_x, s_{\delta_1}, s_{\delta_2})$, a message M and a binary sequence ($E = E_i$) (a public ID) from an EN, a discrete time set t_i, a time window w, a binary sequence ($E = E_i'$) stored in SN and tolerance parameter e =[0, w/4]. The output of this algorithm is the EN's identity validation result A.

Major steps are described in the followings.

1) The first step of ENIV algorithm is that EN sends its signature σ_i and a message M to SN to verify whether EN's signature is valid or not. SN computes $\tilde{R_1} = (u^{s_\alpha}/T_1^c)$, $\tilde{R_2} = (v^{s_\beta}/T_2^c)$, $\tilde{R_3} = e(T_3, g_2)^{s_x} \cdot e(h, w)^{-s_\alpha - s_\beta} \cdot e(h, g_2)^{-s_{\delta_1} - s_{\delta_2}} \cdot (e(T_3, w)/e(g_1, g_2))^c$, $\tilde{R_4} = (T_1^{s_x}/u^{s_{\delta_1}})$, and $\tilde{R_5} = (T_2^{s_x}/v^{s_{\delta_2}})$ to check whether challenge c equals to the result of $H(M, T_1, T_2, T_3, \tilde{R_1}, \tilde{R_2}, \tilde{R_3}, \tilde{R_4}, \tilde{R_5})$. If c equals to the result, the signature σ_i is valid.
2) Then ENIV algorithm is used to encode EN's public ID by EN and decode EN's public ID by SN to validate whether EN is a valid node. At the encoding phase, if the EN's public ID, a binary sequence, E_i equals to 0, time interval of data arrivals ΔT_i mod time window $w = 0$. Otherwise, time interval of data arrivals ΔT_i mod time window $w = \lfloor w/2 \rfloor$. If time interval of data arrivals $\Delta T_i = 0$, the noised new set T_i = [discrete time t_i – discrete time set $\{t_i\}$ mod time-based mapping function $m(x)$] + time interval of data arrivals ΔT_i. Otherwise, the noised new set T_i = (discrete time t_i – discrete time set $\{t_i\}$ mod time-based mapping function $m(x)$) + time interval of data arrivals ΔT_i+ time-based mapping function $m(x)$. And the noise set Δt_i = noised new set T_i – discrete time t_i. At the decoding phase, if $-e <$ actual time intervals $\Delta\mathcal{T}_i' \leqslant e$, actual time intervals $\Delta T_i'$ mod $w = E_i = 0$. Otherwise, actual time intervals $\Delta T_i'$ mod $w = E_i = 1$. Unless $E_i = E_i'$ stored in SN, the EN is a valid node.

In summary, from ENIV algorithm, this validation algorithm first checks EN's signature's validation, then verifies EN's identity by comparing E_i and E_i'. Only if these twofold parts are satisfied, EN is a valid node.

B. Energy Transaction Block Creation Algorithm

ETBC algorithm is designed for implementing smart contract, which constructs a block for recording each energy transaction strategy in SGN. The purpose of creating this block is twofold. The first intention is to create a energy transaction strategy while store it to a block in order to make the transaction traceable. Second, the block created by implementing this algorithm also records the estimated time and energy costs sent from the EN, so that the estimated values can be used for improving RL model, as discussed in Section III-D.

Algorithm 2 presents pseudo codes of ETBC algorithm. Main inputs of this algorithm include an input task, denoted by $M[q]$, as shown in pseudo codes. A parameter set γ is also involved in the input, denoted by $\gamma[n]$. The outputs of ETBC algorithm include a new created block that packs up an optimal node a, an estimated time $t_e[n]$, and an estimated energy cost $E[n]$.

Major steps are aligned with the allocation block creation phase of our model. We present main phases of ETBC algorithm in the followings.

1) The first phase of ETBC algorithm is that the blockchain node administrator receives the estimated time and estimated energy costs from all connected ENs, once an

Algorithm 2 ETBC Algorithm

Require: Input task (M[q]), γ[n]
Ensure: An optimal node a, t_e[n], E[n]

```
Blockchain administrator reads γ[n]
for i=0; i<q; i++ do
    for j=0; j<n; j++ do
        Each edge node sends the estimated time t_ej
        Each edge node sends the estimated energy cost E_ej
        /*Estimated values are sent to blockchain administrator*/
        ∀ edge nodes, send out a broadcast request of packing up t_ej and E_ej on blockchain system via smart contract
        Calculate T_j ← t_ej + γ_j; 𝔼_j ← E_j/T_j (Node administrator)
    end for
    for j=0; j<n; j++ do
        Node administrator calculates 𝔼s and obtain the optimal node a
        Pack up t_e and E to a new block if miners agree with the transaction (t_e[n], E[n]).
    end for
end for
return a, t_e[n], E[n]
```

