A Blockchain-Based Computing Architecture for Mobile Ad Hoc Cloud

Zhenzhen Jiao, Baoxian Zhang, Li Zhang, Min Liu, Wei Gong, Cheng Li

Abstract

Mobile ad-hoc cloud can exploit the computing resources (e.g., smartphones, vehicles, and unmanned systems) scattered in the mobile environment to form a self-organized ad-hoc local resource pool for providing opportunistic computing services. However, the highly dynamic and distributed characteristics of the mobile ad-hoc network environment bring great challenges in privacy and security in such opportunistic resource sharing. In this article, we first discuss the attractive features of blockchain for providing such resource sharing services in the mobile ad-hoc network environment in a secure and trustful way and then discuss the problems caused when using existing consensus protocols in such an environment. We accordingly devise a blockchain based trustful mobile ad-hoc cloud architecture, AdChain cloud. We describe the functions at different layers in this architecture, including the network layer, blockchain layer, and smart contract layer. To adapt to the high dynamics of the mobile ad-hoc network environment, we design a stability-aware consensus protocol at the blockchain layer. Simulation results show that our solution can achieve improved performance as compared with existing work.

Introduction

The Internet computing paradigm is shifting from centralized Cloud Computing toward Mobile Edge Computing (MEC). In MEC, computing is typically performed at the network edge, for example, access points, base stations, and so on. It is widely recognized that MEC can harvest the vast amount of idle computation and storage resources distributed at the network edge to provide satisfactory services to mobile users. However, existing MEC architectures still rely on the deployment of certain infrastructures at the network edge and they are not suitable for many scenarios where such infrastructures are absent, for example, military and space applications.

In this article, we pay attention to an emerging research direction: mobile ad-hoc cloud [1, 2]. Mobile ad-hoc cloud aims to exploit underutilized local computing resources of mobile devices (e.g., smartphones, vehicles, unmanned systems) to form an ad-hoc organized local resource pool (i.e., mobile ad-hoc cloud). Such an ad-hoc cloud can work alone without the assistance of any infrastructure but still provide good cloud services to the temporarily self-organized mobile devices, which is a remarkable and even indispensable merit for many applications. For example, users in a rural area without any network infrastructure can enjoy computing services by using such cloud architecture. Accordingly, a local resource pool can be formed by using a mobile ad-hoc cloud while resources at different mobile devices will be gathered, aggregated, and leveraged to provide computing services. Another example is self-organized vehicles moving in an area, also known as Vehicular Ad-hoc Networks (VANET), which can be temporarily organized to perform collective task processing without the assistance of fixed infrastructure.

Although mobile ad-hoc cloud provides a promising vision for exploiting scattered computing resources in the mobile environment, its service capability depends largely on the quality of the communication services provided by the underlying mobile ad hoc networks (MANETs), which in general have the following characteristics: infrastructure-less and no central control, high network dynamics due to unpredictable node mobility and even frequent network partitions, and limited resources at nodes. These characteristics bring great challenges in privacy and security for resource sharing among users, especially strangers. Such drawbacks hinder the wide usage of mobile ad-hoc cloud in practice. How to maximally attract users to involve for opportunistic resource sharing in the mobile ad hoc network environment while maintaining their privacy and security and also making the service be away from attacks such as selfish attack, Distributed Denial of Service (DDoS), and and so on, is still an open issue and needs in-depth study.

Recently, blockchain as a novel decentralized paradigm for providing trustful and secure resource sharing services has received great attention from both industry and academia. Blockchain is the foundation technology of the famous cryptocurrency Bitcoin [3]. Blockchain and its key features, such as decentralization, persistency, anonymity, and auditability, provide a new direction to conquer existing problems in mobile ad-hoc cloud, especially when it works with smart contract, that is, a self-executing unalterable digital contract deployed on blockchain. However, existing blockchain prototypes and systems are only proven to be effective on the Internet infrastructure; some of their key designs (e.g., consensus protocol) may encounter severe

Zhenzhen Jiao is with the Institute of Digital Economy Industry of ICT-CAS, and also with Chaincomp Technologies; Baoxian Zhang (corresponding author), Li Zhang, and Wei Gong are with the University of Chinese Academy of Sciences; Min Liu is with the Chinese Academy of Sciences; Cheng Li is with Memorial University, St. John's.

Digital Object Identifier:
10.1109/MNET.001.1800064

0890-8044/20/$25.00 © 2020 IEEE

problems when being used in a highly dynamic mobile ad-hoc network environment. This brings big challenges to deploy blockchain into mobile ad-hoc cloud and this issue has not been tackled in previous work.

In this article, we focus on studying scenarios where no fixed infrastructure exists such that mobile nodes need to build a blockchain system themselves for providing secure and trustful resource sharing services. For this purpose, we shall tackle the following issues:

- Why blockchain is attractive for mobile ad-hoc cloud.
- Why existing blockchain solutions cannot work effectively in highly dynamic mobile ad-hoc network environment.
- How to improve a blockchain to enable its practical usage in mobile ad-hoc cloud.

In this article, we first introduce blockchain and smart contract, and discuss their merits and problems in the mobile ad-hoc environment. We then introduce existing mobile ad-hoc cloud architectures and also existing blockchain systems and discuss in detail the problems of their used consensus protocols in the mobile ad-hoc environment. To address the above problems, we devise a new mobile ad-hoc cloud architecture, AdChain cloud. In this architecture, to tackle the high dynamic nature of mobile ad-hoc network environments, we propose a stability-aware consensus protocol at the blockchain layer. Simulation results show that our solution can achieve improved performance as compared with existing work. To the best of our knowledge, this is the first work exploring how to deploy a blockchain based solution in mobile ad-hoc network environments.

The remainder of this article is organized as follows.The following section introduces blockchain and smart contract and discusses their advantages and disadvantages in mobile ad-hoc network environments. We then introduce existing mobile ad-hoc cloud architectures and existing blockchain systems. We discuss in detail the problems of the consensus protocols in mobile ad-hoc network environments. Following that we propose a new architecture, AdChain cloud, including system model, a new stability-aware consensus protocol, and smart contract related functionalities. We then conduct simulations for performance evaluation by comparing our solution with existing work. We conclude this article in the final section.

Blockchain and Smart Contract

In this section, we first briefly introduce blockchain and smart contract, respectively, then discuss security and privacy problems in mobile ad-hoc cloud systems, and finally discuss the suitability of using blockchain and smart contract for enabling secure computing in mobile ad-hoc cloud.

