IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received August 25, 2017, accepted September 24, 2017, date of publication September 29, 2017,
date of current version February 1, 2018.

Digital Object Identifier 10.1109/ACCESS.2017.2757955

A Software Defined Fog Node Based Distributed Blockchain Cloud Architecture for IoT

PRADIP KUMAR SHARMA[1], MU-YEN CHEN[2], AND JONG HYUK PARK[1], (Member, IEEE)
[1]Department of Computer Science and Engineering, Seoul National University of Science and Technology, Seoul 01811, South Korea
[2]Department of Information Management, National Taichung University of Science and Technology, Taichung 404, Taiwan

Corresponding author: Jong Hyuk Park (jhpark1@seoultech.ac.kr)

This research was supported by the MSIT (Ministry of Science and ICT), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2017-2014-0-00720-002) supervised by the IITP (Institute for Information and communications Technology Promotion).

ABSTRACT The recent expansion of the Internet of Things (IoT) and the consequent explosion in the volume of data produced by smart devices have led to the outsourcing of data to designated data centers. However, to manage these huge data stores, centralized data centers, such as cloud storage cannot afford auspicious way. There are many challenges that must be addressed in the traditional network architecture due to the rapid growth in the diversity and number of devices connected to the internet, which is not designed to provide high availability, real-time data delivery, scalability, security, resilience, and low latency. To address these issues, this paper proposes a novel blockchain-based distributed cloud architecture with a software defined networking (SDN) enable controller fog nodes at the edge of the network to meet the required design principles. The proposed model is a distributed cloud architecture based on blockchain technology, which provides low-cost, secure, and on-demand access to the most competitive computing infrastructures in an IoT network. By creating a distributed cloud infrastructure, the proposed model enables cost-effective high-performance computing. Furthermore, to bring computing resources to the edge of the IoT network and allow low latency access to large amounts of data in a secure manner, we provide a secure distributed fog node architecture that uses SDN and blockchain techniques. Fog nodes are distributed fog computing entities that allow the deployment of fog services, and are formed by multiple computing resources at the edge of the IoT network. We evaluated the performance of our proposed architecture and compared it with the existing models using various performance measures. The results of our evaluation show that performance is improved by reducing the induced delay, reducing the response time, increasing throughput, and the ability to detect real-time attacks in the IoT network with low performance overheads.

INDEX TERMS Internet of things, software defined networking, security, blockchain, cloud computing, fog computing, edge computing.

I. INTRODUCTION

Nowadays, a large number of intelligent devices and objects are integrated with sensors, thus enabling them to detect real-time information from the environment. However, this paradigm has changed with the advent of the IoT in which all intelligent things- such as sensors, laptops, smart cars, smart home devices, and industrial and utility modules- are connected through a network of networks and equipped with data analysis capability, thereby changing the way we play, live, and work. A huge volume of data streams is produced by IoT devices at high speed. According to a recent report by Gartner [1], around one million new IoT devices will be sold every hour and some 2.5 million US dollars will be spent per minute on IoT by 2021. With efficient and flexible provisioning in cloud computing [2], [3], the large volume of IoT data produced by distributed IoT devices can be transmitted to the remote cloud for processing through the internet [4], [5]. However, the internet is neither sufficiently efficient nor sufficiently scalable to deal with these enormous amounts of IoT data. In addition, the transfer of important data is expensive, consuming an enormous amount of bandwidth, time, and energy. Since massive flows of IoT data are transmitted to the cloud at high speed in order to explore valuable information in real time, it is necessary to design an efficient data-processing architecture.

For future generation computing, fog computing is an evolving framework that combines cloud computing and IoT [6]. Currently, cloud-based solutions are being widely researched due to growing demand and the limited computing resources of IoT devices. Recent research predicts that centralized clouds will be unlikely to deliver satisfactory services to customers in the near future. From the core to the edge of the network, fog computing can be viewed as a layered service structure that is an extension of the cloud computing paradigm. It will be able to provide faster cloud services such as storage, computing, and networking capabilities to end users, with each fog node located near the IoT devices at the edge of the IoT network. Fog nodes are distributed fog computing entities that allow the deployment of fog services and are formed by multiple computing resources at the edge of the IoT network. Using different technologies, all the physical devices of a fog node are connected, aggregated and abstract to be considered as a single logical entity that is the fog node, capable of performing distributed services, as it is on a single device. Given a finite network bandwidth, centralized cloud storage is unable to handle huge volumes of data in a timely manner. Due to the limited scope of its vision and resources, a fog node is unable to provide permanent and comprehensive computing services to users [7]. Thus, the secure, scalable, and efficient management of resources may well be one of the most important objectives for the realization of the future IoT network. A distributed peer-to-peer decentralized cloud storage solution is required to achieve the objectives for the future IoT network. Recently, blockchains have recently attracted the attention of researchers in a wide range of industries [8]–[10].

The main reason for this increase in interest in blockchain is that, with the blockchain technique, applications can be operated in distributed ways, whereas previously they had to pass through a trusted intermediary. The recent market research indicates that technologies like blockchain and Bitcoins are the future of the finance domain [11], [12]. By creating trust without the need for a trusted third party, it is widely believed that blockchains will overhaul antiquated cloud computing systems. At present, companies like Amazon offer a decentralized storage infrastructure cloud services. Nowadays, most organizations that have hosted their own servers in their own premises have moved to the cloud to reduce the number of servers and maintenance cost drivers; and by replicating data across multiple data centers, a company like Amazon S3 is able to offer a reliable uptime and redundancy service and charge approximately \$ 25 per terabyte per month. This convenience has blinded us to the extent that we now place too much trust in third parties. Due to a lack of good points of reference and low costs, we are obliged to trust these third parties to secure our most private and sensitive data, which are mostly unencrypted. Due to parallelism between the financial and cloud infrastructures, we can replace the existing systems with blockchains and eliminate our reliance on trusted third parties. In the blockchain based cloud infrastructure, storage hosts sell their surplus storage capacity and renters purchase this surplus capacity and upload files, while payments are made over the blockchain. According to a recent report, the world economic forum's survey predicted that by 2027 some 10% of global GDP may be stored with blockchain technology [13].

A. RESEARCH CONTRIBUTIONS

This paper proposes a flexible, efficient, scalable, and securely distributed cloud architecture that uses SDN, fog computing, and a blockchain technique to gather, classify and analyze IoT data streams at the edge of the network and the distributed cloud. SDN facilitates easy management and network programmability [14]. The main contributions to the research for this work are summarized as laid out below.