Algorithm 3 PBC Algorithm

Require: t_r[n], t_e[n]
Ensure: R[n], γ[n]

```
Node administrator reads prior iteration results and obtain prior estimated time t_e; read the real execution time t_r
for j=0; j<n; j++ do
    Calculate R_j = (t_rj − t_ej)/t_ej (Node administrator)
    R[n] ← R_j /*store R_j in R[n]*/
end for
for i=0; i<n; i++ do
    if i=0 then
        Calculate γ_i = 0; γ[n] ← γ_i /*Case 1*/
    else if i=1 then
        Calculate γ_i = R_0/n; γ[n] ← γ_i /*Case 2*/
    else
        Calculate γ_i = γ_{i−2} + R_{n−1}/n; γ[n] ← γ_i /*Case 3*/
    end if
    Packing up R and γ if miners agree with transaction.
    /* Administrator sends request to blockchain system for packing up R and γ via smart contract. */
end for
return R[n], γ[n]
```

input task is coming. The operations are broadcasted via the smart contract so that each EN will let node administrator know about its predictive performance for dealing with the incoming task.

2) Only selected EN's estimated values can be packed into a block. This algorithm utilizes the RL-based method to adjust the estimated time values. Thus, the node administrator recalls records of γ and a set of R from prior blocks to make up the estimated time values. For each EN, an $\mathbb{E}$ value for measuring energy cost in a unit time will be acquired.
3) An optimal EN a can be obtained through comparing all $\mathbb{E}$ values. Return a so that a task allocation strategy is released for edge computing. Meanwhile, the parameters of the selected EN will be packed to a new created block when all miners agree with the transaction.

In summary, this algorithm enables the blockchain system to create an energy transaction strategy via constructing a new block. Some data packed in the block will be used as inputs for PBC algorithm, such as estimated time, which guarantees γ can be updated timely.

C. Performance Block Creation Algorithm

Corresponding to the allocation block construction, we also propose a PBC algorithm. This algorithm is designed for updating γ and R according to the real performance made by the EN. Implementing this algorithm mainly supports the manipulation of the Phase IV in our model. Main inputs of PBC algorithm include real time length ($t_r[n]$) and the estimated time ($t_e[n]$). The acquisition of the real time consumption is obtained from calculating the time period between the service response arrival time and the service request arrival time. Primary intention of implementing PBC algorithm is continuously updating values of γ and R for future adjustments. Main outputs include a new block that records the updated R and γ.

We present pseudo codes of PBC algorithm in Algorithm 3. In line with the pseudo codes, there are three essential phases in PBC algorithm, as displayed in the following.

1) In order to recall prior estimated time length, the node administrator needs to read the reference table of prior iteration results. Meanwhile, the real execution time t_r can be obtained from calculating the gap between the service response arrival time and the service request arrival time. According to these acquisitions, an updated R can be obtained by implementing (6).
2) After gaining a new R, we need to obtain an updated γ for the demand of RL training. As depicted in pseudo codes, PBC considers three situations. Since at least two prior values are needed for creating a latest γ, cases 1 and 2 represent the initial condition. For most γ updates, it follows the case 3, after completing 2 allocation tasks.
3) A new block can be created when all miners agree with the transaction, so that the updated R and γ will be packed and recorded on the block.

Therefore, some data packed in the new created block also are inputs of ETBC algorithm, as the intention of creating a new block is to raise the quality of the input data for measuring ENs' capabilities. A cumulative iteration can efficiently reduce errors of the estimated time. In next section, we will exhibit some evaluation results and findings obtained from our experiments.

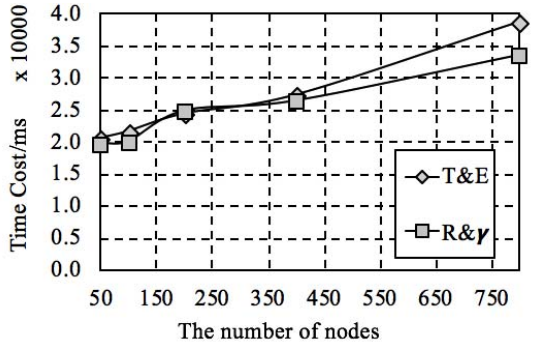


Fig. 4. Time cost for packing up data estimated time and energy consuming and R and γ.

