IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received May 7, 2020, accepted July 29, 2020, date of publication August 6, 2020, date of current version August 19, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3014622

SafeCity: Toward Safe and Secured Data Management Design for IoT-Enabled Smart City Planning

HUI ZHANG[1,2], MUHAMMAD BABAR[3], MUHAMMAD USMAN TARIQ[3], MIAN AHMAD JAN[4,5], VARUN G. MENON[6], (Senior Member, IEEE), AND XINGWANG LI[7], (Senior Member, IEEE)

[1]School of Energy Science and Engineering, Henan Polytechnic University, Jiaozuo 454003, China
[2]Coal Mining and Design Branch, China Coal Research Institute, Beijing 100013, China
[3]Department of Management, Abu Dhabi School of Management, Abu Dhabi, United Arab Emirates
[4]Informetrics Research Group, Ton Duc Thang University, Ho Chi Minh City, Vietnam
[5]Faculty of Information Technology, Ton Duc Thang University, Ho Chi Minh City, Vietnam
[6]Department of Computer Science and Engineering, SCMS School of Engineering and Technology, Ernakulam 683576, India
[7]School of Physics and Electronic Information Engineering, Henan Polytechnic University, Jiaozuo 454003, China

Corresponding author: Mian Ahmad Jan (mianjan@tdtu.edu.vn)

This work was supported in part by the National Natural Science Foundation General Fund under Project 51874109, in part by the Key Scientific and Technological Projects in Henan Province under Grant 182102310005, in part by the Science and Technology Support Plan of Guizhou Province (Science Support of Guizhou Province) under Grant [2019] 2861, in part by the Science and Technology Project for Outing and Young Talents of Guizhou (Talents of Science Platform in Guizhou) under Grant [2019] 5674, in part by the Key Scientific Research Projects of Higher Education Institutions in Henan Province under Grant 20A510007, and in part by the Fundamental Research Funds for the Universities of Henan Province under Grant NSFRF180309.

ABSTRACT The interaction among different Internet of Things (IoT) sensors and devices become massive and insecure over the Internet as we probe to smart cities. These heterogeneous devices produce an enormous amount of data that is vulnerable to various malicious threats. The generated data need to be processed and analyzed in a secure fashion to make smart decisions. The smart urban planning is becoming a reality through the mass information generated by the Internet of Things (IoT). This paper exhibits a novel architecture, SafeCity, that limelight the ecosystem of smart cities consists of cameras, sensors, and other real-world physical devices. SafeCity is a three-layer architecture, i.e., a data security layer, a data computational layer, and a decision-making layer. At the first layer, payload-based symmetric encryption is used to secure the data from intruders by exchanging only the authentic data among the physical devices. The second layer is used for the computation of secured data. Finally, the third layer extracts visions from data. The secured exchange of data is ensured by using Raspberry Pi boards while the computation of data is tested on trustworthy datasets, using the Hadoop platform. The assessments disclose that SafeCity presents precious insights into a secured smart city in the context of sensors based IoT environment.

INDEX TERMS Internet of Things, smart city, symmetric encryption, data management design, data analytics, data mining.

I. INTRODUCTION

Currently, 55% population of the world is in the cities that are expected to grow up to 67% by the year 2050 [1], [2]. The gradual increase in the urbanization poses various encounters for the decision-makers in proposing different facilities to the inhabitants of these cities. The ICT (Information and Communication Technologies) are used to make the cities smart enough by deploying and promoting sustainable development practices for addressing the growing challenges of urbanization. A solid foundation is offered for the Internet of Things (IoT) with an advancement in the field of smart cities' sensors by enabling them to interconnect [3]. Technology in the shape of smartphones, sensors, and other devices is playing a pivotal role in bringing the era of ubiquitous computing. In 2017, Gartner predicted that the number of interconnected devices will increase by 31% in 2017 by getting 8.5 billion and exceeded 20+ billion by the year 2020.

The IoT-enabled environment is a pattern where the processing of information is connected with every encountered

The associate editor coordinating the review of this manuscript and approving it for publication was Jesús Hamilton Ortiz.

activity [4]. A huge number of real-world physical devices in a ubiquitous environment will generate voluminous data containing a variety of information that needs new forms of computation to facilitate enhanced decision making. The vast amount of data generated by the ubiquitous devices will add veracity, value, and variability to the Internet [5]. Advancement in the ubiquitous computing is causing in a large-scale valuable data or information, and with the assistance of Big Data tools and proficient machine learning methods, there is a great potential of analytical amenities to the smart cities [6]–[9]. A number of proposals are found to process and analyze the data generated by heterogeneous devices to perform efficient decision making.

Smart city data computation and pervasive intelligence expose the networks to security attacks, malware, and other cyber breaches. The inter-connectivity requirements of everyday physical devices would probably add numerous groundbreaking and resourceful malicious prototypes to IoT data computing [10]. The presence of malicious intruders may generate fabricated data to manipulate the sensed information of legitimate devices. The intruders may adversely affect the services and decision making in a ubiquitous environment. Furthermore, these malicious entities may liftoff attacks like denial-of-service by disrupting the transmission, and sensing of a ubiquitous environment to reduce the eminence of smart services [11].

Security provisioning in a ubiquitous environment is an intricate work since every machine possesses its identifiable unique characteristics and the uniqueness to be verified when connected to the Internet. The solutions for these ubiquitous devices in the marketplace lack the secured characteristics and are exposed to an extensive kind of adversarial attacks [12]. Besides, the existing privacy-preserving and authentication algorithms for smart ubiquitous environments involve complex and resource-intensive operations that require an abundance of resources. Most of these algorithms are not suitable for delay-sensitive and priority-based traffic generated in these environments.

In this article, we propose a safe and secured data management design for smart city planning using ubiquitous computing. The key contributions of the proposed architecture are as follows.

1. Payload-based symmetric encryption is proposed for a smart ubiquitous environment that is simple, lightweight, robust, and resilient against various malicious threats. The proposed approach uses 128-bit security primitives for secured exchange of data among the real-world physical devices.
2. A customized utility is proposed for the efficient loading of secured data into Hadoop. The proposed loading utility is efficient in terms of time and storage. The default HDFS (Hadoop Distributed File System) architecture is customized to achieve effective data storage. Our customized HDFS reduces storage consumption along with the network overhead.
3. The traditional YARN (Yet Another Resource Negotiator) Hadoop definition is customized for efficient data computation. This is accomplished by introducing the concept of dynamic scheduling into the Hadoop YARN definition.