Blockchain and Smart Contract

Blockchain: Blockchain is the foundation technology of the first cryptocurrency Bitcoin [3] and it is expected to become the infrastructure of the next-generation Internet. In blockchain, transactions are packaged into blocks. Each block contains its previous block's hash value, which eventually results in a hash chain called "blockchain." Figure 1 gives a common structure of a blockchain. Peers in the system offer computing resources to compete for packaging transactions into the blockchain and the winner will be rewarded with coins from the system and also charged with transaction fees. Such a recording process is also referred to as mining. Since peers carry out block mining in a purely distributed manner (hereafter we shall call these peers miners), a consensus protocol becomes indispensable for determining which miner's block will be the next block appended onto the blockchain. In the Nakamoto consensus used in Bitcoin, the miner gets the packaging right by affording a valid proof that it has solved a specific crypto-puzzle. This consensus is also known as Proof of Work (PoW).

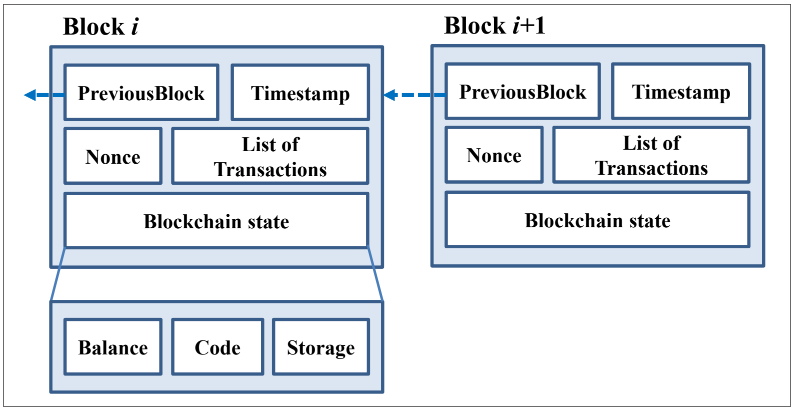

FIGURE 1. Illustration to design of blockchain.

Smart Contract: Blockchains that support smart contract are often referred to as Blockchain 2.0. The essence of smart contract is a self-executing digital contract in a secure environment with no intervention from the external environment. The main hindrance for realizing smart contract in past decades is that it is hard to find a secure decentralized environment. However, blockchain solves this problem. Currently, there are two famous smart contract platforms, that is, Ethereum [4] and Hyperledger [5], which have already launched thousands of decentralized applications (DAPPs) based on smart contract and they are running without frauds, downtime, or third party interference.

Security and Privacy Problems in Mobile Ad-Hoc Cloud

In mobile ad-hoc cloud, some fundamental functions must consider security and privacy issues, such as device discovery and registration, task partition and distribution among different devices, result transmissions, incentive to participation, and so on. Existing trust management solutions are not practically feasible in mobile ad-hoc cloud. This is because, in mobile ad-hoc network environments, node anonymity, high network dynamics, and even frequent network partitions make some basic requirements for maintaining a trustful system are hard to be met, for example, key management and entity authentication. For such a reason, system security in such cases is hard to maintain. Furthermore, there exist several potential security threats to mobile ad-hoc cloud from both interior users and external adversaries, as listed in the following.

> Blockchain and smart contract have remarkable features and they can help establish secure and trustful cloud architecture in a dynamic, distributed, and untrustworthy mobile ad-hoc network environment while encouraging mobile users' involvement and resource sharing.

Selfish and Cheating Behaviors: Devices with selfish and cheating behaviors aim to satisfy their own lust but hurt others' benefits and also fairness of the system. To facilitate the following discussions, we divide devices in the system into three types as in existing work, that is, task publisher, helper, and handler. A handler is often realized by a designated node that receives, partitions, and distributes tasks. Accordingly, potential security risks are as follows:

- A helper receives payment but works negatively, or leaves the network without providing working results, or provides fake results.
- A task publisher receives results from helpers but denies to pay or just leaves the network.
- A device intentionally publishes meaningless or bogus tasks but with intensive computation complexity to consume others' resources and thus wins rewarding tasks with no competitor, or even causes a DDoS attack.

Intentional Attacks: The inherent distributed and dynamic nature of a mobile ad-hoc cloud system makes it vulnerable to intentional attacks from adversaries. The potential risks include the following:

- An adversary may attack and paralyze a physical device who is playing the role of handler and thus lead to a DDoS attack.
- An adversary may lure task assignment to it by announcing availability of computational resources or a cheap price, and then provides false results back to the task publisher.
- An adversary may capture the key information of users during their dissemination in the multi-hop ad-hoc network.
- An adversary may pretend to be another user to steal rewards and information of others, which results in privacy leaks and monetary loss.

To address the above issues, a secure architecture AMCloud was proposed in [1]. The solution suggested by AMCloud is the introduction of a reward and reputation system. However, establishing an efficient reward and reputation system in an ad-hoc environment still faces many challenging issues, such as dealing with the falsifying to reputation values from malicious nodes [1].

Suitability of Blockchain and Smart Contract for Mobile Ad-Hoc Cloud

Blockchain and smart contract are suitable to address the above security and privacy concerns in mobile ad-hoc cloud due to their salient features such as decentralization, persistency, anonymity, and auditability. More details are as follows.

First, blockchain is secured by public key cryptography. Each node can generate its public and private keys and thus can generate its address with hashes to the public key, without the need to get authentication from a third entity or Certification Authority (CA) as required in Public Key Infrastructure (PKI). Therefore, blockchain eliminates the issue of key spoofing by a third party and thus largely decreases the possibilities of information hijacking and identity spoofing due to the existence of information encryption and digital signatures based on these security keys.

Second, by leveraging smart contract, task publishing, distribution, and commitment can be performed in a safeguarded manner. For example, a peer that wants to publish a task (represented by a smart contract in the system) must submit a deposit in advance to ensure the subsequent reward and also enough transaction fee to cover the system consumption. Thus, malicious publishing of tasks is inhibited due to the high cost of task publishing. Furthermore, a helper should also provide a deposit before it undertakes a task. The submitted results by helpers will be verified by randomly-chosen third parties without conflict-of-interests. Fake results will be picked out and punished. Apparently, this is useful to solve the selfish and cheating issues from task helpers since their negative work or fake results can lead to a punitive deduction of their deposits. Finally, once the results are verified, the related fees (i.e., reward or punishment) will be settled automatically by smart contract without any interference.