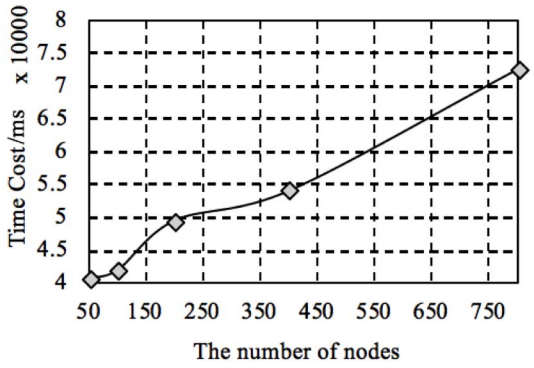


Fig. 5. Time cost for packing up data in blockchain system.

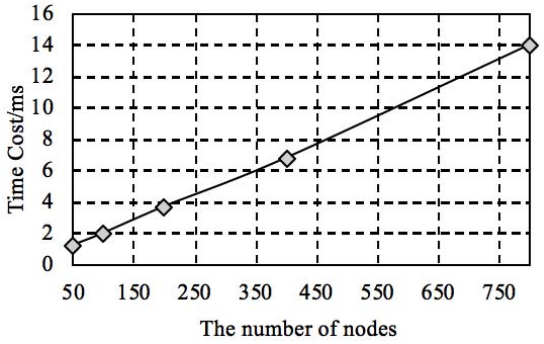


Fig. 6. Time cost for task allocation.

V. Evaluations

We implemented an experiment evaluation to assess the performance of our PBEM-SGN model. Energy-saving efficiency was not an emphasis in our evaluation, since our approach inherently had optimal solutions under the given criterion (energy cost counted in a unit time). In our evaluation, we emphasized the adaptability from the perspective of efficiency so that execution time was measured under various settings. We presented our experiment configuration and partial results in Sections V-A and V-B, respectively.

A. Experiment Configuration

The principle of configuring our experiments was simulating an edge-based SGN application. Applying edge computing in SGN system generally consisted of a limited number of edge devices (nodes), which primarily was deployed for a specific purpose/system. Hence, our proposed model had a smaller scope of the EN, which was distinct from a public-oriented blockchain system.

The software configuration included an Ethereum client Geth (1.8.3-stable) running on a computer (MacBook Pro 2017 version) as well as an Ethereum Wallet 0.10.0. The hardware configuration included an macOS 10.13.4 operating system, a CPU with 2.3 GHz, an i5 version Intel Core, and a memory with 8 GB of 2133-MHz LPDDR3. The deterministic programming for task allocations was written by C++ computing language that was running on an Xcode version 9.3.1.

The number of the EN was settled between 50 and 800, in order to simulate the scenario in which a certain number of edge devices were deployed. The exhibited four settings were associated with four parameters, which were estimated time, energy cost, R, and γ. Main measurement objectives included data packing-up time, task allocation time, and gas cost. We presented a few evaluation results in the next section.

B. Experiment Results

Due to page length limit, we demonstrated partial evaluation results gathered from our experiments in this section. We selected a number of representative cases that had a variety of amounts of ENs to depict the growing trend of the examined variable, which were 50, 100, 200, 400, and 800. In each case, we ran five rounds and counted average values for statistics.

Fig. 4 displayed results of the time length for packing up paired parameters to the block, which were estimated time (T) and energy consumption (E), and R & γ. According to our data collections, the time scope of the block creation for packing up T and E was within a period of 20–40 s; the time length of packing up R & γ was a little shorter than that of T and E, which was in a scope of 20–35 s. Both trends represented a linear growth, which depicted that the time costs had a positive relationship with the number of ENs.

Similarly, Fig. 5 further exhibited the total time cost of packing up all parameters in blockchain system. It also proved that the number of ENs had a direct impact on the time consumption for creating blocks after packing data and attaching the block to the chain.

Next, we found that the efficiency of the task allocation was acceptable. Fig. 6 showed time consumptions of the task allocation when various amounts of the EN were measured. The time cost was associated with the number of the EN, in this case. The execution time range was from 1 to 14 ms, which was an acceptable latency range for task allocations in edge computing.

Furthermore, we assessed gas costs under four settings and the corresponding results were presented in Figs. 7–10. In a private blockchain system, gas cost is a parameter for measuring the mining cost, which was generally attached to miner server(s). On the whole, we observed that the growth rates of the gas cost under four settings were remarkably close. An average growth rate was a 2.6×10^4, while the gas cost was counted in units. More specifically, Fig. 7 depicted that the time cost of packing up estimated time to the block. We could

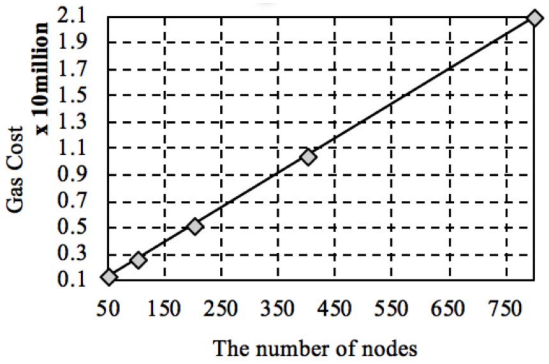


Fig. 7. Gas cost of packing up estimated time in blockchain.