- This paper proposes a distributed cloud architecture based on the blockchain technique, which provides low-cost, secure, and on-demand access to the most competitive computing infrastructures in the IoT network.
- It also proposes a secure distributed fog node architecture using SDN and blockchain techniques by bringing computing resources to the edge of the IoT network so that traffic in the core network can be secure and streamlined and have a minimal end-to-end delay between IoT devices and computing resources. In our proposed architecture, security must automatically adapt to the threat landscape, thus dispensing with the need for administrators to manually review and apply thousands of recommendations and opinions at the edge of the network.
- The performance of the proposed technique was evaluated and compared with the existing model with respect to various performance metrics.

B. ORGANIZATION

The rest of the paper is structured as follows: Section II discusses the need for blockchains in distributed cloud storage and the design principles required for a distributed architecture in the IoT network; Section III presents the proposed blockchain-based distributed cloud architecture and secure distributed architecture of the fog node at the edge of the IoT network; Section IV presents the evaluation of the proposed model based on different performance metrics; and Section V presents the conclusions of this research.

II. PRELIMINARIES

A. NEED FOR BLOCKCHAINS IN DISTRIBUTED CLOUD STORAGE

''Blockchain'' is the newest buzzword on the block. Recently, this new concept has caught the attention of many researchers and developers who have recognized the opportunity offered by the blockchain and its potential impact. Based on the present survey, the need for the blockchains technology in the distributed cloud storage is summarized below.

- *Full decentralization and true redundancy:* Using the blockchains technique, it is possible to build a distributed cloud data storage where data is stored in dozens of discrete nodes intelligently disbursed across the globe, making it extremely difficult to cause significant disruptions.

- *Facilitates resource usage:* The blockchains technique can be used to facilitate the use of resources on demand simply by running the on-demand resource algorithm from the smart contract, whereby payment will automatically occur upon completion of the requested service.
- *Complete privacy:* Since, with the blockchains technique, each user manages its own keys and each block node stores only encrypted fragments of user data, it is possible to achieve complete privacy without any third party having access to and control of the data.
- *Improves the Quality-of-Services:* By using the blockchain technique, we can offer traceability of the use of resources in order to properly verify the service level agreement by both the client and the service provider.
- *Cost reduction:* Due to its efficiency and the small costs incurred by each host, compared to Amazon S3's $ 25 per terabyte per month, blockchain storage costs about $2 per terabyte per month [11].

B. REQUIRED DESIGN PRINCIPLES FOR SECURELY DISTRIBUTED ARCHITECTURE IN AN EDGE COMPUTING SCALABLE IoT NETWORK

To design a high-performance architecture in the edge computing scalable IoT network that is securely distributed with the ultimate goal of meeting both current and future challenges and meeting the new service requirements, the following design principles must be taken into consideration:

- *Resilience:* Even if some nodes fail, computation continues on other work nodes.
- *Efficiency*: Even though the computing nodes are very heterogeneous, users receive excellent performance.
- *Ease of deployment:* Even the nodes located on the edge of the internet, allow all the nodes to be used without a specific configuration.
- *Adaptability:* The architecture of the network should be able to adapt to the changing environment and broaden its use to meet the increasing needs and demands of customers.
- *High availability and fault tolerance:* The high availability of a network control system is imperative in the actual operation of the IoT network. Thus, the provisioning of priority redundancies, the identification of failures, and the invocation of alleviation mechanisms are important steps for the activity.
- *Performance (linear adaptability of performance):* In a large scale distributed network architecture, the need to achieve linear performance is an important challenge.
- *Scalability:* Scalability is an important principle in designing a future-proof distributed IoT network architecture so as to manage future growth in the number of devices and the amount of information they produce.
- *Security:* Securing the IoT network is one of the important objectives of designing new distributed architectures. To ensure the holistic design of the network, data protection, confidentiality, and information security must be adequately addressed.

III. DISTRIBUTED BLOCKCHAIN CLOUD ARCHITECTURE

Based on the analysis presented in the previous sections concerning the expansion of the cloud-based IoT networks created by new communication paradigms, we found that existing clouds cannot meet the need to ensure a completely distributed infrastructure for their executions. At the same time, there is growing demand among scientific communities and enterprises for sufficient computing power to process enormous volumes of data and execute substantial applications. Fog computing is an emerging computing model that brings computing abilities to the edge of the distributed IoT network. It is characterized as a distributed computing infrastructure that includes a set of physical machines with high-performance capabilities that are linked to one another. Rather than transferring raw IoT data streams to the cloud, we can locally gather, categorize, and analyze data by deploying a number of fog nodes in the IoT network. This can greatly mitigate traffic in the core network and potentially speed up the processing of large amounts of IoT data. However, the efficient and secure deployment of fog nodes to facilitate communications between fog nodes and IoT devices is always an open problem. The secured deployment of cloud computing ensures that every IoT device has access to computing capabilities everywhere with a low end-to-end delay, without significantly increasing the amount of major network traffic. In this section, we propose a novel blockchain-based distributed cloud architecture with a SDN enabled controller at the edge of the network to meet the required design principles for current and future challenges and to be able to satisfy new service requirements.

A. ARCHITECTURE DESIGN OVERVIEW

Figure 1 presents an overview of the architecture of the proposed model, which is categorized into three layers, i.e. device, fog, and cloud. At the edge of the network, the device layer is used to monitor the various public infrastructure environments and sends the filtered data that is consumed locally to the fog layer and uses the request services. In the fog layer, the device layer transmits the filtered raw data to the fog layer, which consists of high-performance distributed SDN controller. Each fog node covers the small associated community and is responsible for data analysis and service delivery in a timely manner. The fog layer reports the results of processed output data to the cloud and device layers if needed. The fog layer provides localization, while the cloud layer provides wide-area monitoring and control. It is used to provide large-scale event detection, behavioral analysis, and long-term pattern recognition by offering distributed computing and storage. In the cloud layer, we propose a distributed cloud based on the blockchain technique that provides secure, low-cost, and on-demand access to the most competitive computing infrastructures. Clients can search, find, provide, use and automatically free up all the computing resources, such as servers, data, and applications, they need. For the fog layer, we propose the use of a blockchain-based distributed