Finally, as a decentralized ledger, blockchain can let a mobile ad-hoc cloud system be completely away from the issue of single-point of failures in a dynamic network environment since each node maintains the statuses and records of the whole system. Furthermore, such statuses and records are tamper-resistant and immutable to any entities in the system once the consensus completes.

Therefore, blockchain and smart contract have remarkable features and they can help establish secure and trustful cloud architecture in a dynamic, distributed, and untrustworthy mobile ad-hoc network environment while encouraging mobile users' involvement and resource sharing.

Existing Mobile Ad-Hoc Cloud Architectures and Blockchain Systems

In this section, we first briefly introduce existing mobile ad-hoc cloud architectures and then introduce existing blockchain systems and applications. We then introduce existing consensus protocols used in blockchain systems and further discuss their problems in dynamic mobile ad-hoc network environments.

Existing Mobile Ad-Hoc Cloud Architectures

In [6], Fernando *et al*. raised three most challenging issues (including security, incentives, and connectivity) in mobile ad-hoc cloud architectures and further defined five functional roles (i.e., job handler, resource handler, cost manager, privacy and security manager, and context manager) for such an environment. In [2], Hou *et al*. proposed a Vehicular Fog Computing (VFC) framework, which utilizes closely-located vehicle resources and further enables them to collaborate with each other to form a resource pool. They quantitatively characterized the communication and computational capacities that can be brought by VFC. AMCloud [1] is the first mobile ad-hoc cloud architecture that takes security and privacy concerns into account and it aims to build secure, automatic, and efficient ad-hoc clouds for static-and-mobile environments. They further discussed security and privacy risks in providing such ad-hoc cloud services. However, how to technically realize a mobile ad hoc cloud was not provided in all the above work. In contrast, in this article, we shall devise a blockchain based secure mobile ad-hoc cloud architecture to push a step closer toward providing such services.

Existing Blockchain Systems and Applications

Bitcoin [3], Ethereum [4], and Hyperledger [5] are the three most influential implementations of blockchain. Among them, Bitcoin mainly acts as a decentralized cryptocurrency, and the other two act as smart contract's run-time infrastructures. There also exists some work for improving the performance of existing systems such as Bitcoin-NG [7], a next-generation Bitcoin. In Bitcoin-NG, the consensus was decoupled into two planes: leader election and transaction serialization. Leader is randomly elected infrequently. Each newly elected leader is responsible for packaging transactions independently. Bitcoin-NG has improved throughput performance compared with Bitcoin.

Another application of blockchain is to manage Internet-of-Things (IoT) systems such as access control, smart home, smart city, intelligent transportation systems, robotic swarm, and trust computing environments. Please refer to [8-9] for more details in this aspect. However, existing work in this aspect did not discuss how to deploy the blockchain based solutions into mobile ad-hoc network environments.

In [10], a blockchain-based decentralized task cooperation framework for crowdsourcing, called CrowdBC, was proposed. CrowdBC adopts incentive based collaborative task processing like our work in this article. However, in [10], CrowdBC was implemented based on Ethereum, which may lead to low efficiency in an ad-hoc network environment due to the use of Proof of Work based consensus. In [11], the authors proposed a blockchain based carpooling system where the Road Side Units (RSUs) form a private blockchain to undertake and record carpooling processes to guarantee data auditability. However, the static RSUs based blockchain in [11] is quite different from the mobile ad hoc network environment in this article where no infrastructure is available.

FlopCoin [12] is a cryptocurrency that is used to compensate helpers who are willing to execute offloadable tasks in a device-to-device (D2D) network. In [12], FlopCoin payments take the form of transactions and are stored in a blockchain, and the blockchain is assumed to be deployed in the cloudlets at the mobile network edge. The authors focused mostly on the design of an incentive mechanism, reputation mechanism, and pricing mechanism for the computation offloading in such a network environment and further descrbied how to use FlopCoin in the implementation of these mechanisms. In comparison, in this article, we focus on stand-alone mobile ad hoc networks and study how to design an executable blockchain based mobile ad-hoc cloud architecture managed completely by the mobile nodes for such a network environment and further describe in detail the functions to be supported at different layers in this architecture.

In highly dynamic mobile ad-hoc networks or even intermittently connected networks, it is difficult to guarantee the authenticity of the proof of stake at different nodes since network partitions may happen frequently and the communications between nodes might also be unreliable. Furthermore, the unfairness issue may be aggravated in dynamic mobile ad hoc networks.

Existing Consensus Protocols and Their Problems in Mobile Ad-Hoc Cloud

In a blockchain, reaching consensus among untrustworthy nodes is a challenging issue in distributed environments and also has a big impact on the performance of blockchains. In the following, we will introduce typical consensus protocols and further discuss their problems in ad-hoc network environments.

PoW (Proof of Work) [3]: PoW is the consensus protocol used by Bitcoin and Ethereum. In PoW, each miner in the network needs to calculate the hash value of a potential block's header, which must be smaller than a certain given value. Once the calculation at a node reaches the target value, the node will broadcast the block to other nodes to verify its correctness. In general, nodes with higher computing powers are probably to reach their target values earlier. All nodes in the network can reach consensus on whether appending the new block (if validated) onto their locally kept blockchains. PoW has good anti-attack performance. However, PoW and its variants are not suitable for mobile ad-hoc cloud environments due to the following reasons. First, PoW based consensus is often not affordable for resource-limited mobile devices due to its high expenditure of energy for crypto-puzzle solving. Second, in PoW, the difficulty for calculating a new block is determined based on the speculation on the total computing power of the whole network, which is estimated from previous blocks by each node locally. Such a way of determining mining difficulty is to ensure that successive blocks have a fixed interval. This is based on an assumption that the set of nodes and therefore the total computational power will not fluctuate dramatically with time. However, in a mobile ad-hoc network, nodes may leave the network due to running out of energy, user mobility, and so on, and network partitions may happen frequently. Therefore, traditional PoW based consensuses have the stability issue in mobile ad-hoc networks due to rapid change of network connectivity. Furthermore, such a stability issue still exists in their variants such as Bitcoin-NG [7], in which an elected leader may leave the network suddenly in the middle of its epoch or become isolated due to node mobility.