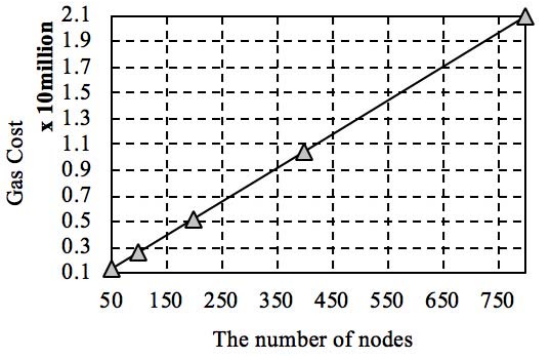


Fig. 8. Gas cost of packing up energy costs in blockchain.

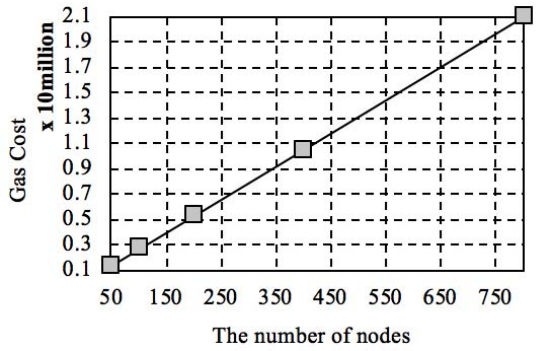


Fig. 9. Gas cost of packing up R values in blockchain.

observe that the growing trend was a near-linear increment. This observation implied that mining cost was in a stable manner, so that authorized miner servers could guarantee a balanced mining manipulation. The decentralization could be ensured from the perspective of gas cost.

Table I gave a comparison between our PBEM-SGN model and other access control model. We considered four types of attacks which are reply attack, collusion attack, impersonation attack, and eavesdropping attack to consider the comparison of security.

According to this paper, we found that our approach could resist all kinds of attacks shown in Table I while other approacher only could resist some kinds of attacks. In the table, a "$\surd$" means defendable and a "$\times$" means nondefendable. We found that the reason for resisting all kinds of attacks

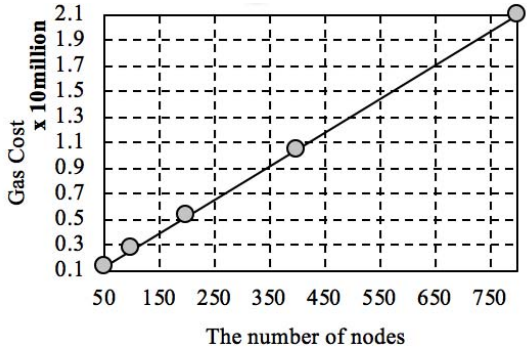


Fig. 10. Gas cost of packing up γ values in blockchain.

TABLE I
COMPARISONS BETWEEN PBEM-SGN AND OTHER ACCESS CONTROL MODELS. (RA: REPLAY ATTACK; CA: COLLUSION ATTACK; IA: IMPERSONATION ATTACK; EA: EAVESDROPPING ATTACK)

Method	RA	CA	IA	EA
PBEM-SGN	$\surd$	$\surd$	$\surd$	$\surd$
Lin [23]	$\times$	$\times$	$\surd$	$\times$
Seol [24]	$\times$	$\times$	$\times$	$\times$
Liu [25]	$\times$	$\times$	$\times$	$\times$
Chatterjee [26]	$\surd$	$\times$	$\surd$	$\surd$
Qiu [27]	$\times$	$\surd$	$\times$	$\times$

for our model is that our model had used CCA. So, authorized users' public IDs could not be achieved easily to implement reply attack, impersonation attack and eavesdropping attack by attackers. Because those attackers could not be verified by SN. And we used group signatures to defense collusion attack for a group user not knowing other group users except group manager.

In summary, main findings of our evaluations included: 1) the number of EN in the system had a positive relationship with the block creation time lengths; 2) energy transaction strategy could be created in an efficient manner; and 3) gas costs were similar when various packing tasks were implemented and the growth of gas cost was associated with the number of the EN.

C. Security Analysis

In this section, we would analyze security of our PBEM-SGN from security, privacy and data auditability these three parts.