The remaining paper is ordered as follows. In Section 2, we spotlight the existing studies. In Section 3, we spotlight our proposed SafeCity framework for an IoT sensors based environment. In Section 4, the experimental results for secured data transmission and processing are presented. Finally, the paper is concluded in Section 5.

II. LITERATURE REVIEW

In this section, first we highlight the current works about the secure transmission of ubiquitous data collected from the smart cities, followed by their processing to extract valuable features.

A. SECURED TRANSMISSION OF DATA

Over the last decade, a lot of hype has been witnessed around building the concept of smart cities. Finally, the presence of sensor-embedded Internet of Things (IoT) platforms, ubiquitous connectivity, and cloud and data analytics has turned this concept into a reality. Although cities around the globe are seeking to become smarter, the applications of smart cities face a plethora of challenges in terms of security and privacy. These applications need to secure the gathered data from unauthorized access, disruption, annihilation, modification, inspection, and various other malevolent activities. In literature, numerous studies exist to protect the voluminous data traffic of smart ubiquitous cities from malicious entities. The error-prone communication channels used by the resource-starving sensors of smart cities limit the usage of TLS (Transport Layer Security) for seamless traffic flow [13]. As a result, most of the sensor nodes in smart ubiquitous environments rely on DTLS (Datagram Transport Layer Security) for the secured transmission of their data [14]. Nonetheless, the record layers of DTLS and handshake have a collective overhead of 25 bytes in each datagram header. The DTLS needs to be stripped of the resource-intensive operations to suit the resource-starving sensor nodes of smart cities [15].

In [16], the authors proposed an extremely lightweight encryption approach for the secured establishment of a unicast communication system in smart cities. The authors claimed that their model decreases the energy consumption and computational time of the sensor nodes. However, they did not provide any experimental and analytical results to verify their claim. In [17], the authors studied the use of DTLS for secured communication in a smart ubiquitous environment. They argued that the streaming applications of smart cities require an abundance of memory space and the use of DTLS is not feasible for them. The authors emphasized the use of compressed IPSec to offer security at the network layer for streaming applications.

A robust and resilient secured scheme for ubiquitous applications of smart cities was proposed in [18]. An RSA-based DTLS implementation was used for the secured exchange of ubiquitous data. However, both the RSA and DTLS have higher computational overheads due to resource-intensive handshake mechanisms. The presence of complex cipher suites of RSA incurs a higher energy consumption and computational overhead for the ubiquitous operation of sensor nodes. The performance of the DTLS handshake was evaluated for ubiquitous smart devices using the Elliptic Curve Cryptography (ECC) [19].

In [20], a DTLS implementation for smartphones was proposed using the Constrained Application Protocol (CoAP). The proposed scheme involves computationally difficult encryption suites, requires ample processing and power memory, and is not suitable for sensor nodes of the smart cities. In [21], a lightweight encryption approach was proposed for ubiquitous communication in a smart city environment. Prior to establishing a secured session, the proposed approach validates the identities of clients and servers. For authentication, symmetric encryption with 128- bit security primitives were used. However, the proposed scheme is not validated experimentally to verify its efficiency, robustness, and resilience.

B. DATA PROCESSING AND FEATURE EXTRACTION

In this section, the challenges and issues in the existing works for smart city planning utilizing the Big Data analytical techniques are presented. In [22], the authors designed a model to compute Big Data generated in the loT-based smart health setting. It involves the separation of vigorous data into subclasses that are based on hypothetical simulation of data fusion to improve computational effectiveness. The key issues underlined in this model are the use of customary MapReduce Cluster management for Apache Hadoop server, insufficient data loading to Hadoop, a conceptual framework, and the utilization of only healthcare datasets.

A Big Data analytics framework comprised of various tiers was proposed for urban planning in [23]. Each tier of the framework is responsible for different activities of the Big Data analytics to have efficient modularization of the overall process. Although, it is a complete framework from data generation and collection to application and usage of the analyzed data, it causes significant delay in processing and the use of classical MapReduce deteriorates the performance [24]. Moreover, prior to data loading, the authors focused on data aggregation while overlooking the data loading competence.

An IoT-enabled framework using Hadoop-based Big Data analytics was proposed in [25] for a smart city application. The proposed framework has different layers from data acquisition to the application. The main problem of this framework is that the data loading efficiency was ignored.

A proposal based on the analysis of Big Data that endorses the perception of SCC (smart and connected societies) for smart cities was proposed in [26]. The SCC model is a conceptual framework that was not implemented. A similar model was proposed for the ubiquitous smart city application in [27]. However, this model was not implemented as well. Moreover, [26] and [27] overlooked the data loading and ingestion into a distributed ubiquitous smart city environment. In addition, many solutions have been proposed to treat similar problems of Big Data analytics in smart ubiquitous environments [28], [29]. Vecular fog computing may also be utilized for smart city planning [30]. However, a critical issue in the design of these methods is the deployment of a traditional cluster resource management scheme and insufficient data loading to the Hadoop server.

A graph-oriented architecture to analyze the Big Data in a smart ubiquitous transportation system was proposed in [31]. This graph-based solution is more scalable and efficient, but it incurs additional delay due to graph processing. In addition to processing delay, the proposed solution was tested only for the transportation dataset, and loading the Big Data to the Hadoop server and its efficiency was overlooked. The proposed architecture was tested only for a healthcare dataset. The authors proposed a multi-level data processing scheme, based on parallel processing, for Big Data analysis. However, a YARN-enabled solution was provided but the data ingestion efficacy was ignored.

III. A SAFE AND SECURED DATA MANAGEMENT FRAMEWORK

For a smart and safe city to perform intelligent and secure decisions, the ubiquitous data collected by the devices are processed using different approaches. In SafeCity, the data analysis and machine learning approaches are applied to the data generated and acquired in a ubiquitous environment. The acquisition is carried out by systems that convert the analog information into digital. The cellular technology, i.e, 4G/LTE, is used as a ridging technology between the users, devices, and the system, as shown in Figure 1.