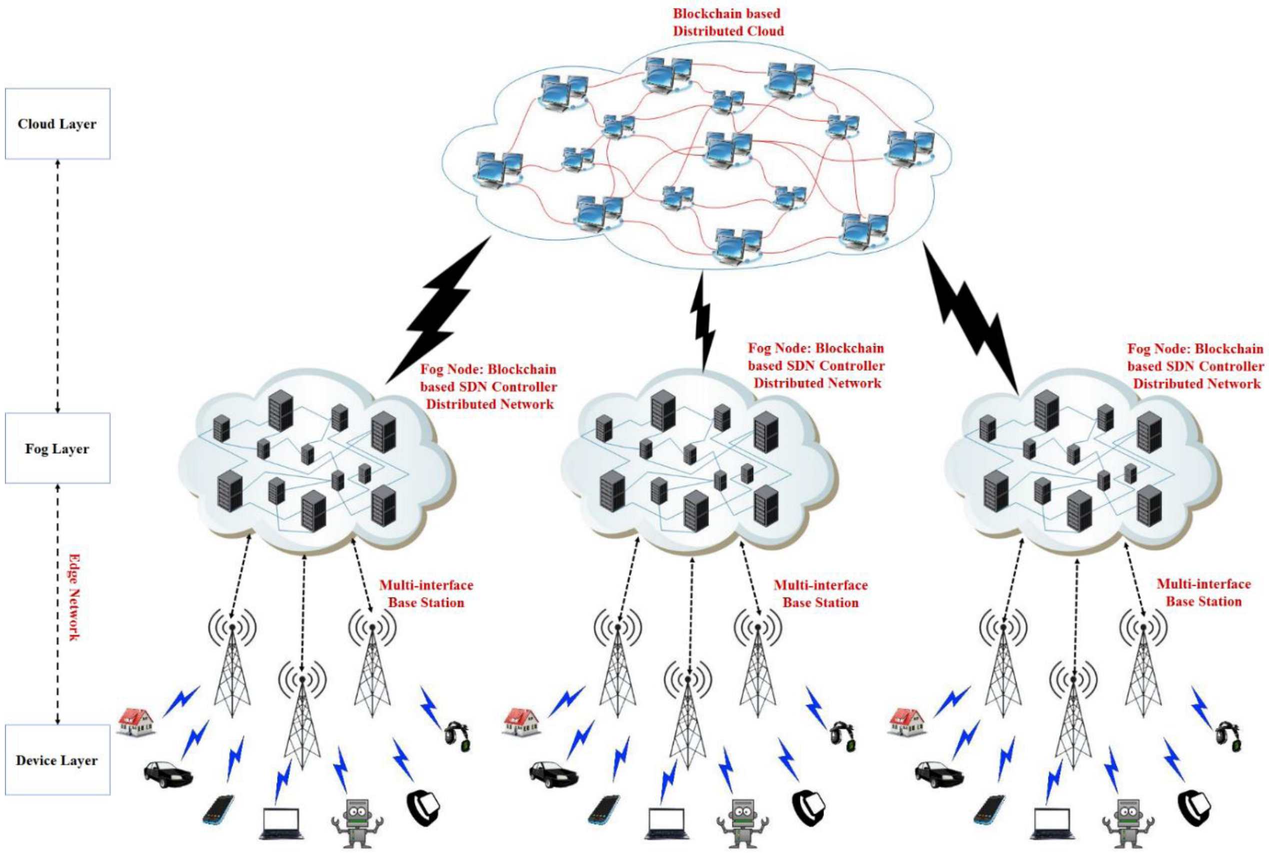

FIGURE 1. Overview of the distributed blockchain cloud architecture.

secure SDN controller network architecture for the fog node. In the fog node, all the SDN controllers are connected in a distributed manner using the blockchain technique. Each SDN controller is empowered by an analysis function of the flow rule and a packet migration function to secure the network during saturation attacks. At the edge of the network we deployed multi-interfaced Base Stations (BSs) equipped with an SDN switch to facilitate the new wireless communication technologies that are based on IoT. To aggregate all of the raw data streams emanating from local IoT devices, we considered the multi-interfaced BSs to constitute a wireless gateway. The BSs act as a forwarding plan for the SDN controllers of the fog node, which monitors traffic at the data plane and establishes user sessions. The SDN controllers in the fog node also provide programming interfaces to network management operators to offer various essential networking capabilities. A fog node can access the distributed cloud over the internet to flexibly deploy the application service and computing availability. Fog nodes can offload their computing workloads to the distributed cloud when they do not have sufficient computing resources to process their local data streams at the expense of increased latency in communications and network resource consumption.

B. DISTRIBUTED BLOCKCHAIN CLOUD ARCHITECTURE

The proposed model opens up new markets by allowing wider use of the existing cloud infrastructure, and brings together data from consumers and producers. The proposed model consists of the following four steps: selection of resource providers, provision of services, registration of transactions, and payment. In the first step, the cloud user must select the resource provider from the service provider pool in the blockchain-based distributed cloud. Once the selection has been made, the selected service provider will then provide the required services, such as task execution, data management, and the provision of servers, to that user. After providing the requested services, the service provider registers the transaction in the form of a blockchain and shares it with all distributed peer service providers. Finally, the user will pay and reward the provider. This model reduces the cost and transparent reputation of resource providers and rewards reliable providers with an integrated quality of service controls that can provide the required level of computing resources. Providing support to different resource providers and making the partial contributions of each provider fully visible also contribute to transparency in this model.

1) PROOF-OF-SERVICE

A conventional blockchain such as Ethereum [15], for example, depends on the Proof-of-Work protocol, which guarantees that token exchanges occurring in the blockchain between members are approved by a vast number of nodes that utilize cryptographic challenges. In our blockchain-based distributed cloud model, a contribution is some action that occurs outside the blockchain, such as the performance of a computation, the transfer of a file, or the provision of a set of data, which will lead to the occurrence of token exchanges between members. This implies that another protocol is required to prove that the contribution has taken place in an accurate manner and that associated token exchanges can occur in the blockchain. We have named this consensus protocol ''Proof-of-Service.'' We gave careful consideration to the approval of contributions because, in an attempt to claim illegitimate rewards, some pernicious clients may try to distort their contributions. In order to design this protocol, we used the 2-hop blockchain technique proposed by Duong *et al.* [16], which combines the mechanisms of proof-of-stakes and proof-of-work. The security of this blockchain depends on whether the legitimate participant controls a greater part of the collective resources, which include both stakes and computing power.

2) MATCHMAKING ALGORITHM

We have used a matchmaking algorithm in our model to link a resource request to a resource offer based on their description. When designing a distributed cloud, the matchmaking algorithm is an essential element in ensuring the provision of resources. It essentially answers the question: ''Can I perform this task on this machine?'' We have stored smart contracts that describe the requirements for performing a task or deploying a VM instance, such as an expected hypervisor, a GPU runtime requirement, RAM, minimal disk space, etc. A matchmaking contract does the matching, perhaps actualizing distinctive sorts of strategies. To implement a matchmaking contract in our model, we used the classified advertisement (classad) matchmaking technique proposed by Raman *et al.* [17] at the University of Wisconsin.