Security: We assume that attackers can obtain ENs' valid group signatures so that attackers can pretend to be valid ENs. When an invalid EN registers at PBEM-SGN model, it needs to be validated as a valid EN by implementing the ENIV algorithm. However, before validating ENs registrations, they send their public IDs (a kind of physical signals) to the SN via a secure channel. Even if an invalid EN's σ_i obtains validation information (e.g., private key), its public ID E_i does not match E_i' which is stored in SN before ENs' registration, because we use a CCA to transfer public IDs.

Privacy: As we use permissioned blockchain in our model, an EN is marked according to address after registration rather than its real ID. And we use a group signature technique, such that ENs in the group do not know each other except

the SN. They only can verify whether ENs' signatures are valid. Besides, blockchain only records the estimated time (T), energy consumption (E), two adjustment parameters (γ and R), and energy transactions. Blockchain does not record anything about ENs' identities and group signatures; therefore, the model ensures the privacy from the perspective of the signature.

Data Auditability: Since we use group signature and blockchain techniques, we guarantee data auditability for our model. There are two basic reasons for this expectation. First, if a valid EN does tasks illegally, such as allocating energy illicitly to users in smart grid, an SN will implement the ENIV algorithm to trace the EN's real identity to expose it. Moreover, we use blocks to record energy transactions and parameters estimated time (T), energy consumption (E), γ and R; hence, we can consult task assignments data from blockchain to verify whether energy allocation is right or not.

VI. Conclusion

This paper focused on privacy-preserving problems in SGN and proposed an approach that could increase ability of data protections as well as ensure the performance of smart grid. The proposed approach used a permissioned blockchain system to bridge up all entities in SGN and introduced a special type of nodes (SNs) for validating participant voting nodes. Two authorizations were involved for node validations, including traditional access control schemes and the proposed CCA method. An RL-based method was designed to make sure the proposed approach could satisfy a high performance of SGN application without lowering down security quality. Our evaluations had provided a practical proof for the proposed approach.

References

[1] Y. Dai *et al.*, "Blockchain and deep reinforcement learning empowered intelligent 5G beyond," *IEEE Netw.*, to be published.

[2] N. Abbas, Y. Zhang, A. Taherkordi, and T. Skeie, "Mobile edge computing: A survey," *IEEE Internet Things J.*, vol. 5, no. 1, pp. 450–465, Feb. 2018.

[3] S. Wang *et al.*, "A survey on mobile edge networks: Convergence of computing, caching and communications," *IEEE Access*, vol. 5, pp. 6757–6779, 2017.

[4] S. Maharjan, Q. Zhu, Y. Zhang, S. Gjessing, and T. Basar, "Dependable demand response management in the smart grid: A Stackelberg game approach," *IEEE Trans. Smart Grid*, vol. 4, no. 1, pp. 120–132, Mar. 2013.

[5] N. Nikmehr and S. N. Ravadanegh, "Optimal power dispatch of multi-microgrids at future smart distribution grids," *IEEE Trans. Smart Grid*, vol. 6, no. 4, pp. 1648–1657, Jul. 2015.

[6] Y. Zhang *et al.*, "Cognitive machine-to-machine communications: Visions and potentials for the smart grid," *IEEE Netw.*, vol. 26, no. 3, pp. 6–13, May/Jun. 2012.

[7] K. Wang *et al.*, "Wireless big data computing in smart grid," *IEEE Wireless Commun.*, vol. 24, no. 2, pp. 58–64, Apr. 2017.

[8] Y. Zhang *et al.*, "Home M2M networks: Architectures, standards, and QoS improvement," *IEEE Commun. Mag.*, vol. 49, no. 4, pp. 44–52, Apr. 2011.

[9] R. Lee, M. Assante, and T. Conway. (2016). *Analysis of the Cyber Attack on the Ukrainian Power Grid*. [Online]. Available: https://www.nerc.com/pa/CI/ESISAC/Documents/E-ISAC_SANS_Ukraine_DUC_18Mar2016.pdf

[10] H. Kim, R. B. Chitti, and J. Song, "Novel defense mechanism against data flooding attacks in wireless ad hoc networks," *IEEE Trans. Consum. Electron.*, vol. 56, no. 2, pp. 579–582, May 2010.

[11] G. Gogniat *et al.*, "Reconfigurable hardware for high-security/high-performance embedded systems: The SAFES perspective," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 16, no. 2, pp. 144–155, Feb. 2008.

[12] Y. Cai, Y. Cao, Y. Li, T. Huang, and B. Zhou, "Cascading failure analysis considering interaction between power grids and communication networks," *IEEE Trans. Smart Grid*, vol. 7, no. 1, pp. 530–538, Jan. 2016.