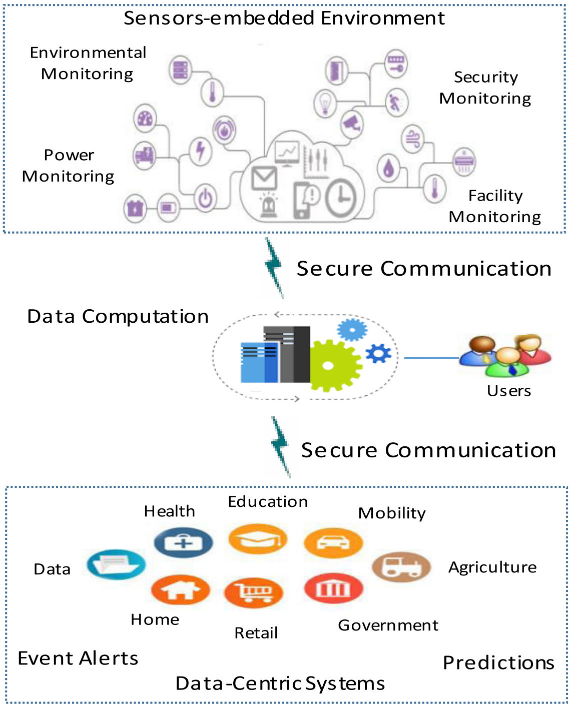

FIGURE 1. Overview of the proposed system.

To design a ubiquitous environment, numerous surveillance cameras, wired and wireless sensors, and devicemounted sensors are deployed. Data sensing, acquisition, and collection are performed in this environment. Digital loggers and digital data acquisition systems are used to detect and collect data from devices and disseminate them with the help of the Internet. The produced ubiquitous data are secured before forwarding to a computational unit for safe and secured processing and transmission. Afterward, the decisions are made on the secured ubiquitous data. The proposed system is a three-layer architecture, i.e., a ubiquitous data security layer, a ubiquitous data computation layer, and a decision-making layer. A payload-based authentication approach is utilized in the first layer to make the ubiquitous data secured from adversaries.

This layer ensures that only secured data is forwarded. The second layer is accountable for the resource-intensive processing of secured ubiquitous data at the conventional computing platforms. Finally, the third layer provides insights from the ubiquitous data and makes smart decisions. The proposed architecture is shown in Figure 2. The comprehensive description of each layer is given in the following subsections.

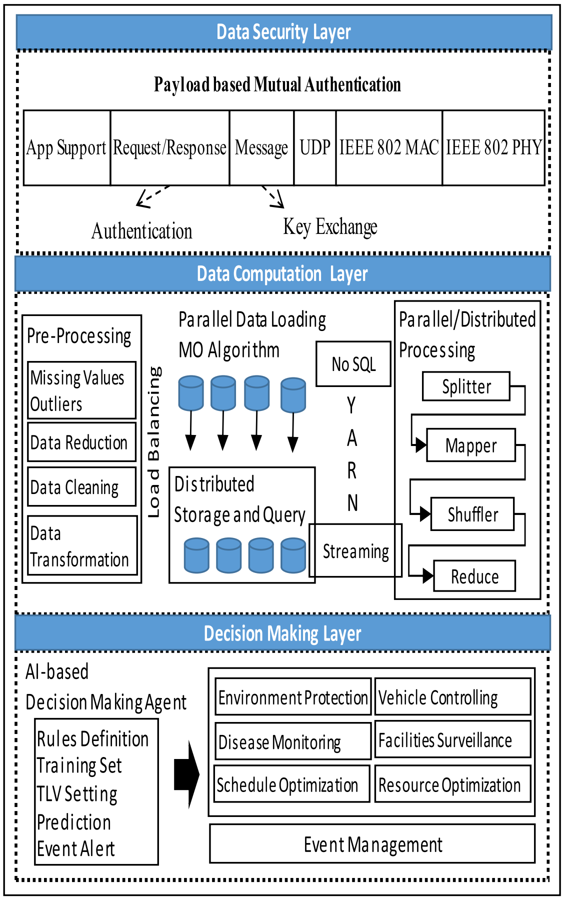

FIGURE 2. System architecture of SafeCity.

A. DATA SECURITY LAYER

This layer of SafeCity is linked to the data sources. The data received from the sensors are in the form of messages. At this layer, message identification and authentication are performed using a simple payload-based authentication scheme. The proposed scheme uses the CoAP protocol [32] for message exchange and authentication at the application layer of each data source. In ubiquitous environments, most of the CoAP-based solutions are relied on the use of DTLS to ensure the protected transfer of resources between the devices. However, the DTLS-enabled CoAP stack incurs an excessive computational and communication overhead. Furthermore, the use of DTLS in combination with CoAP adds an extra layer of protocol header for security provisioning. In our approach, the security of data messages is not compromised while transferred between clients and servers. The session key is transmitted within the payload messages while authentication is achieved at the request-response communication, as shown by the top layer in Figure 2. In SafeCity, CoAP is equipped with secured features for authentication, efficiency, robustness, and defense against a number of malevolent threats.

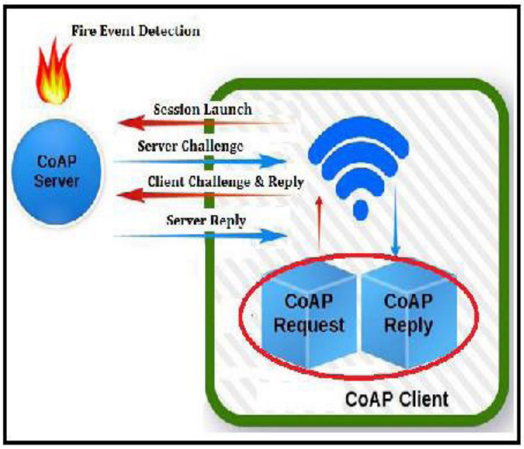

FIGURE 3. Mutual authentication.