3) SCHEDULING ALGORITHM

In the distributed cloud, a scheduling algorithm distributes a set of enterprise tasks to run on a computing asset. For any distributed computing system, the scheduler is a key component because the performance of the application depends primarily on its efficiency. More specifically, a test consists in designing a multi-criteria scheduler, which is an algorithm that includes several strategies for scheduling tasks and choosing computing resources. For example, one client may want the best performance even if it costs more, whereas another client may want to minimize the cost even if that means that the computation will take longer. In our proposed model, we used CLOUDRB, a technique for managing and scheduling the high-performance computing application in the cloud proposed by Somasundaram and Govindarajan [18]. We used this technique in our model to define their own preferences based on such factors as reliability, confidence, performance, cost, etc.

C. EDGE COMPUTING NETWORK ARCHITECTURE

The majority of the information created by clients' devices contains individual data, such as videos or photos taken on smartphones, GPS data, health data detected by wearable devices, and smart home statuses detected by the sensors installed in a smart home. The analysis of these gigantic amounts of data can benefit the user and society as a whole. A major challenge imposed by the concept of fog is service orchestration. Orchestration includes automated migration, replication, and the instantiation of service instances across many fog nodes with a broad range of functionality. Numerous IoT applications address the dynamic workload produced by event-driven or intermittent data delivery systems. Ideally, applications should be seamlessly sized at run-time without the over-provisioning of resources. The speed and complexity of this development creates new categories of attacks brings together known and mysterious threats, takes advantage of ''zero-day'' vulnerabilities, and uses malicious software concealed in documents and networks.

To address these challenges, we propose a fog computing architecture in the edge network. In the edge network, all the IoT devices' communication to the fog nodes happens through multi-interface BSs. BSs will consider as a gateway or as a forwarding SDN switch for the fog controller, which collects all the data from IoT devices and forwards it to the fog node controller. Each fog node is composed of distributed SDN controllers and uses the blockchain technique to provide scalable, reliable, and high-availability services. Each SDN controller includes packet migration and flow rule analysis functions. The analysis module maintains the main utility of the system foundation during an immersion attack. Note that we leveraged the strengths of the FS-OpenSecurity SDN pragmatic security architecture model based on our previous work [14]. Figure 2 shows the architecture of the SDN controller of the fog node in the edge network. This model consists of three different phases. In the first phase, to build an overall network view, this model monitors and parses to identify essential OpenFlow messages from the arrival OpenFlow packets. In the second phase, it analyzes the parsed data set and extracts the routing topology state and metadata features sets to construct a network flow topology graph with the traffic flow. In particular, our model maintains the topological status of the metadata, outbound flow path design rules, and stores the forwarding of inbound packet headers, etc. In the third phase, the metadata flow of a set validates the allowable metadata values collected over the duration of the flow and management strategies. The model flags known attacks by the administrator-specified strategies, even though it is the most specific flow activities that are carried out over time to detect potentially malicious activity. When the model discovers a new flow behavior, it does not

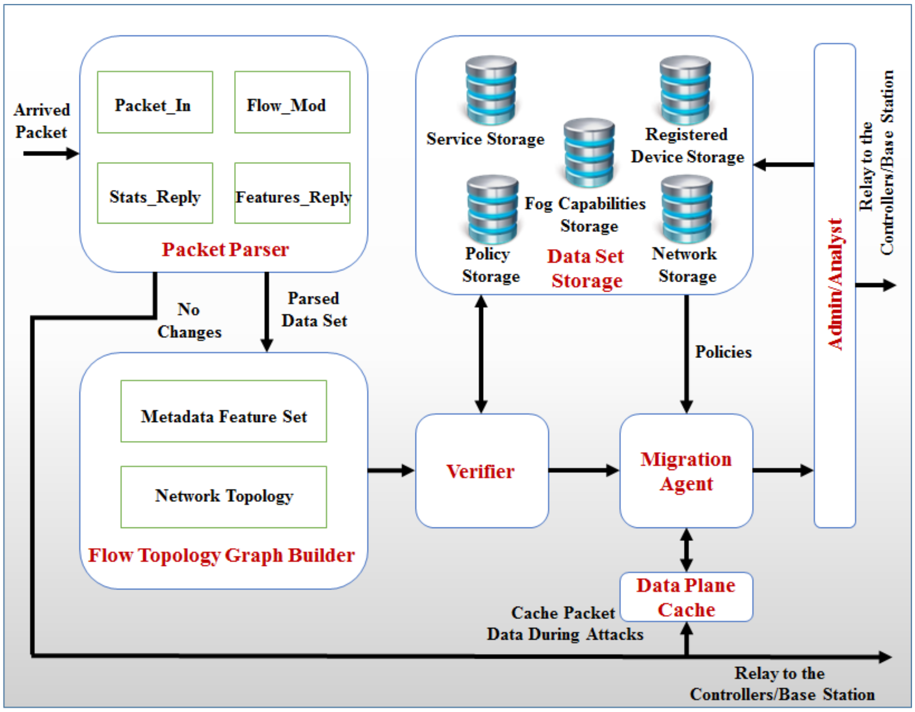

FIGURE 2. Architecture of the SDN controller of the fog node in the edge network.

trigger an alarm: rather, it triggers alarms when it recognizes unreliable entities that cause changes to the behavior of an existing flow or a flow that challenges a specified security policy.

1) PACKETS PARSER

In order to identify abnormal behavior, the model must catch each piece of network data. To modify the network view of the controller, attackers use a subset of all OpenFlow messages. These messages include Features_Reply, Stats_Reply, Flow_Mod and Packet_In. However, to configure the connectivity between endpoints, the trusted controller uses Flow_Mod messages to direct the forwarding devices. Thus, to extract significant metadata, our packets parser dynamically monitors these incoming packets. All reset all packets are simply transmitted.

2) FLOW TOPOLOGY GRAPH BUILDER

These graphs analyze the parsed data set to construct and alter the flow diagrams connected to the network flows. They then distinguish between infringements or attacks in the security strategies perceived by the actual changes made to the topological exchange metadata and to the system data plan related with each flowchart. There are three elements in an SDN domain that precisely outline the metadata for each stream in the network as streams, forwarding switches and end hosts. Our model retrieves and stores the metadata data related to each item to execute a set of features. For all the hosts in the network, the IP/MAC destination and source connections have a match. To recognize the flow between the ends, Port/MAC links exclusively. For each flow, flow statistics offer bytes/packets transferred. In addition, to know the malicious update metadata, our model retains the flows of the physical and logical topology, and the Flow_Mod transmission status messages.