PoS (Proof of Stake) [13]: PoS works based on an assumption that users with more currencies or tokens would be less likely to attack the system. Thus, in PoS, block-creators are determined based on their account balance, which often causes fairness issues in the system. Accordingly, some enhancements (e.g., introduction of randomization) had been proposed. A popular variant DPoS (Delegated Proof of Stake) was proposed in [14]. PoS is democratic among all network nodes. In contrast, in DPoS, stakeholders will elect their delegates to generate and validate blocks, which

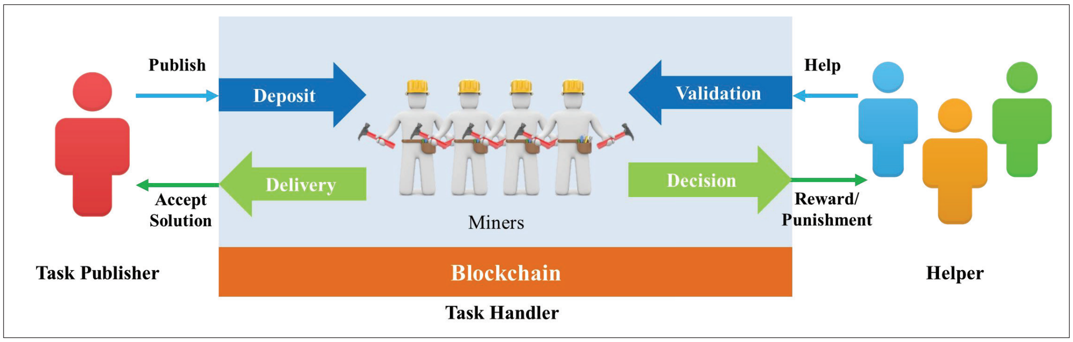

FIGURE 2. Illustration of the blockchain based resource sharing system model.

belong to representative democratic. Thus, fewer nodes are needed to validate blocks, which lead to quick confirmation of transactions. PoS based consensus is more energy-efficient than PoW based consensus. However, in highly dynamic mobile ad-hoc networks or even intermittently connected networks, it is difficult to guarantee the authenticity of the proof of stake at different nodes since network partitions may happen frequently and the communications between nodes might also be unreliable. Furthermore, the unfairness issue may be aggravated in dynamic mobile ad hoc networks. For example, some rich nodes may dominate the system such that only those transactions from such rich nodes are processed, which largely hurts the system fairness.

PBFT (Practical Byzantine Fault Tolerance) [5]: PBFT is utilized in Hyperledger Fabric [5], which is hosted by the Linux Foundation. In each round of a new block generation, a recorder is first selected and its recording process is divided into three phases: pre-prepared, prepared, and commit. Each transformation among these three phases needs permissions from over 2/3 of all nodes in the network. Some variations to PBFT were also proposed. For example, in Tendermint [15], nodes need to lock some cryptocurrency as pledge to become validators. One major hindrance for PBFT-based consensuses to be used in mobile ad-hoc networks is the requirement of more than 2/3 of all nodes to reach agreement, which may cause the scalability issue. More specifically, frequent network partitions in a mobile ad hoc network environment may make more than 1/3 nodes to form isolated network components, which makes PBFT no longer effective in such cases.

DAG (Directed Acyclic Graph) [9]: DAG is a new kind of ledger for storing transactions and it works based on the concept of directed acyclic graph. Tangle [9] is a typical DAG based blockchain. In tangle, no block is generated. When a new transaction joins the tangle network, it chooses two previous transactions for the approval, adding two new edges to the graph. The nodes do not have to achieve consensus on which valid transactions have the right to be in the ledger, that is, all transactions can stay in the tangle. However, nodes can decide which transactions will become orphaned (i.e., being approved by no one). Tangle is a basis of IOTA (a cryptocurrency for the Internet-of-Things (IoT) industry), which aims to become the next generation IoT blockchain. However, how to realize a safe and fair reward mechanism in DAG based systems is still an open issue. Furthermore, in a mobile ad-hoc network, the dynamic joining and leaving of nodes can even affect the approval status of transactions.

AdChain Cloud: A Blockchain Based Computing Architecture for Mobile Ad-Hoc Cloud

In this section, we devise a blockchain based mobile ad-hoc cloud architecture, AdChain cloud. We first model the system under study. We give an overview of the AdChain cloud architecture, then present a new consensus protocol to work at the blockchain layer in order to adapt to the high network dynamics in mobile ad hoc environments, and finally describe the functions at the smart contract layer.

System Model

The service framework for a mobile ad-hoc cloud is illustrated in Fig. 2. In this framework, system participants are divided into three categories, task publisher, helper, and task handler, which will be introduced in the following. Furthermore, we will also model the underlying mobile ad hoc network.

Task Publisher: A task publisher posts its task by creating a new smart contract with an amount of deposit. The deposit is composed by two parts: reward(s) to be paid to helper(s) and a transaction fee to the system. The latter is determined by the smart contract compiler.

Helper: Helpers can be any devices in the system, which can contribute their spare computational power to undertake published tasks and thus obtain corresponding rewards. Unlike other cloud based systems, in AdChain cloud, helpers do not need to report their available resources to others for maintaining a distributable resource pool. This protects participants' privacy in wireless environments.

Task Handler: Task handler is a logical concept rather than a third-party entity. In other words, the whole blockchain plays the role of task handling in the system rather than any individual node. The functions of task handling in a blockchain system typically include result verification, reward and punishment decision, and so on. In the implementations, these functions can be undertaken by block-creators in the system. Hereafter, we use the term *miner* to refer to a block-creator in the system. Besides, we assume that there exist some verification functions that can be utilized by miners to verify the results provided by helpers as assumed in [10].

Network Model: The mobile cloud is built on a mobile ad-hoc network (MANET), which can be modeled as an undirected graph $G = (V, E)$, where $V(G)$ and $E(G)$ represent the sets of nodes and the set of links connecting nodes in the network, respectively. We assume nodes in the network have a uniform communication range R. Each node is equipped with an omnidirectional antenna such that there exists a link between a pair of nodes u and v ($u, v \in V(G)$) if $d_{uv} \leq R$, where d_{uv} represents the geometrical distance between u and v. The existence of a link between a pair of nodes means they are neighbor nodes. Nodes in the network may have different mobility patterns. Some nodes may stay static most of the time or move infrequently. The existence of such nodes is helpful to increase the stability of the network and therefore improve the stability of blockchain in the network.