During the authentication process, the resource-constrained clients communicate with a server to verify each other identities. As an example, the ubiquitous clients of Figure 1 observe various events such as, temperature, humidity, pollution, and fire eruption, at the server. For a server to provide access to the residing resources, both the parties need to be mutually authenticated. In SafeCity, the authentication is accomplished using four handshake messages. A maximum of 256-bits is used within the payload of each message. The four handshake messages are session launch, server challenge, client challenge and reply, and server reply, as shown in Figure 3. The session launch is headed by a provisioning stage where the clients share a secret key with the server. The server conserves a trace of keys, based on an associated unique identifier (ID). The exchange of a session key between the client and server takes place upon successful authentication. For each client, a session key is implanted on the device at the manufacturing time. If an impostor strives to rage the client, a specific alarm is spawned to notify the crack. To encode the payload of authentication information, the Advanced Encryption Standard (AES) is utilized.

During the session launch, a secret key λ_i is shared with the server, where λ_i is 128-bit long. The λ_i is identified only by client$_\mathrm{i}$ (it belongs) and server, where $i \in \{1,2,3,\ldots,I\}$. Each i has a unique identifier that helps the server to execute a look-up table for verification of identity. The session launch is similar to a Hello message and its payload consists of CoAP options fields, i.e., **Auth** and **Auth-Msg-Type**, to indicate the type of operations performed between the client and a server. After the session launch, the next step is the server challenge, in which the server creates a challenge for the client. The encounter containing a pseudo-random nonce η_r and a session key μ produced by the server. The following equations are used to create a challenge.

$$\vartheta = \lambda_i \oplus \mu \tag{1}$$

$$\mathrm{C_r} = \mathrm{AES}\{\lambda_i, (\vartheta|\eta_r)\} \tag{2}$$

where, i is the ID of a client, ϑ is the intermediate value generated by the server, and $\mathrm{C_r}$ is the challenge sent to i. In the client challenge and reply message, the client retrieves η_r and λ from the server challenge and creates a challenge in response using the following equations.

$$\vartheta' = \eta_r \oplus \lambda_i \tag{3}$$

$$\mathrm{C}_i = \mathrm{AES}\{\mu, (\vartheta'|\eta_i)\} \tag{4}$$

where η_i is the pseudo-random nonce and ϑ’ is the intermediate value generated by the client, and C_i is the challenge sent to the server. Upon receiving, the server tries to retrieve η_r from the client’s challenge. If this nonce is present, the status of i changes to **Authenticated,** and the server responses to the client’s challenge to complete the authentication process, using the following equation.

$$\mathrm{C_r} = \mathrm{AES}\{\lambda_i, (\eta_r|\mu)\} \tag{5}$$

B. DATA PROCESSING AND COMPUTATION LAYER

Versatile analysis and intelligent processing on huge data streams can be unrealistic and infeasible if the data streams are not properly pre-processed. Data pre-processing are performed prior to the core computation and processing. The pre-processing steps involve the reduction to realize the reduced data with similar properties, data transformation to standardize data to an appropriate arrangement for processing, and data cleansing. These activities are carried out using machine learning approaches. The objective is to dig out the data about various sets of an IoT domain, based on its characteristics. Next, the data loading is performed using multiple attribute criteria model (MACM) in the context of the Hadoop ecosystem. The MACM includes parallel data loading using the customized utility. The HDFS saves the huge files in small chunks that are customized to avoid too much data and metadata, that would otherwise create the overhead.

In HDFS, a replication method is to replicate the original chunk of data which is a time-consuming task. As a result, customized replication is proposed in this paper. Moreover, the Sqoop utility is used due to the parallel loading of data using the map method. The proposed scheme utilizes Sqoop that offers connectivity to the external databases. The utilization of Sqoop brings a variety of features in SafeCity, such as loading with increments, complete import, parallel import, and corresponding export, compression, easy movement, enterprise independence, and auto-generation of tedious user side’s code. Data processing and analytics are carried out using the MapReduce programming paradigm. Hadoop divides input dataset into small blocks of same size files, known as input splits. The size of the split is usually identical to the block or chunk size. One specific task (known as map task) is formed for each split that performs the function of the map, defined by the programmer, for each row (a record). A RecordReader is used to arrange the rows as a pair (key-value). The MapReduce process is depicted in Figure 4. The outputs of the map are not stored in HDFS, these results are stored in the local storage. Results from a number of mappers are the input for the reduce task. Reduce tasks do not include the advantage of data locality characteristic. Therefore, the stored map results have to transfer crossway the system to that specific location, where the job of reducing is performing. The of the reducer result is stored on HDFS.

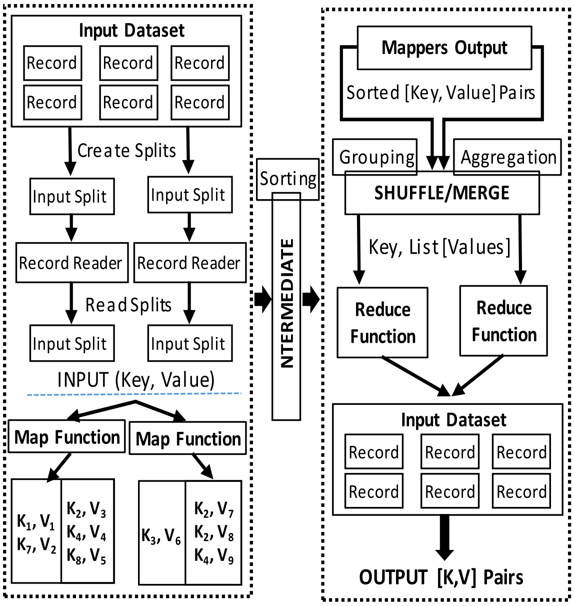

FIGURE 4. MapReduce paradigm.