3) VERIFIER

We divided the entire process into two steps: generating path conditions (offline) and generating reactive rules (online). In the path generation step, to generate the path condition, we used the conventional symbolic algorithm to navigate the possible paths and collect all path conditions. To avoid increasing system overhead at runtime, we processed this step offline. In the reactive rule phase, it will monitor and assign the current value of the global variables to the status path. Then, only the input variables are symbolized in the path conditions and the reactive flow rule dispatcher components are used to parse each status path. Only the paths, the final decision is in handling a small set of change to generate a status message is considered. Finally, the reactive flow rules we need are determined.

4) MIGRATION AGENT

Depending on the type of alarms received, this agent recognizes attacks and makes decisions. During saturation attacks, it triggers the flow rule of the parser to generate new rules and to migrate the missing table packet in the data cache. During the flow rule generation and update period, it migrates all missing packets to the data plan cache. Therefore, the flood packets do not flood or overload the controller. Later, once the flow rule is updated, it will process all the missed packets stored in the cache.

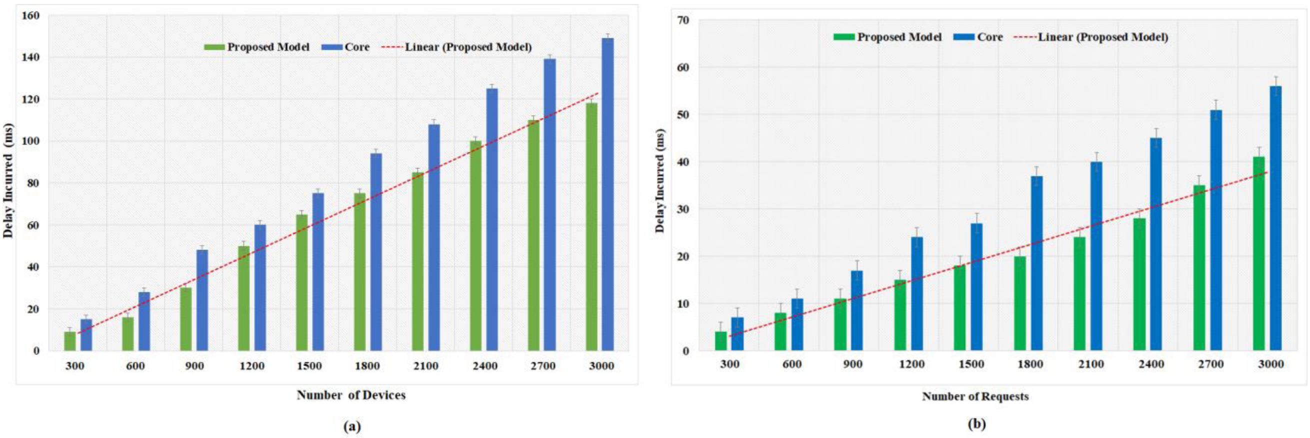

FIGURE 3. Delay incurred by: a) an increase in the number of devices; and b) an increase in the number of requests.

5) DATA PLAN CACHE

This is a temporary storage that caches missing packets during a saturation attack. During flooding attacks, instead of flooding the controller, most flood packages are redirected to the data plan cache. When it receives migrated packets, it parses the packet headers and stores them in the appropriate buffer queue by using the Packet_In generator, the buffer queue, and the classifier.

IV. PERFORMANCE EVALUATION

In this section, we discuss the simulation in different experimental environments in detail and provide our evaluation of the proposed model. We evaluate the proposed scheme by using the throughput, response time and the delay-incurred performance metrics. We also evaluate the accuracy of the proposed model by measuring the speed with which it can detect and mitigate saturation attacks at the edge of the network.

As shown in Fig. 1, numerous IoT devices are connected to the blockchain based distributed cloud environment on the edge of the network and are used to perform various tasks. As discussed in the previous section, to conduct load shedding using an intelligent scheduler in the distributed cloud, different applications/tasks are executed at the closest fog node or distributed cloud, as this lessens the overhead of performing all tasks in the centralized cloud server. We gathered real-time traces of our own distributed private cloud data center and the Amazon EC2 cloud data center. Our own distributed private cloud, which has abundant resources was rendered on 6 desktops, wherein each desktop had 64 GB DDR3 RAM and an Intel i7 processor. At the same time, the end user devices and fog nodes were rendered by laptops with relatively limited computing resources. We considered the laptops to be fog nodes that are linked in a distributed manner using the blockchain technique. Each of the laptops that we used had a 16 GB DDR3 RAM and an Intel i5 CPU. We used the TFN2K tool to generate real-time attacks. TFN2K is a well-known attack tool for generating attacks such as ICMP, TCP / SYN, and UDP flood attacks and widely used to attack several famous websites. For a variety of real-time applications, the delay incurred using the conventional cloud computing infrastructure and the blockchain-based distributed cloud is illustrated in Fig. 3(a). As shown in this figure, the response time for various real-time applications is reduced when the number of interconnected devices is increased.

Fig. 3(b) illustrates the variation in the delay associated with the number of requests produced by the installed clients. The figure shows that there is an increase in the delay as the number of requests for services increases. However, this delay is smaller in the proposed model than in the case using the core cloud infrastructure, which demonstrates the efficiency of our proposed model. By offloading clients' service datasets onto fog nodes from the cloud, all computations are executed at the edge of the network as a result of faster computations.

Fig. 4 (a) demonstrates the response time of end clients awaiting various services from the cloud. We considered that the response time is the first response time after processing the dataset. Compared to the core model of the cloud, the proposed model provides a lower response time when the number of requests is increased. Due to the services provided by the fog nodes, the response time is less than that of the core infrastructure of the cloud. Thus, the proposed model has fewer delays compared to the core infrastructure.

The throughput variation using the core cloud infrastructure and the proposed model is shown in Fig. 4(b). As the results show, compared to the core cloud infrastructure, the proposed model provided a higher throughput. As previously discussed, due to the availability of services at the edge of the network, we achieved improvements. Due to the high availability of resources, the execution time to offload the data to the cloud of applications running on the edge of the network is less.