It should be noted that the underlying MANET in our architecture is quite different from an underlying P2P network used in many blockchain systems in the Internet. On the one hand, both of them face membership dynamics. Such dynamics in a P2P network lie mainly in peers' dynamic joining and leaving, which, however, are typically considered to happen at a low rate compared to the large number of peers in the P2P network. For a MANET, such dynamics also exist due to nodes' arriving and leaving, power off and on. On the other hand, neighbor relationship in mobile ad-hoc networks is highly dynamic due to the uncontrollable node mobility while the neighbor relationship in a P2P network is in general quite stable after a node joins the network, consider the fact that the paths connecting different peers in the Internet are generally quite stable. Thus, how to provide stable blockchain based services in dynamic mobile ad hoc network environments is a challenging issue.

Overview of AdChain Cloud Architecture

The AdChain cloud architecture consists of three layers: the underlying network layer, blockchain layer, and the overlay smart contract based application layer (also called smart contract layer) (see Fig. 3). In AdChain cloud, mobile devices form a self-organized wireless ad-hoc network where nodes communicate via wireless links. The block chain layer provides the run-time infrastructure for smart contracts and fulfills the following system design objectives, that is, decentralization, persistency, anonymity, and auditability. Users (mobile devices) post their tasks out for seeking help from other nodes in the network, without considering whether the helpers are strangers or not. Nodes can join the system and contribute their idle computing resources to help others. Cryptocurrency/token is leveraged for incentive purposes and trust maintenance. Involved operations including task publishing, help request, and reward and punishment, are performed using smart contract. Next, we describe the design at each layer in details.

Blockchain Layer

The main functions at the blockchain layer include broadcast communications of various messages including transactions/blocks, information encryption and decryption, consensus protocols, and so on. Our new design at this layer is to tackle the high network dynamics in mobile ad-hoc environments, and we accordingly devise a new Stability-Aware PoW consensus Protocol (SEAP). The design of such a consensus protocol is due mainly to the inefficiency of existing consensus protocols in mobile ad-hoc networks as we discussed in the preceding section.

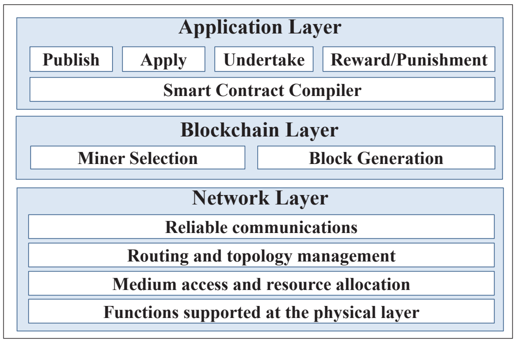

FIGURE 3. Structure of AdChain cloud architecture.

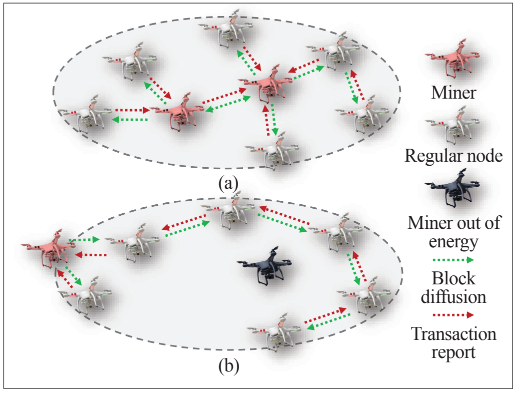

FIGURE 4. An example of a swarm of networked UAVs for illustrating the impact of network dynamics to blockchain.

To achieve high stability and also improved system performance in dynamic mobile ad-hoc environments, SEAP incorporates network-specific characteristics into the miner election process in PoW consensus. The design of SEAP comes from the following observation: network-specific characteristics of nodes (e.g., their node degrees and moving patterns) have a big impact on the system stability and also performance, which therefore largely affect the performance of the consensus protocol. Figure 4 gives an example for this purpose. Here, suppose DPoS is used as the consensus protocol. In Fig. 4a, two nodes are serving as delegated block generators. However, one of them suddenly moves to the boundary of the network and the other exhausts its energy and is down (Fig. 4b). As a result, nodes in the network need to re-establish routing paths for block and transaction dissemination. Furthermore, the existence of only one miner in DPoS may lead to a security leak issue. Also, this may hurt the robustness of the overlaying blockchain. Situations like that in Fig. 4b can eventually get self-healed after another round of delegated miner election but may lead to degraded mining performance.

SEAP decouples the mining process into leader election and transaction serialization like in Bitcoin-NG [7]. The key feature that SEAP differs

The leader election in SEAP needs to address the following two issues: how a network node determines its stability value, and how other nodes in the network can validate the truthfulness of such information in an untrusted environment. For this purpose, we need a metric characterizing the stability of a node in the network, which can be easily verified by other nodes in the network.

from the consensus protocol in Bitcoin-NG is the use of stability-aware leader election in the PoW consensus process. The purpose is to provide a stable mining process for generating blocks by determining a leader having high stability and low mobility in the network to provide stable block-packaging service at its epoch.

The leader election in SEAP needs to address the following two issues: how a network node determines its stability value, and how other nodes in the network can validate the truthfulness of such information in an untrusted environment. For this purpose, we need a metric characterizing the stability of a node in the network, which can be easily verified by other nodes in the network. That is, at each time when a node wants to elect for miner, it should provide not only its hash puzzle answer but also a proof for its stability in the network, both of which can be verified by other nodes (e.g., whether the candidate is a top stable node in the network).

In SEAP, the stability of a node x, denoted by $St(x)$, is calculated as follows.

$$St(x) = x_m - \alpha * x_{var}^{\beta}, \tag{1}$$

where x_m represents the mean degree of node x during the past n time slots, x_{var} denotes the standard deviation of degree of node x during the last n time slots, α and β are network parameters. Equation 1 prefers to pick out a node with low mobility and high degree to serve as a miner. That is, a node with more neighbor nodes is more likely to be reachable by other nodes in the network.