Our projected scheme is grounded on the up-to-date depiction of Apache Hadoop framework which is embedded with Yet Another Resource Negotiator (YARN) and is accountable for data computation and cluster management. Unlike conventional MapReduce, the computation elements and resource management is separated by YARN. The YARN-enabled model is not limited to the MapReduce classical mechanism. The YARN is preferred due to limitations of classical MapReduce that are mostly associated with scalability and workload support. In the proposed architecture, YARN has a ResourceManager that runs as a master daemon by managing the accessible cluster resources among a wide range of competing and contending applications. The ResourceManager keeps track of the available resources and live nodes on the cluster. As it is the solo process having this information, so it coordinates the resource allocation and scheduling between the submitted applications. The allocation decisions are made in a secured, multi-tenant, and shared way, e.g. based on queuing capacity, data locality, an application priority, etc. On the submission of an application, a lightweight process instance, also known as ApplicationMaster, is initiated that is responsible for the execution of all the tasks within an application. It is comprised of tasks monitoring, restarting failed tasks, and calculating the overall values of the used application counters. In the existing literature, the classical MapReduce framework is utilized where a single JobTracker is responsible to take care of these responsibilities for all the jobs. Utilizing a single JobTracker in huge clusters exposes them to the scalability bottleneck.

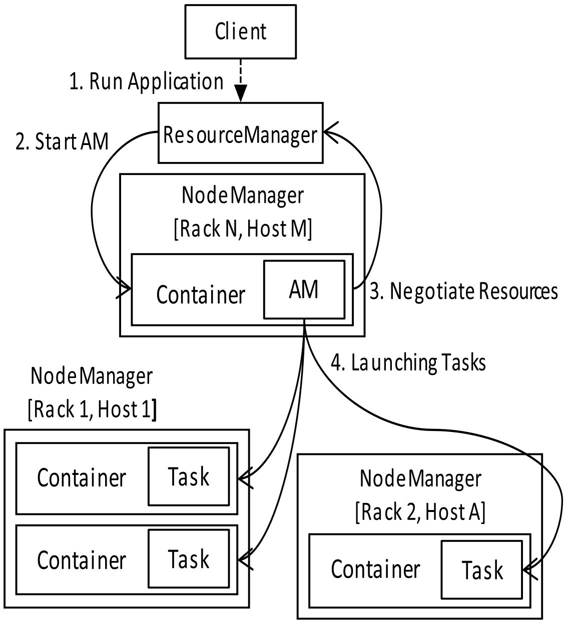

FIGURE 5. **Yet another resource negotiator (YARN).**

Different tasks associated with a particular application and an ApplicationMaster are controlled, monitored, and managed by the corresponding NodeManagers. Unlike the TaskTracker of a classical MapReduce framework, NodeManager is an efficient and more generic version of the TaskTracker. The NodeManager has many resource containers that are created dynamically, rather than having a defined number of slots (maps and reduces). All the components of the YARN such as ResourceManager, NodeManagers, ApplicationMaster, and containers cooperate with each other in a specific way upon the submission of an application in the cluster of YARN. This interaction of different parts of a YARN framework is shown in Figure 5.

The application is submitted using the Hadoop jar command in CLI or using Java IDE to RM, in a similar way to classical MR. A complete list of running jobs on the Hadoop cluster and all the available and accessible resources on every NM (live) are maintained by RM. The RM needs to decide which application is the next to acquire a piece of cluster resource. A number of constraints are taken into consideration while taking this decision such as fairness and capacity of the queue. The RM employs a scheduler that focuses mainly on scheduling activities. It deals with accessing the resources of a cluster and decides when and who will access them. Within an application, the task monitoring is not carried out by the scheduler and it never tries to restart a failed task. When the submission of a new application is accepted by ResourceManager, first the scheduler decides to select a container where ApplicationMaster will be started and run.

The ApplicationMaster will be in charge of the entire life cycle of the application when it starts. Primarily, ApplicationMaster would be requesting for various resources to the overall manager (ResourceManager) in order to inquire for different containers that are required to execute tasks of a particular application. A request for a particular resource is just a demand for several containers to assure various resource necessities, i.e., a number of resources. For example, CPU share, MB memory, preferred location, e.g. rack name, hostname or if no preference is required then $*$ is used, and priority inside the current application.

The ResourceManager grants a container, whenever possible, that satisfies the request made by an ApplicationMaster. On a specific host, the application is permitted by the container to utilize specified resources. ApplicationMaster requests the NodeManager to launch an application-specific task to utilize these resources after a container is granted.

Please recall that the NodeManager is responsible to manage the host on which a particular container is assigned. The application-specific task could be any particular task written in any framework, e.g. MapReduce. The NodeManager only monitors and examines the resource usage in the containers. It does not monitor the tasks and destroys them if they use more than the allocated memory.

The ApplicationMaster is responsible for monitoring the restarting tasks in fresh containers that are failed, the progress of tasks and its application, and provides the progress back to a client. The ApplicationMaster closes itself and releases its container on completion of the application. Nevertheless, the RM does not check the tasks inside an application at all. It only confirms the health of the ApplicationMasters. In this paper, a flowchart is proposed using the MapReduce programming paradigm that is applied to a water dataset. This flowchart is used to collect the values/quantity of water consumption against different houses to govern the level of water and its demand. The pictorial illustration of recommended MapReduce is depicted in Figure 6.

The mapper gets the offset of a line as a specific key and the entire row is considered as a value. The time parameter (timestamp) and associate values are produced as output

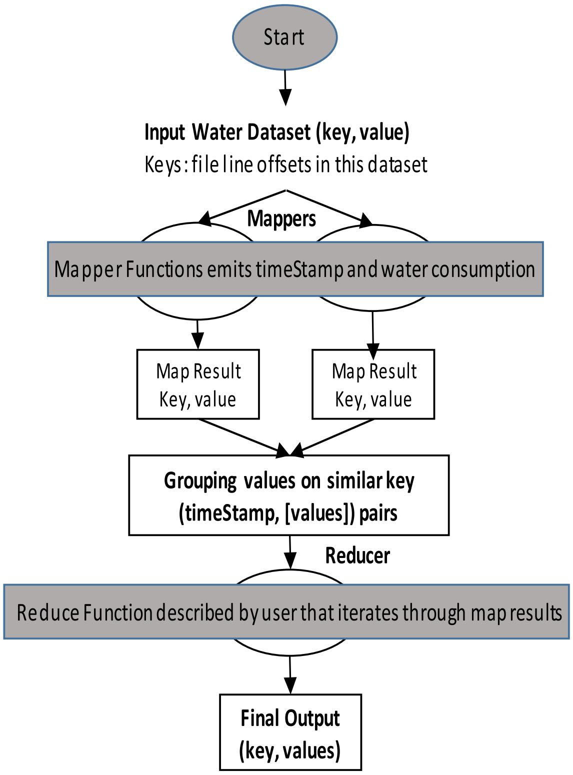
Start
Input Water Dataset (key, value)
Keys: file line offsets in this dataset
Mappers
Mapper Functions emits timeStamp and water consumption
Map Result
Key, value
Map Result
Key, value
Grouping values on similar key
(timeStamp, [values]) pairs
Reducer
Reduce Function described by user that iterates through map results
Final Output
(key, values)

FIGURE 6. MapReduce flowchart for water dataset.