We also performed the file create, update, delete, and share scenarios to evaluate the performance of our distributed blockchain-based cloud architecture. We created random file names and contents with a size of up to 4 MB with

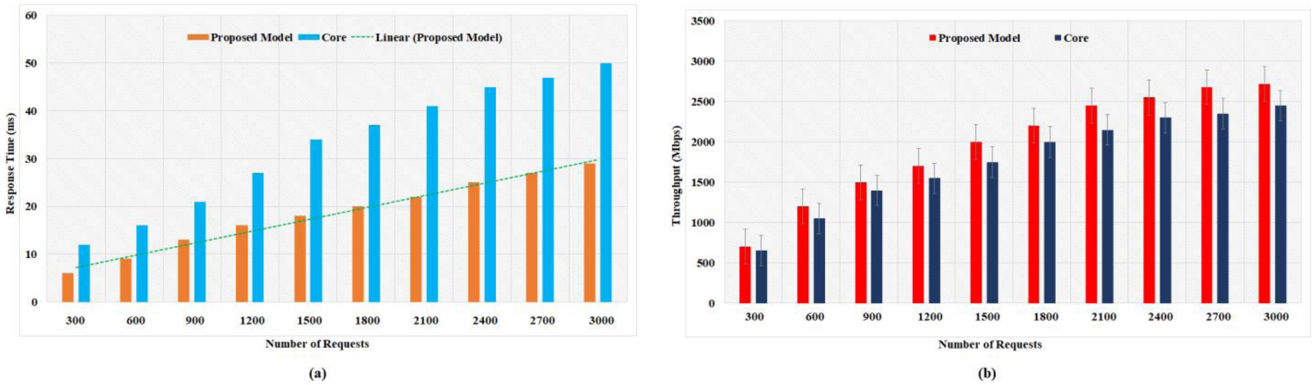

FIGURE 4. With regard to the number of requests: a) the variation in response time; b) the variation in throughput.

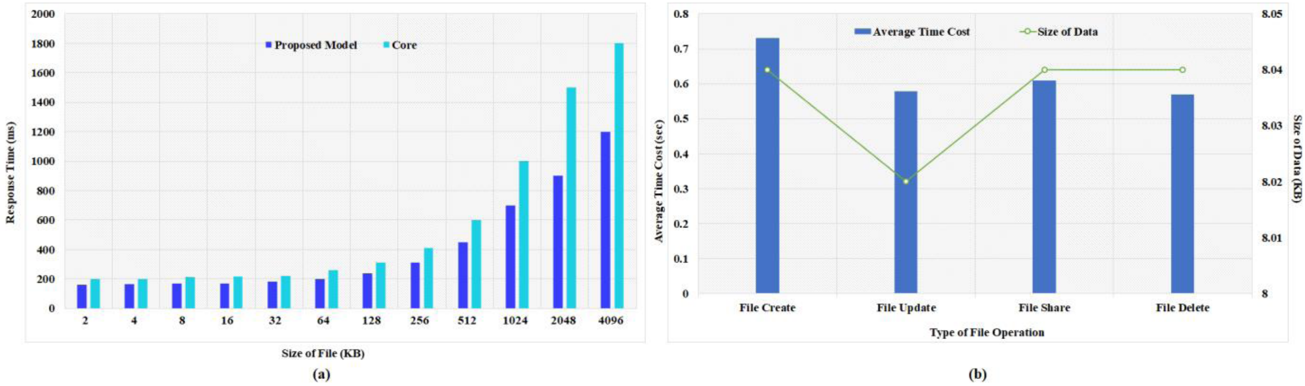

FIGURE 5. File operations: a) average response time with different file size; b) average data retrieval time cost.

300 repetitions. Fig. 5 (a) and Fig. 5 (b) show the average response time with different file sizes and the data retrieval overhead. The results show that the performance of the proposed model was superior to that of the core model, with a minimum overhead.

Based on two different parameters, we evaluated the accuracy rate of the detection of attacks of the proposed model at the fog node. We considered one parameter in the presence of different traffic and many distinctive defects in the network, and another parameter in the real-time identification of the attacks in the network. The proposed model can quickly recognize each attack. In the first case, to increase the number of hosts to 30K, we used Mininet. We then propelled fake topology, ARP poising and DDoS attacks to the blockchain-based SDN controller network fog node in the edge network. On the other hand, synthetic defects were used in parallel with the proper traffic with 6K for the Mininet emulsified hosts on our physical test bench for real-time identification. As such, when the proposed model received the offending packet, the time required to issue an alert is viewed as being the detection time. To produce 1500 Flow_Mod/sec, we used the custom traffic generator. When Packet_In messages are processed, such attacks as false topology and ARP attacks are easily distinguished. To identify DDoS attacks, the detection times may vary because the proposed model occasionally executes the flowchart validator, and, consequently, the size of the flow graph increases. We repeated each experiment approximately 20 times and observed that our proposed model effectively identified each of the issues under the distinctive topologies.

For the given ϑ margin of similarity used to perform outlier detection using the conflicting iperf TCP streams, we ran a pessimistic scenario for the false alerts. At each switch along the flow path, the margin of similarity lies between SI/ϑ and $SI \times \vartheta$. Here, SI is a similarity index at each switch to represent the traffic flow nature, which is calculated at timestamp t as follows:

$$SI_t = SI_{t-1} + \omega,$$

Where ω represents the latest byte-level statistics available at timestamp t. In the current flow path or each flow per switch, we consider SI to be the moving average of the difference in byte-level statistics.

To cause changes in the switches along the flow path, the fairness of the TCP will create fluctuations in flow, which raises the need for some precautions. Fig. 6 (a) shows that the probability of false alarms decreases with the increase of ϑ. Due to the absence of a true positive, both recall and precision are zero. As shown in this figure, 7 alarms out of 10 concurrent streams over 6 minutes at the default value of $\vartheta = 1.06$.

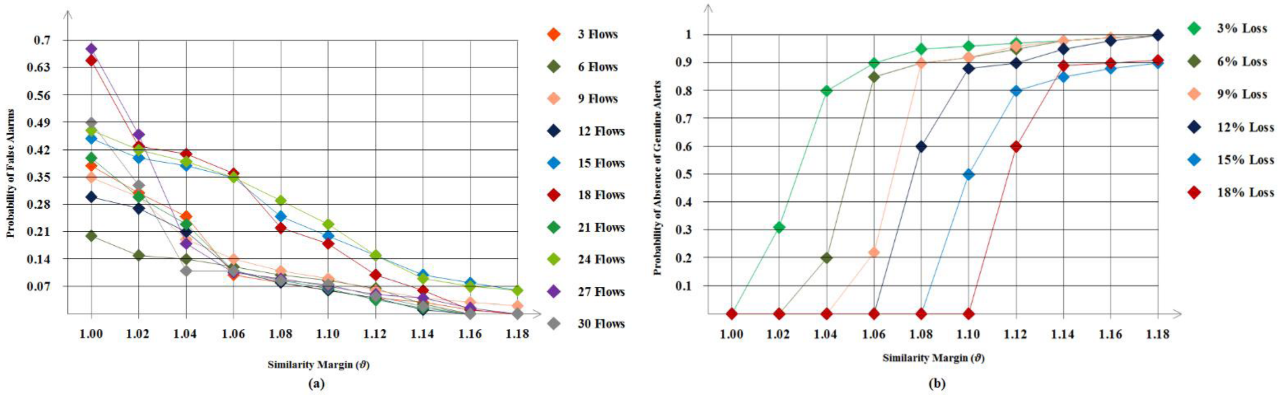

FIGURE 6. Accuracy rate of the proposed model: a) probability of false alarms with in flows and ϑ; b) probability of absence of genuine alerts vs. loss rate and ϑ.