For the stability value claimed by a node to be verifiable by other nodes, SEAP adopts the following procedures. Each node needs to locally and periodically exchange its location information, extracted exactly at the beginning instant of a slot, with its direct neighbor nodes. Moreover, each node periodically broadcasts a message across the network, which contains its own information (including its ID and location at the beginning of the current slot) and all its neighbors (including their IDs and their locations at the beginning of the last slot). Other nodes receiving such a message can verify whether the claim in the broadcast message is true or not and locally record the information in the message (if confirmed true). Once nodes receive an election message from a leader-elector, they can also check that elector's stability. Furthermore, in SEAP, nodes in the network are required to wait for a certain long period for receiving more election messages to choose the most stable candidate. In this process, honest nodes can judge the authenticity of the information from other nodes and also report and block a fraudster's message. The multi-hop relaying manner and broadcast nature of the wireless medium in wireless ad-hoc networks can help nodes overhear others' messages and use the conflict between different messages to find out fraudster(s).

The following issues need to be addressed in the implementation of SEAP. The first issue is membership dynamics since the calculation of a node's stability using Eq. 1 involves the node's status in the past n time slots. Moreover, such statuses need to be verifiable by other nodes in the network. For this concern, we require only nodes having joined the network for a certain period of time can be used in the calculation in Eq. 1. The second issue is Sybil attacks such that some node may own multiple IDs, which can artificially increase its degree and thus lead to increased stability at such a node(s). In that case, such a node may own falsely high stability than the ground truth. One way to handle such a case is as follows. Since all such IDs keep sharing the same location from time to time, certain methods can be used to restrict the contribution of stability from such IDs in leader election. In addition, to avoid a node from claiming too many IDs, a considerable amount of computing/storage resources can be reserved for each claimed ID. The third issue is in case of fork occurrence. In this case, in SEAP, the longest chain is preferred, and in the case of equal chain length, the branch containing the highest number of transactions is chosen. The fourth issue is the setting of mining difficulty in SEAP since nodes in mobile ad-hoc networks typically have quite limited computing resources. For this concern, in SEAP, the mining difficulty is set in a way such that all nodes qualified to serve the leader role can get a PoW based puzzle answer in time with high probability. Moreover, when two or more nodes have equal top stability, the node which first provides a puzzle answer will be elected as leader. In this way, the concern for resource wastes in puzzle solving by PoW consensus are greatly alleviated. Moreover, such a stability-aware PoW consensus can also greatly alleviate the necessity of high computing power for puzzle solving at mobile devices and is therefore attractive for resource-limited MANETs.

Smart Contract Layer

In this subsection, we describe the functionalities at the smart contract layer for resource sharing in our architecture. Major functionalities at this layer include the following: publish task, apply task, undertake task, and reward and punishment.

Publish Task: Once a task publisher needs to post a task, it will create a new smart contract. A smart contract T published by a node n should explicitly provide the following information:

- The publishing time of T.
- The deadline for the task to be finished.
- The expected number of divided subtasks, which may not be exactly equal to the number of helpers since one helper may apply to undertake more than one subtask if they can finish them before the task deadline.
- The digital signature of n using their private key.
- The address of n.
- The upper bound of the affordable payment of the task publisher, which denotes the cryptocurrency that is paid by the publisher to cover the reward.
- Gas used to support running of the contract.
- Optionally, some thresholds for selecting helpers can also be provided.

Apply and Undertake Task: Once a task is published, users can apply to undertake one or multiple subtasks of it by accessing the corresponding contract. An applier must provide a deposit for possible punishment when it cannot provide a correct solution before the task deadline. Tasks will be undertaken by the applying helpers and processed using their computing resources.

Reward and Punishment: Reward will be fulfilled when a helper submitted a correct solution and punishment will be performed when no correct solution is received on time. A helper sends its solution, which is encrypted using its own private key, to validators. In our architecture, block generators play the role of validators, which can validate the solutions by using validation functions provided by the task publisher. When solutions are validated correct, validators will transmit these solutions to the task publisher by encrypting using publisher's public key for security and then send the reward to corresponding helpers. The reason for not allowing the publisher to be the validator is to avoid the case that the publisher receives a correct solution but denies paying by accusing a wrong solution.

Simulation Results

In this section, we evaluate the performance of AdChain cloud via simulations. We consider the following scenario: crowdsourcing in scientific exploration or a rural environment where no network infrastructure exists. Many mobile users holding smart devices move in such a scenario and they form an opportunistic mobile ad-hoc network environment. Some of the users may have certain tasks and need others' assistance. We assume all nodes have similar computing powers. We put the focus of our implementation of AdChain on examining the performance of our consensus protocol SEAP in a mobile ad hoc network environment. Accordingly, the simulations focus mainly on the blockchain layer and the network layer while those functions at the smart contract layer are greatly abstracted and are simply reflected in the transaction generation, dissemination, packaging, and confirmation. Thus, the simulations run at a transaction/block-level. The simulator was written in python.

The settings at the blockchain layer are as follows. We implement two different blockchains: Bitcoin-NG [9] and an implementation of Bitcoin-NG but replacing the consensus protocol therein with our SEAP protocol, referred to as Bitcoin-SEAP. In Bitcoin-NG, at each round of block generation, a node that first provides a PoW based puzzle answer will be elected as leader and will be responsible for the subsequent transaction packaging, which is equivalent to random election among all network nodes. In contrast, in a round of election in Bitcoin-SEAP, a certain election period is specified, during which the node providing correct hash puzzle answer and also having top stability will be elected as the leader. Upon receiving such election messages, nodes in the network will also verify the correctness of the provided answer and also figure out which elector has top stability. Regarding the stability calculation in Eq. 1, we set $\alpha = 0.5$ and $\beta = 0.25$.

The settings at the network layer are as follows. There are 50 nodes in an 1800 x 1800 m^2 area. Regarding node movement, preference based random way point mobility is adopted such that with 80 percent probability the movement destination is chosen in a central 1000 x 1000 m^2 area (representing popular area) while with 20 percent probability the moving destination is chosen in the remaining area. Each node stays at its current location for a period of time (called stationary time) and then moves to another location (the corresponding time is called moving time). Each node repeats this behavior, alternatively between moving and staying. The moving speed of nodes was set to 5 m/s. The mobility ratio of a node is the ratio between its total moving time and the total time (including both moving time and stationary time). We varied the mobility ratio of nodes to reflect different levels of network dynamics. The uniform transmission range of nodes in the network was set to 250 meters. With the above settings, we try to simulate a sparse mobile ad hoc network environment.

> We will further consider more realistic issues and their impact on the performance of AdChain cloud. Some of them are as follows. First, how the lossy nature of wireless links can affect the leader election and also transactions/block dissemination performance. Second, how different node distribution and mobility patterns affect the performance of our devised blockchain system.