Algorithm 1 Mapper for Water Dataset

```
BEGIN
    Input:
        key: line-offset
        value: = row
    Output:
        key: fecilityID
        value: LOTLINK
        //containing water consumption measurement

        // line splitting
    fecilityID, LOTLINK: = line.split ('\t')
    key: = fecilityID
    value: = LOTLINK
    emit (key, value)
END
```

by the mapper. The reducer clusters the necessary associate values alongside every timeStamp and relates with the TLV (threshold limit value). Information with regard to the water consumption of different houses is obtained with the help of such algorithms. As the MapReduce executes various jobs in 2 phases, i.e., Map phase and Reduce phase, therefore, a separate Map function and a Reduce function is proposed for the flowchart of Figure 6. In Algorithm 1, we present the mapper for the water dataset and in Algorithm 2, we present the reducer for the same dataset.

Algorithm 2 Reducer for Water Dataset

```
BEGIN
    Input:
        key: fecilityID
        value: LOTLINK
    Output:
        key: fecilityID
        value: LOTLINK greater than threshold
    initialize threshold
    final []
    FOR each (LOTLINK) at fecilityID DO
    IF (LOTLINK > threshold)
    Begin
            final.append (LOTLINK)
            key: = fecilityID
            value: = final
            emit (key, vaue)
        End IF
END
```

C. DECISION-MKING LAYER

The intelligent decision making is the key to our SafeCity framework that includes the prediction, creation of training sets, thresholds setting, rules definition, and event management. It acts as the moderator between the end-users and it is carried out by the decision-making agent, based on AI approaches. Various limits are defined and several rules are set for the assessment of different datasets. The processing of data is carried out using these rules according to proposed algorithms. The TLV (Threshold Limit Value) is a precise value set for each dataset also known as threshold or limit which is the base for event generation and decision making. Likewise, several rules are set centered on corresponding limits in the form of if/then statements that are utilized for decision making. The notification and event alert component determines the specific recipient of a generated event. Hence, it notifies the operator with the generated event for further actions.

IV. SYSTEM EVALUATION AND ANALYSIS

The detailed analysis and discussion of results achieved using SafeCity discuss in this segment. The secured data authentication is realized using Raspberry Pi boards for the client-server interface model. The Libcoap library is used for Raspbian operation system that provides basic communication among the ubiquitous devices. The analysis is carried out on a dataset that is realistic to evaluate the SafeCity scheme using the premeditated algorithms. The implementation of our ubiquitous data computation layer is carried out using the Hadoop cluster on Ubuntu OS along with Sqoop. Moreover, Java is used for the MapReduce implementation by utilizing the pre-defined classes (mapper and reducer). The data is received from diverse but trustworthy sources that are authentic. These datasets contain the transportation data, i.e., vehicles on roads in Aarhus city, Denmark. The water dataset homes are gained from the houses in Surrey, Canada.

A. SYSTEM EVALUATION FOR SAFETY AND SECURITY

The experimental results concerning the ubiquitous data security layer are illustrated here. A comparison of our payload-based authentication for SafeCity and CoAP-based DTLS implementation for smartphones is provided in Figure 7. DTLS+ denotes a smartphone (ubiquitous device) operating as a server and a workstation as a client. On the other hand, DTLS∗ denotes the handshake between a smartphone and a workstation, where the smartphone operates as a client and the workstation as a server. As the figure shows, SafeCity has a much lower handshake duration and standard deviation in comparison to DTLS∗ and DTLS+.

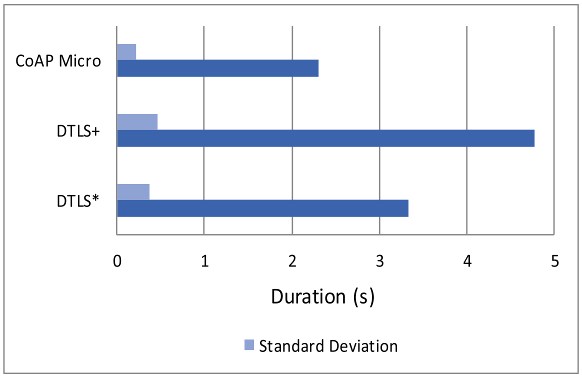

FIGURE 7. Handshake duration.

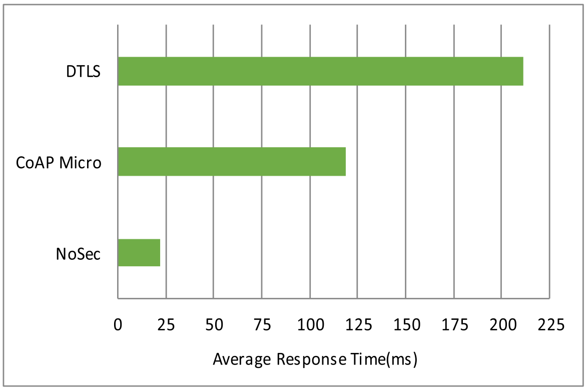

FIGURE 8. Average response time.

Similarly, SafeCity focuses on asynchronous communication of CoAP messages over the UDP sockets. A record of transferred Confirmable (CON) requests is maintained by every client. The mean reaction time for one CON request message of 1 byte is compared with DTLS exchange and the CoAP protocol with no added security, in Figure 8. SafeCity has a much lower average response time in comparison to DTLS because the latter involves computationally complex cipher suites and a resource-intensive record layer. CoAP with no added security has a slower response time but it is prone to various malicious and adversarial attacks.