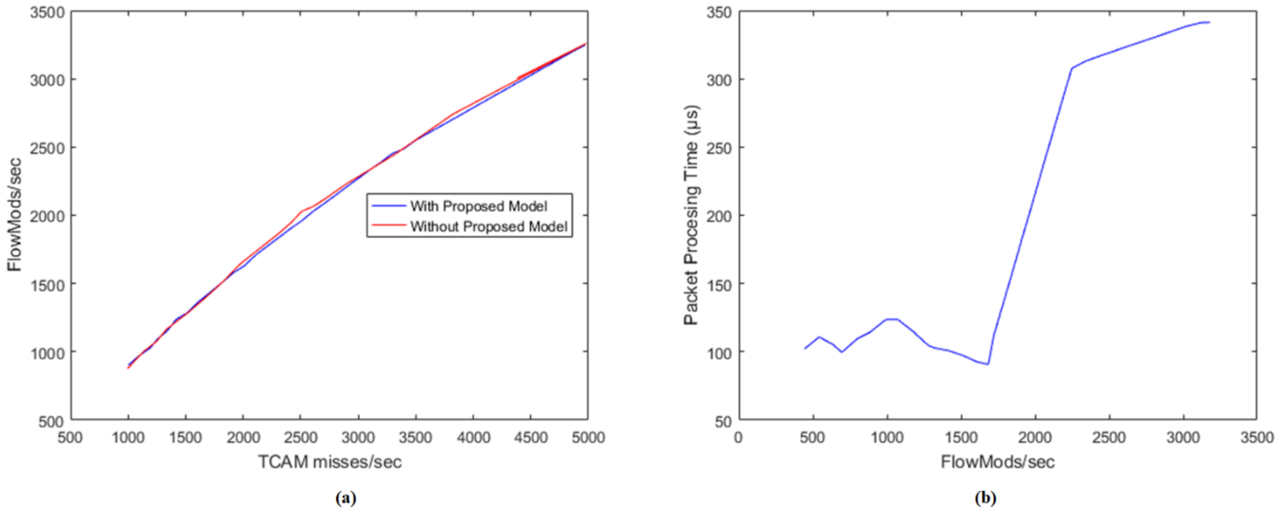

FIGURE 7. Proposed model performance overhead at fog node: a) Flow_Mod throughput vs TCAM miss rates; b) variation in Flow_Mod processing time.

For a given ϑ, in order to evaluate the absence of genuine alerts, we defined the proportion between the number of checks that did not produce alerts and the total number of checks that raised alerts throughout the verification. We assessed this performance metric in the Mininet hosts for controlled flows, which are 8 hops apart. The absence of genuine alerts throughout the verification increases as δ increases as shown in Fig. 6(b). The precision and the recall are identical for a given ϑ, which is equal to 1- the probability of absence of genuine alerts at each data point.

For each SDN controller in the fog node of the proposed model, we evaluated the overhead of processing the Flow_Mod messages. With an increasing rate of incoming TCP connections, we observed the Flow_Mod throughput without and with the proposed model. Each TCP packet results in an absence of Ternary addressable memory (TCAM), which then produces a Packet_In and elicits a Flow_Mod message from the controller. Fig. 7 (a) shows the results where, even at a high Packet_In (or TCAM rate), the proposed model retains a high Flow_Mod flow. The throughput is simply compelled by the overhead of the controller, which is clearly seen by the fact that, whether with pr without the proposed model, the Flow_Mod throughputs are practically equivalent. In addition, we evaluated the Flow_Mod processing times as shown in Fig. 7 (b). We noticed that the processing time is more than 100 μs for the 2K Flow_Mod/sec flow rate, but that it increases as the flow increases. It increases processing times because at higher Flow_Mod speeds, the controller piggybacks many OpenFlow messages in the same TCP payload.

V. CONCLUSION

In this paper, we proposed a new distributed blockchain cloud architecture model to meet the design principles required to efficiently manage the raw data streams produced by large IoT devices in the distributed cloud and at the edge of the network. It is based on three emerging technologies: fog computing, SDN, and blockchain. The proposed architecture was designed to support high availability, real-time data delivery, high scalability, security, resiliency, and low latency. To facilitate the provision of IoT services, the proposed architecture can significantly reduce the end-to-end delay between IoT devices, computing resources and traffic load in the core network compared to the traditional IoT architecture. The results of our performance evaluation clearly indicate that, compared to the traditional core-based cloud computing infrastructure, our model is a more efficient solution for offloading data to the cloud. It also demonstrates the efficiency and effectiveness of the proposed model and that it meets the required design principles with minimal overhead.

In the future, we will explore the various energy harvesting technique aspects of our proposed model for energy efficient communication among devices at the edge of the IoT network.

REFERENCES

[1] *Top Strategic Predictions for 2017 and Beyond: Surviving the Storm Winds of Digital Disruption*. Accessed: Oct. 28, 2017. [Online]. Available: https://www.gartner.com/doc/3471568?ref=unauthreader

[2] S. Singh, Y.-S. Jeong, and J. H. Park, "A survey on cloud computing security: Issues, threats, and solutions," *J. Netw. Comput. Appl.*, vol. 75, pp. 200–222, Nov. 2016.

[3] X. Sun, N. Ansari, and R. Wang, "Optimizing resource utilization of a data center," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 4, pp. 2822–2846, 4th Quart., 2016.

[4] H. L. Truong and S. Dustdar, "Principles for engineering IoT cloud systems," *IEEE Cloud Comput.*, vol. 2, no. 2, pp. 68–76, Mar. 2015.

[5] L. Wang and R. Ranjan, "Processing distributed Internet of Things data in clouds," *IEEE Cloud Comput.*, vol. 2, no. 1, pp. 76–80, Jan. 2015.