At the MAC layer, we assume there are two separate channels in the network, one for exchanging signaling messages and the other for exchanging transactions and blocks. Moreover, we assume a TDMA based network such that there are 750 slots per second and one transaction fits exactly for one slot. For the selection of transmitters and receivers in each slot, the calculation of a maximal independent set using a greedy strategy is adopted such that each time the transmitter candidate leading to the maximal number of receiver candidates covered is always chosen. In this way, we are aimed to obtaining a view on the best performance of different solutions. Note that both transactions and blocks need to be flooded across the network. In the transmission, block transmission always has priority over transaction transmission. Moreover, the MAC layer is assumed to be ideal, which can guarantee packet delivery without loss.

Each simulation lasts for 900 seconds. Each network node generates transactions at a rate of one transaction per second. The leader generates blocks at a rate of one block per 10 seconds during its tenure. The size of a block depends on the actual number of packaged transactions with an upper limit of 500. A coarse explanation of the above settings is as follows. Suppose a transaction is equivalent to one packet of 128 bytes, then the channel rate of 750 slots/s will be 750 kb/s, and a block can have a maximum size of 64 kByte. Extra experiments were also carried out, which show that further increase of the channel rate will not lead to increased system performance based on our settings of transaction and block generation rate, and transaction size.

The performance metrics for comparison include Transactions Per Slot (TPS) and transaction packaging ratio. TPS is defined as the average number of transactions successfully packaged into the blockchain per slot. The transaction packaging ratio is defined as the ratio of the number of trans-

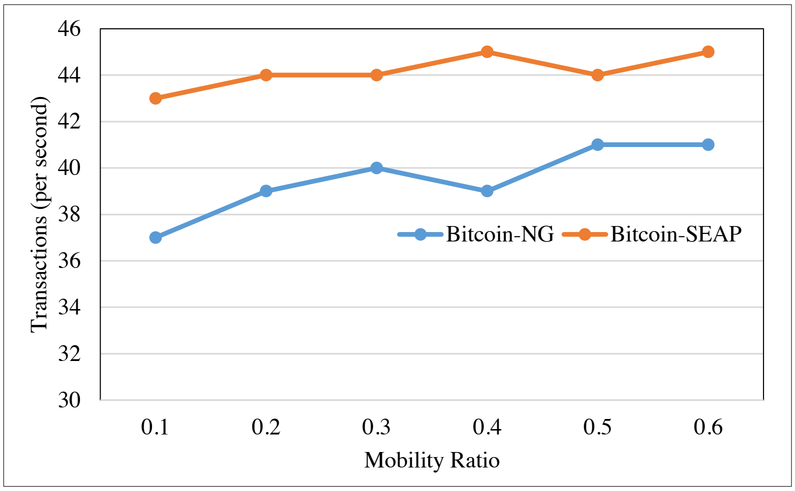

FIGURE 5. Comparison of TPS.

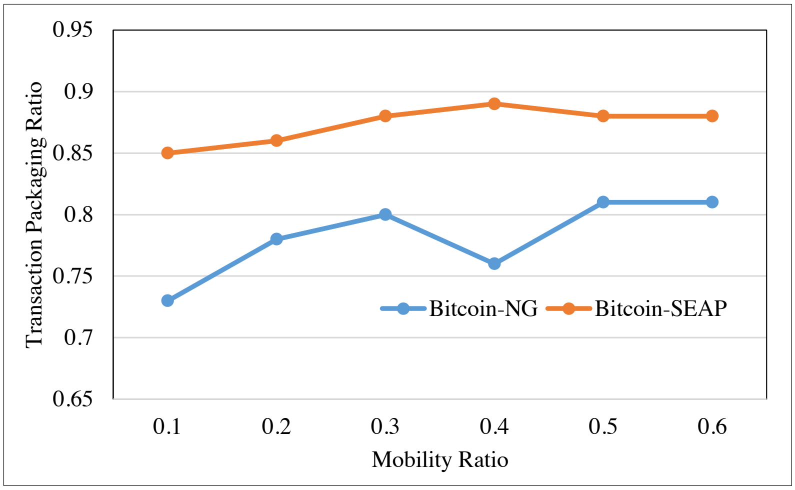

FIGURE 6. Comparison of transaction packaging ratio.

actions packaged in the blockchain to the total number of transactions generated. TPS reflects the throughput of a blockchain system. A higher packaging ratio means more generated transactions can be packaged and thus rewarded. Each value reported is the average result of multiple tests.

Figure 5 compares the TPS performance by different solutions under different mobility ratios. In Fig. 5, it is seen that Bitcoin-SEAP outperforms Bitcoin-NG significantly under different mobility ratios. Fig. 6 compares the transaction packaging ratio performance by different solutions under different mobility ratios. In Fig. 6, it is also seen that Bitcoin-SEAP outperforms Bitcoin-NG in terms of transaction packaging ratio under different mobility ratio settings, which means more transactions can be packaged. In Fig. 6, the significant fluctuation for the curve by Bitcoin-NG is due to its randomness selection of leader. Moreover, in Fig. 5 and Fig. 6, there in general show a trend that both TPS and transaction packaging ratio performance slightly increase with mobility ratio. This is because node mobility accelerates the block and transaction dissemination.

Conclusion

The inherent distributed and dynamic characteristics of mobile ad-hoc cloud bring severe privacy and security challenges in opportunistic resource sharing in such an environment. In this article, we studied how to use blockchain to conquer these problems. We devised a blockchain based trustful mobile ad-hoc cloud architecture, AdChain cloud. We further designed a stability-aware consensus protocol to handle the high dynamics in mobile ad-hoc network environments. Simulation results show that our solution can achieve improved performance as compared with existing work.

In the future, we will further consider more realistic issues and their impact on the performance of AdChain cloud. Some of them are as follows. First, how the lossy nature of wireless links can affect the leader election and also transactions/block dissemination performance. Second, how different node distribution and mobility patterns affect the performance of our devised blockchain system.

Acknowledgment

This work was supported in part by the National Natural Science Foundation of China under Grant no. 61872331, 61471339, and 61902273; by the Natural Sciences and Engineering Research Council (NSERC) of Canada (Discovery Grant RGPIN-2018-03792), and the InnovateNL SensorTECH Grant 5404-2061-101.