TABLE 1. Average consumption (kb).

CoAP Micro	HTTP	HTTP/UDP	CoAPBlip	TinyCoAP
207	802	4009	7160	8498

The memory utilization of a CON request is evaluated at the compile time in Table 1. The proposed SafeCity is compared with the existing schemes for a CON message of minimum 500 bytes, as depicted in Figure 9 too. Among the current schemes, CoAPBlip [33] allocates considerable storage to messages at the compile time of the message. TinyCoAP [34] is a variation of the standard libraries of C that need the TinyOS element for its installation on a ubiquitous device. HTTP has a short foot-print of memory as it doesn't offer a trustworthiness method or correlation of a request/response. Both TinyCoAP and CoAPBlip use resource-consuming libraries and have a much higher memory consumption.

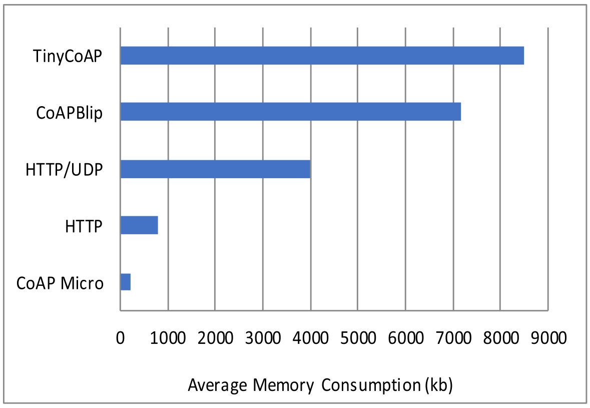

FIGURE 9. Average memory consumption.

B. SYSTEM EVALUATION FOR DATA PROCESSIGN AND COMPUTATION

Our SafeCity architecture generates alerts in real-time for a particular ubiquitous environment. In this section, we evaluate SafeCity in terms of efficiency by considering the execution time and throughput. To examine the system performance in real-time, various datasets, such as vehicular and water, are replayed to our Hadoop-based YARN framework of SafeCity. The throughput is assessed using datasets by increasing the data size. The efficiency concerning throughput is measured as shown in Figure 10. It can be observed that with the growth in size, the processing speed is reduced. The system throughput of Yarn-based framework is considerably higher in comparison to the existing classical MR-based solution.

Table 2 reveals the processing time, also known as the execution time proposed framework in the context of data volume. The execution time is evaluated for different sizes

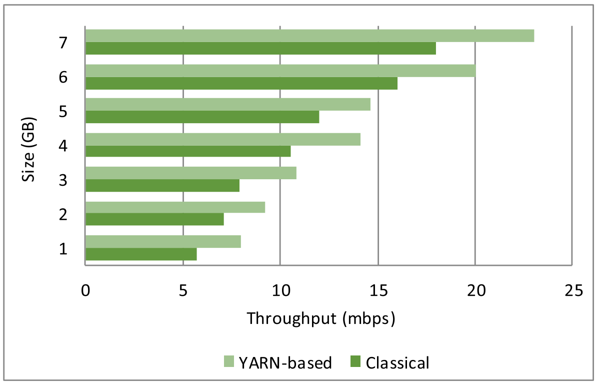

FIGURE 10. System throughput.

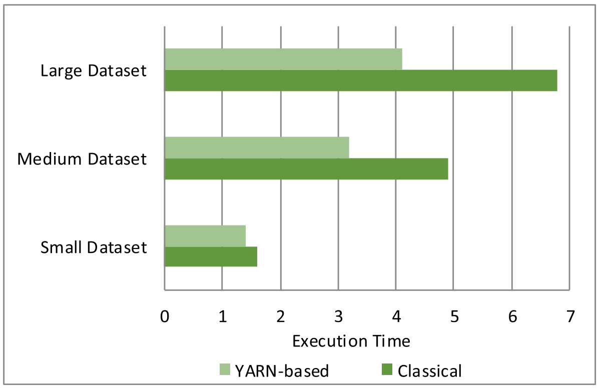

FIGURE 11. Execution time (s).

TABLE 2. Processing time of proposed framework.

Size (GB)	Time (ms)
1	67
2	78
3	96
4	119
5	133.5
6	150.9
7	168.3
8	185.7
9	203.1
10	220.5
11	237.9
12	255.3
13	270

TABLE 3. Average consumption (kb).

	Minor	Average	Huge
Traditional	1.4	4.9	6.8
Proposed	1.6	3.2	4.1

of data. The data size is started from 500MB and experienced up to 13 GB of data.

Table 3 determines the processing time in comparison to the classical structure. The time is calculated for minor, average, and huge datasets. It is observed that the processing time improves when the dataset size is increased. Figure 11 demonstrates the execution time of jobs using our Yarn-based framework in comparison to the existing scheme. The execution time is evaluated for small, medium, and large datasets. It is observed that the processing time improves when the dataset size is increased. It is mostly because of the data loading efficiency and improvement.

C. DATA ANALYSIS

The time difference of data loading is not perceptible when the size is smaller. The data ingestion time is pretty evident when the bulk of a dataset is larger due to the replication approach. The query that arises is the threshold data, to discover the TLV size, the data loading performance is measured by testing the different sizes of data.

The TLV size is the point where the time difference becomes positive (greater than 0) which means a significant change occurs. The TLVs for various attributes are set using the outputs of similar trials. Taking into account the data ingestion tool experiments, the TLV size is 900MB (size of data). At this value, the effect of the data ingestion period is experienced as shown in Figure 12. This figure demonstrates that 1GB of size does not generate any change even if the automated ingestion is practice. The productivity is attained when dataset size is greater than 900 MB at least.