[13] K. Gai, M. Qiu, Z. Ming, H. Zhao, and L. Qiu, "Spoofing-jamming attack strategy using optimal power distributions in wireless smart grid networks," *IEEE Trans. Smart Grid*, vol. 8, no. 5, pp. 2431–2439, Sep. 2017.

[14] G. Liang, J. Zhao, F. Luo, S. Weller, and Z. Y. Dong, "A review of false data injection attacks against modern power systems," *IEEE Trans. Smart Grid*, vol. 8, no. 4, pp. 1630–1638, Jul. 2017.

[15] Z. Zhang *et al.*, "When privacy meets economics: Enabling differentially-private battery-supported meter reporting in smart grid," in *Proc. IEEE/ACM 25th Int. Symp. Qual. Service*, 2017, pp. 1–9.

[16] Z. Zhang, Z. Qin, L. Zhu, J. Weng, and K. Ren, "Cost-friendly differential privacy for smart meters: Exploiting the dual roles of the noise," *IEEE Trans. Smart Grid*, vol. 8, no. 2, pp. 619–626, Mar. 2017.

[17] K. Gai, K.-K. R. Choo, M. Qiu, and L. Zhu, "Privacy-preserving content-oriented wireless communication in Internet-of-Things," *IEEE Internet Things J.*, vol. 5, no. 4, pp. 3059–3067, Aug. 2018.

[18] P. McDaniel and S. McLaughlin, "Security and privacy challenges in the smart grid," *IEEE Security Privacy*, vol. 7, no. 3, pp. 75–77, May/Jun. 2009.

[19] K. Gai and M. Qiu, "Blend arithmetic operations on tensor-based fully homomorphic encryption over real numbers," *IEEE Trans. Ind. Informat.*, vol. 14, no. 8, pp. 3590–3598, Aug. 2018.

[20] M. A. Khan and K. Salah, "IoT security: Review, blockchain solutions, and open challenges," *Future Gener. Comput. Syst.*, vol. 82, pp. 395–411, May 2018.

[21] Y. Jiao, P. Wang, D. Niyato, and Z. Xiong, "Social welfare maximization auction in edge computing resource allocation for mobile blockchain," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2018, pp. 1–6.

[22] K. Gai, Y. Wu, L. Zhu, M. Qiu, and M. Shen, "Privacy-preserving energy trading using consortium blockchain in smart grid," *IEEE Trans. Ind. Informat.*, to be published.

[23] L. Lin, T. Liu, S. Li, C. M. S. Magurawalage, and S. Tu, "Priguarder: A privacy-aware access control approach based on attribute fuzzy grouping in cloud environments," *IEEE Access*, vol. 6, pp. 1882–1893, 2018.

[24] K. Seol, Y.-G. Kim, E. Lee, Y.-D. Seo, and D.-K. Baik, "Privacy-preserving attribute-based access control model for XML-based electronic health record system," *IEEE Access*, vol. 6, pp. 9114–9128, 2018.

[25] Q. Liu, H. Zhang, J. Wan, and X. Chen, "An access control model for resource sharing based on the role-based access control intended for multi-domain manufacturing Internet of Things," *IEEE Access*, vol. 5, pp. 7001–7011, 2017.

[26] S. Chatterjee *et al.*, "On the design of fine grained access control with user authentication scheme for telecare medicine information systems," *IEEE Access*, vol. 5, pp. 7012–7030, 2017.

[27] M. Qiu, K. Gai, B. Thuraisingham, L. Tao, and H. Zhao, "Proactive user-centric secure data scheme using attribute-based semantic access controls for mobile clouds in financial industry," *Future Gener. Comput. Syst.*, vol. 80, pp. 421–429, Mar. 2018.

[28] K. Zhang, Y. Mao, S. Leng, Y. He, and Y. Zhang, "Mobile-edge computing for vehicular networks: A promising network paradigm with predictive off-loading," *IEEE Veh. Technol. Mag.*, vol. 12, no. 2, pp. 36–44, Jun. 2017.

[29] K. Wang, H. Yin, W. Quan, and G. Min, "Enabling collaborative edge computing for software defined vehicular networks," *IEEE Netw.*, vol. 32, no. 5, pp. 112–117, Sep./Oct. 2018.

[30] M. Liu *et al.*, "Computation offloading and content caching in wireless blockchain networks with mobile edge computing," *IEEE Trans. Veh. Technol.*, vol. 67, no. 11, pp. 11008–11021, Nov. 2018.