References

[1] D. Shila *et al.*, "AMCloud: Toward a Secure Autonomic Mobile Ad Hoc Cloud Computing System," *IEEE Wireless Commun.*, vol. 24, no. 2, Apr. 2017, pp. 74–81.
[2] X. Hou *et al.*, "Vehicular Fog Computing: A Viewpoint of Vehicles as the Infrastructures," *IEEE Trans. Veh. Technol.*, vol. 65, no. 6, June 2016, pp. 3860–73.
[3] S. Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System," 2008, see https://bitcoin.org/bitcoin.pdf.
[4] W. Gavin, "Ethereum: A Secure Decentralised Generalised Transaction Ledger," Ethereum Project Yellow Paper, 2016.
[5] Hyperledger Fabric, https://www.hyperledger.org/announcements/2017/07/11/hyperledger-announces-production-ready-hyperledger-fabric-1-0, 2017.
[6] N. Fernando, S. W. Loke, and W. Rahayu, "Mobile Cloud Computing: A Survey," *Future Generation Computer Systems*, vol. 29, no. 1, Jan. 2013, pp. 84–106.
[7] I. Eyal *et al.*, "Bitcoin-NG: A Scalable Blockchain Protocol," *Proc. USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, Boston, MA, Mar. 2016, pp. 45–59.
[8] H. Yin *et al.*, Hyperconnected Network: A Decentralized Trusted Computing and Networking Paradigm," *IEEE Network*, vol. 32, no. 1, Jan. 2018, pp. 112–17.
[9] S. Popov, "The Tangle," IOTA white paper, Oct. 2017.
[10] M. Li *et al.*, "CrowdBC: A Blockchain-Based Decentralized Framework for Crowdsourcing," IACR Cryptology ePrint Archive, 2017.
[11] M. Li, L. Zhu, and X. Lin, "Efficient and Privacy-Preserving Carpooling Using Blockchain-Assisted Vehicular Fog Computing," *IEEE Internet of Things J.*, early access, DOI: 10.1109/JIOT.2018.2868076.
[12] D. Chatzopoulos *et al.*, "FlopCoin: A Cryptocurrency for Computation Offloading," *IEEE Trans. Mobile Computing*, vol. 17, no. 5, Sept. 2018, pp. 1062–75.
[13] B. Vitalik and G. Virgil, "Casper the Friendly Finality Gadget," arXiv:1710.09437v2, Oct. 2017.
[14] F. Schuh and D. Larimer, "BitShares 2.0: Financial Smart Contract Platform," https://bitshares.org, 2016.
[15] J. Kwon, "Tendermint: Consensus Without Mining," http://tendermint.com/static/docs/tendermint.pdf, 2016.

Biography

Zhenzhen Jiao (jiaozhenzhen@ict.ac.cn) received his Ph.D. degree from the Research Center of Ubiquitous Sensor Networks, University of Chinese Academy of Sciences, Beijing, China, in 2015. He is currently an associate professor and also the director of the Blockchain Research Center of the Institute of Digital Economy Industry, Institute of Computing Technology, Chinese Academy of Sciences. From 2015 to 2016, he was a researcher at the China Academy of Information and Communications Technology, Beijing, China. His research interests include blockchain, wireless multi-hop network protocol and algorithm design.

Baoxian Zhang [SM'12] (bxzhang@ucas.ac.cn) received his B.S., M.S., and Ph.D. degrees in electrical engineering from Northern Jiaotong University (now Beijing Jiaotong University), China, in 1994, 1997, and 2000, respectively. He is currently a full professor with the Research Center of Ubiquitous Sensor Networks at the University of Chinese Academy of Sciences (UCAS), Beijing, China. Prior joining UCAS, he was a research scientist with the School of Information Technology and Engineering, University of Ottawa, Canada from 2002 to 2005. From 2001 to 2002, he was a postdoctoral fellow with the Department of Electrical and Computer Engineering, Queen's University, Kingston, Canada. He is currently an associate editor of *IEEE Systems Journal* and has served as a guest editor of several special issues including for *IEEE JSAC* and *Elsevier Ad Hoc Networks Journal*. He has published over 150 refereed technical papers in archival journals and conference proceedings. His research interests cover network protocol and algorithm design, wireless ad hoc and sensor networks, Internet of Things, and IP networks.

Li Zhang (zhangli183@mails.ucas.ac.cn) received her B.S. degree in 2015 from the School of Mathematics and Statistics of Hainan Normal University. She received the M.S. degree in 2018 from the School of Mathematical Sciences, Beijing Normal University. She is currently working toward her Ph.D. degree at the School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing, China. Her research interests include blockchain, consensus protocols, and so on.

Min Liu [M] (liumin@ict.ac.cn) received her B.S. and M.S. degrees in computer science from Xi'an Jiaotong University, China, in 1999 and 2002, respectively. She received her Ph.D. degree in computer science from the Graduate University of the Chinese Academy of Sciences in 2008. She is currently a professor at the Institute of Computing Technology, Chinese Academy of Sciences. Her current research interests include mobile computing, opportunistic routing, and transport control.

Wei Gong (gongwei11@mails.ucas.ac.cn) received the B.S. degree in computer science in 2011 from Zhejiang University, Hangzhou, China, and the Ph.D. degree in computer science from the University of Chinese Academy of Sciences in 2019. He has been a postdoctoral fellow at Western University, London, ON, Canada, since late 2019. He has served as a technical program committee member for IEEE ICC 2020. He has published over 10 papers in archival journals and conference proceedings. His research interests include mobile crowdsensing and mobile opportunistic networks.

Cheng Li [SM] (licheng@mun.ca) received the B.Eng. and M.Eng. degrees from Harbin Institute of Technology, Harbin, P. R. China, in 1992 and 1995, respectively, and the Ph.D. degree in electrical and computer engineering from Memorial University, St. John's, Canada, in 2004. He is currently a full professor with the Faculty of Engineering and Applied Science of Memorial University, St. John's, Canada. His research interests include mobile ad hoc and wireless sensor networks, wireless communications and mobile computing, switching and routing, and broadband communication networks. He is an editorial board member of *J. Networks*, *China Communications*, and an associate editor of *Wiley Security and Communication Networks*. He has served as a co-chair for various technical symposia of many international conferences, including IEEE GLOBECOM and IEEE ICC. He has served as a TPC member for many international conferences. He is a registered Professional Engineer (P.Eng.) in Canada and is a member of the IEEE Communication Society, Computer Society, Vehicular Technology Society, and Ocean Engineering Society.