[6] M. Chiang and T. Zhang, "Fog and IoT: An overview of research opportunities," *IEEE Internet Things J.*, vol. 3, no. 6, pp. 854–864, Dec. 2016.

[7] A. Alrawais, A. Alhothaily, C. Hu, and X. Cheng, "Fog computing for the Internet of Things: Security and privacy issues," *IEEE Internet Comput.*, vol. 21, no. 2, pp. 34–42, Mar. 2017.

[8] K. Christidis and M. Devetsikiotis, "Blockchains and smart contracts for the Internet of Things," *IEEE Access*, vol. 4, pp. 2292–2303, May 2016.

[9] F. Tschorsch and B. Scheuermann, "Bitcoin and beyond: A technical survey on decentralized digital currencies," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 3, pp. 2084–2123, 3rd Quart., 2016.

[10] P. K. Sharma, S. Y. Moon, and J. H. Park, "Block-VN: A distributed blockchain based vehicular network architecture in smart city," *J. Inf. Process. Syst.*, vol. 13, no. 1, pp. 184–195, Mar. 2017.

[11] Z. Herbert. *Why Blockchains are the Future of Cloud Storage*. Accessed: Aug. 24, 2017. [Online]. Available: https://blog.sia.tech/why-blockchains-are-the-future-of-cloud-storage-91f0b48cfce9

[12] B. Butler. *How IBM Wants to Bring Blockchain From Bitcon to Your Data Center*. Accessed: Aug. 24, 2017. [Online]. Available: http://www.networkworld.com/article/3182806/cloud-computing/how-ibm-wants-to-bring-blockchain-from-bitcoin-to-your-data-center.html

[13] World Economic Forum. (Sep. 2015). *Deep Shift Technology Tipping Points and Societal Impact*. [Online]. Available: http://www3.weforum.org/docs/WEF_GAC15_Technological_Tipping_Points_report_2015.pdf

[14] Y. Sung, P. K. Sharma, E. M. Lopez, and J. H. Park, "FS-OpenSecurity: A taxonomic modeling of security threats in SDN for future sustainable computing," *Sustainability*, vol. 8, no. 9, p. 919, Sep. 2016.

[15] G. Wood, "ETHEREUM: A secure decentralised generalised transaction ledger," Tech. Rep., 2014, Accessed: Oct. 28, 2017. [Online]. Available: http://gavwood.com/Paper.pdf

[16] T. Duong, L. Fan, and H.S. Zhou, "2-hop blockchain: Combining proof-of-work and proof-of-stake securely," Cryptol. ePrint Arch., Tech. Rep. 2016/716, 2016, Accessed: Oct. 28, 2017. [Online]. Available: https://eprint.iacr.org/2016/716.pdf

[17] R. Raman, M. Livny, and M. Solomon, "Matchmaking: Distributed resource management for high throughput computing," in *Proc. Int. Symp., High Perform. Distrib. Comput.*, Jul. 1998, pp. 140–146.

[18] T. S. Somasundaram and G. Kannan, "CLOUDRB: A framework for scheduling and managing high-performance computing (HPC) applications in science cloud," *Future Generat. Comput. Syst.*, vol. 34, pp. 47–65, May 2014.

PRADIP KUMAR SHARMA received the dual master's degree in computer science from Tezpur University and Thapar University, India, in 2012 and 2014, respectively. He was a Software Engineer with MAQ Software, India. He is a Ph.D. Scholar with the Seoul National University of Science and Technology. He is with the Ubiquitous Computing and Security Research Group under the supervision of Prof. J. H. Park. He was involved on variety of projects, proficient in building large-scale complex data warehouses, OLAP models and reporting solutions that meet business objectives and align IT with business. His current research interests are focused on the areas of ubiquitous computing and security, cloud computing, SDN, SNS, and IoT. He is also a Reviewer of the *Journal of Supercomputing* and the IEEE INTERNET OF THINGS JOURNAL.

MU-YEN CHEN received the Ph.D. degree in information management from National Chiao-Tung University, Taiwan. He is a Professor of information management with the National Taichung University of Science and Technology, Taiwan. His current research interests include artificial intelligent, soft computing, bio-inspired computing, data mining, deep learning, context-awareness, machine learning, and financial engineering, with over 100 publications in these areas. He has co-edited 12 special issues in international journals, such as the *Computers in Human Behavior*, the *Applied Soft Computing*, the *Soft Computing*, *Information Fusion*, the *Neurocomputing*, the *Journal of Medical and Biological Engineering*, *The Electronic Library*, the *Library High Tech*. He has served as an Editor-in-Chief and an Associate Editor of international journals, such as the *International Journal of Big Data and Analytics in Healthcare*, the *Journal of Information Processing Systems*, the *International Journal of Social and Humanistic Computing* while he is an editorial board member on several SCI journals.

JONG HYUK (JAMES J.) PARK (M'10) received the Ph.D. degrees from the Graduate School of Information Security, Korea University, South Korea, and the Graduate School of Human Sciences, Waseda University, Japan. From, 2002 to 2007, he was a Research Scientist with the Research and Development Institute, Hanwha S&C Co., Ltd., South Korea. From 2007 to 2009, he was a Professor with the Department of Computer Science and Engineering, Kyungnam University, South Korea. He is currently a Professor with the Department of Computer Science and Engineering and the Department of Interdisciplinary Bio IT Materials, Seoul National University of Science and Technology (SeoulTech), South Korea. He has authored about 200 research papers in international journals and conferences. His research interests include security and digital forensics, human-centric ubiquitous computing, context awareness, multimedia services, human-centric ubiquitous computing, vehicular cloud computing, information security, digital forensics, secure communications, and multimedia computing. He is a member of the IEEE Computer Society, KIPS, and KMMS. He received the best paper awards from ISA-08 and ITCS-11 conferences and the outstanding leadership awards from the IEEE HPCC-09, ICA3PP-10, IEE ISPA-11, and PDCAT-11. He received the outstanding research awards from the SeoulTech, in 2014. He has been serving as chairs, program committee, or organizing committee chair for many international conferences and workshops. He is the Founding Steering Chair of some international conferences, such as MUE, FutureTech, CSA, and UCAWSN. He is an associate editor/editor of 14 international journals including eight journals indexed by SCI (E). He has been serving as a Guest Editor of international journals by some publishers: Springer, Elsevier, John Wiley, Oxford Univ. press, Hindawi, Emerald, and Inderscience. He is an Editor-In-Chief of the *Human-Centric Computing and Information Sciences* (Springer), the *Journal of Information Processing Systems* (KIPS), and the *Journal of Convergence* (KIPS CSWRG).