[31] R. Yu *et al.*, "Optimal resource sharing in 5G-enabled vehicular networks: A matrix game approach," *IEEE Trans. Veh. Technol.*, vol. 65, no. 10, pp. 7844–7856, Oct. 2016.

[32] K. Zhang, S. Leng, Y. He, S. Maharjan, and Y. Zhang, "Mobile edge computing and networking for green and low-latency Internet of Things," *IEEE Commun. Mag.*, vol. 56, no. 5, pp. 39–45, May 2018.

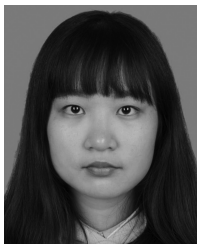
[33] I. Psaras, "Decentralised edge-computing and IoT through distributed trust," in *Proc. 16th Annu. Int. Conf. Mobile Syst. Appl. Services*, 2018, pp. 505–507.

[34] Z. Li *et al.*, "Consortium blockchain for secure energy trading in industrial Internet of Things," *IEEE Trans. Ind. Informat.*, vol. 14, no. 8, pp. 3690–3700, Aug. 2018.
[35] J. Kang *et al.*, "Enabling localized peer-to-peer electricity trading among plug-in hybrid electric vehicles using consortium blockchains," *IEEE Trans. Ind. Informat.*, vol. 13, no. 6, pp. 3154–3164, Dec. 2017.
[36] V. Sharma, I. You, F. Palmieri, D. N. K. Jayakody, and J. Li, "Secure and energy-efficient handover in fog networks using blockchain-based DMM," *IEEE Commun. Mag.*, vol. 56, no. 5, pp. 22–31, May 2018.
[37] J. Kang *et al.*, "Blockchain for secure and efficient data sharing in vehicular edge computing and networks," *IEEE Internet Things J.*, to be published.
[38] Z. Xiong, Y. Zhang, D. Niyato, P. Wang, and Z. Han, "When mobile blockchain meets edge computing," *IEEE Commun. Mag.*, vol. 56, no. 8, pp. 33–39, Aug. 2018.
[39] H. Liu, Y. Zhang, and T. Yang, "Blockchain-enabled security in electric vehicles cloud and edge computing," *IEEE Netw.*, vol. 32, no. 3, pp. 78–83, May/Jun. 2018.
[40] Y. Brun and N. Medvidovic, "Entrusting private computation and data to untrusted networks," *IEEE Trans. Depend. Secure Comput.*, vol. 10, no. 4, pp. 225–238, Jul./Aug. 2013.
[41] Y. Zhang and N. Ansari, "HERO: Hierarchical energy optimization for data center networks," *IEEE Syst. J.*, vol. 9, no. 2, pp. 406–415, Jun. 2015.
[42] J. Barr and R. Majumder, "Integration of distributed generation in the Volt/VAR management system for active distribution networks," *IEEE Trans. Smart Grid*, vol. 6, no. 2, pp. 576–586, Mar. 2015.
[43] Z. Su, Q. Xu, H. Zhu, and Y. Wang, "A novel design for content delivery over software defined mobile social networks," *IEEE Netw.*, vol. 29, no. 4, pp. 62–67, Jul./Aug. 2015.
[44] C. Y. T. Ma, D. K. Y. Yau, N. K. Yip, and N. S. V. Rao, "Privacy vulnerability of published anonymous mobility traces," *IEEE/ACM Trans. Netw.*, vol. 21, no. 3, pp. 720–733, Jun. 2013.
[45] R. Lu, X. Li, T. H. Luan, X. Liang, and X. Shen, "Pseudonym changing at social spots: An effective strategy for location privacy in VANETs," *IEEE Trans. Veh. Technol.*, vol. 61, no. 1, pp. 86–96, Jan. 2012.
[46] C. Turkay, A. Slingsby, H. Hauser, J. Wood, and J. Dykes, "Attribute signatures: Dynamic visual summaries for analyzing multivariate geographical data," *IEEE Trans. Vis. Comput. Graphics*, vol. 20, no. 12, pp. 2033–2042, Dec. 2014.
[47] L. Duan, L. Peng, and B. Li, "Predicting architectural vulnerability on multithreaded processors under resource contention and sharing," *IEEE Trans. Depend. Secure Comput.*, vol. 10, no. 2, pp. 114–127, Mar./Apr. 2013.
[48] D. He, C. Chen, S. Chan, and J. Bu, "Secure and efficient handover authentication based on bilinear pairing functions," *IEEE Trans. Wireless Commun.*, vol. 11, no. 1, pp. 48–53, Jan. 2012.
[49] C.-K. Chu, S. S. M. Chow, W.-G. Tzeng, J. Zhou, and R. Deng, "Key-aggregate cryptosystem for scalable data sharing in cloud storage," *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 2, pp. 468–477, Feb. 2014.
[50] G. Shah, A. Molina, and M. Blaze, "Keyboards and covert channels," in *Proc. USENIX Security Symp.*, vol. 15, 2006, pp. 59–75.
[51] D. Boneh, X. Boyen, and H. Shacham, "Short group signatures," in *Advances in Cryptology—CRYPTO*, M. Franklin, Ed., Santa Barbara, CA, USA: Springer, 2004, pp. 41–55.

Keke Gai (S'13–M'17) received the B.Eng. degree in automation from the Nanjing University of Science and Technology, Nanjing, China, in 2004, the M.E.T. degree in educational technology from the University of British Columbia, Vancouver, BC, Canada, in 2010, the M.B.A. degree in business administration and the M.S. degree in information technology from Lawrence Technological University, Southfield, MI, USA, in 2009 and 2014, respectively, and the Ph.D. degree in computer science from Pace University, New York, NY, USA.

He is currently an Associate Professor with the School of Computer Science and Technology, Beijing Institute of Technology, Beijing, China. He has authored or co-authored 2 books and over 100 high-quality journal/conference papers. His current research interests include cyber security, edge computing, cloud computing, blockchain, and reinforcement learning.

Dr. Gai was a recipient of five IEEE Best Paper Awards at conferences such as TrustCom'18 and HPCC'18 and two IEEE Best Student Paper Awards over the last five years.



Yulu Wu is currently pursuing the master's degree in computer science at the School of Computer Science and Technology, Beijing Institute of Technology, Beijing, China.

Her current research interests include cybersecurity, blockchain, and cloud computing.

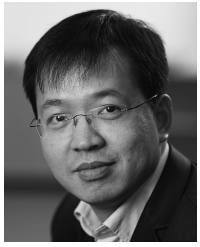
Liehuang Zhu (M'11) received the B.E. and M.E. degrees from Wuhan University, Wuhan, China, in 1998 and 2001, respectively, and the Ph.D. degree in computer science from the Beijing Institute of Technology, Beijing, China, in 2004.

He is currently a Professor with the School of Computer Science and Technology, Beijing Institute of Technology. He has authored or co-authored over 100 peer-reviewed journal or conference papers, including over 10 IEEE/ACM TRANSACTIONS papers, such as the IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, the IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, the IEEE TRANSACTIONS ON SMART GRID, *Information Sciences*, *IEEE Network*, and *Computer and Security*. His current research interests include security protocol analysis and design, wireless sensor networks, and cloud computing.

Dr. Zhu was a recipient of a number of IEEE Best Paper Awards at conferences such as IWQoS'17 and TrustCom'18.

Lei Xu received the Ph.D. degree in electronic engineering from Tsinghua University (THU), Beijing, China, in 2015.

She was as a Post-Doctoral Fellow with the Department of Computer Science and Technology, THU, from 2015 to 2017. She is currently an Associate Professor with the School of Computer Science and Technology, Beijing Institute of Technology, Beijing. Her current research interests include privacy issues in social networks, data mining, and application of game theory.



Yan Zhang (M'05–SM'10) received the Ph.D. degree from the School of Electrical and Electronics Engineering, Nanyang Technological University, Singapore.

He is a Full Professor with the Department of Informatics, University of Oslo, Oslo, Norway. His current research interests include next-generation wireless networks leading to 5G, green and secure cyber-physical systems (e.g., smart grid, healthcare, and transport).

Dr. Zhang was a recipient of the 2018 Highly Cited Researcher Award (top 1% by citations) according to Clarivate Analytics. He is an Associate Technical Editor of the *IEEE Communications Magazine*, an Editor of the IEEE TRANSACTIONS ON GREEN COMMUNICATIONS AND NETWORKING, IEEE COMMUNICATIONS SURVEYS AND TUTORIALS, and the IEEE INTERNET OF THINGS JOURNAL, and an Associate Editor of IEEE ACCESS. He serves as the Chair for a number of conferences, including IEEE GLOBECOM 2017, IEEE VTC–Spring 2017, IEEE PIMRC 2016, IEEE CloudCom 2016, IEEE ICCC 2016, IEEE CCNC 2016, IEEE SmartGridComm 2015, and IEEE CloudCom 2015. He serves as a TPC member for numerous international conferences, including IEEE INFOCOM, IEEE ICC, IEEE GLOBECOM, and IEEE WCNC. He is an IEEE Vehicular Technology Society (VTS) Distinguished Lecturer. He is also a Senior Member of IEEE ComSoc, IEEE CS, IEEE PES, and IEEE VTS. He is a Fellow of IET.