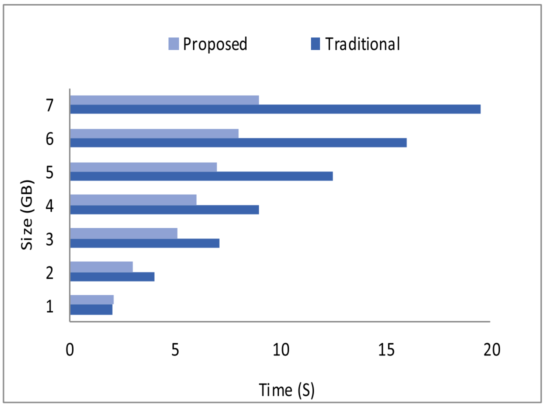

FIGURE 12. Data loading efficiency.

The water consumption is evaluated to achieve sustainable water management in the city due to the inconsistent consumption of water could be a disaster in the future. The data utilized in our research contains information about the city of Surrey, Canada. It comprises of the water intake of the houses in Surrey that is processed using our proposed algorithms. The results are demonstrated in Figure 13.

It shows the houses consumed more than 82000 liters each month. The defined TLV is 82000 found from the rule engine. The water usage higher than the TLV is particularly highlighted in this figure and this can cause frightening

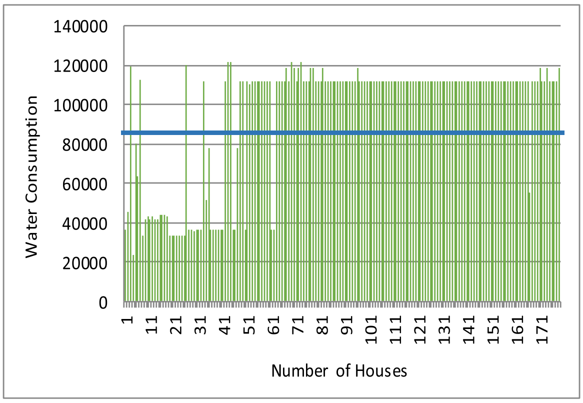

FIGURE 13. Water consumption.

situations for the authorities. It is observed that almost 50% of the consumers consumed more than the threshold limit. Most of the consumers, above the TLV limit, consumed water between 110000 to 120000 liter, which is quite alarming. Up-to-date fabrication methods could be industrialized to control the issues of the consumers in a city.

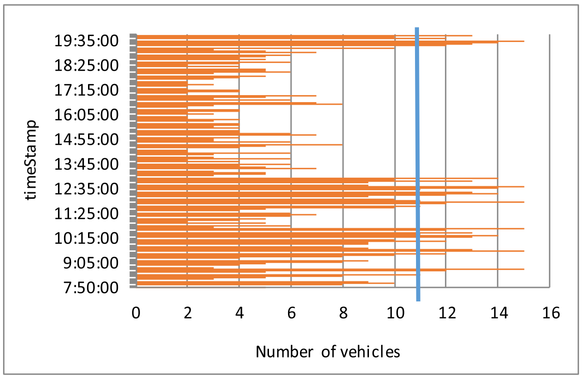

FIGURE 14. Number of vehicles on the road.

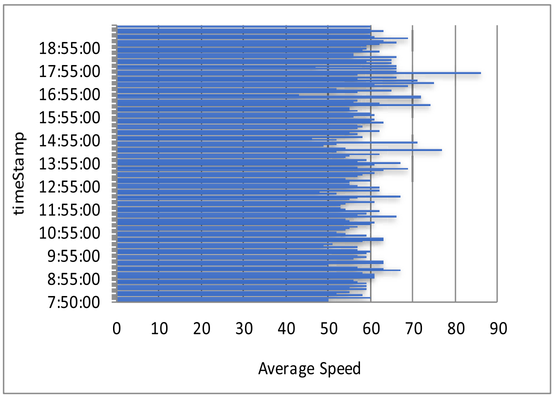

FIGURE 15. Average speed of automobiles.

Regarding traffic management, we consider the traffic data about road congestion. The data is intelligently processed using the SafeCity framework to overcome the traffic issues when the vehicles on roads surpass TLV. Figure 14 reveals the vehicles and the corresponding TLV. It depicts vehicles at a different time on the roads. It is observed that due to schooling hours, there are more cars between 8:05-12:15 PM due to school and office timing in the city.

Furthermore, the average speed of vehicles is revealed in Figure 15. It is noticed that the average speed of the vehicles is quite alike all day, except from 13:00 to 18:00, when there are few vehicles.

V. CONCLUSION

This paper has envisioned the vital role of safety and security in IoT-enabled data computation and communication to achieve safe and secure decisions. The data generated by IoT sensors exploit the association between various features of data and enables the meaning of a safe city. We have suggested the conception of SafeCity and proven its applicability using apache and Hadoop, via cautious investigation and assessment of the presence of residents in the evolving smart cities. SafeCity carefully controls the encounter of security and computation faced by the ubiquitous data. It is a layered architecture that is composed of a data security layer, data computation layer, and decision-making layer. A payload-based authentication approach is utilized at the ubiquitous data security layer to secure the ubiquitous data from malevolent entities.

The data computation layer is liable for the processing of secured data. Finally, the decision-making layer extracts insights for making smart decisions. The ubiquitous data security is evaluated using the Raspberry Pi boards while the ubiquitous data computation is tested on trustworthy datasets, using Hadoop. In association with the current methods, SafeCity is trivial about handshake duration, response time, and average memory consumption. Furthermore, it attains a lesser processing time, greater throughput, and efficient about massive data ingestion.

REFERENCES

[1] P. Bocquier, "World urbanization prospects: An alternative to the UN model of projection compatible with the mobility transition theory," *Demograph. Res.*, vol. 12, pp. 197–236, May 2005.

[2] J. L. Hernández, R. García, J. Schonowski, D. Atlan, G. Chanson, and T. Ruohomäki, "Interoperable open specifications framework for the implementation of standardized urban platforms," *Sensors*, vol. 20, no. 8, p. 2402, Apr. 2020.

[3] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of Things: A survey on enabling technologies, protocols, and applications," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 4, pp. 2347–2376, 4th Quart., 2015.

[4] M. Weber and I. P. Žarko, "A regulatory view on smart city services," *Sensors*, vol. 19, no. 2, p. 415, 2019.

[5] A. Entezami, H. Sarmadi, B. Behkamal, and S. Mariani, "Big data analytics and structural health monitoring: A statistical pattern recognition-based approach," *Sensors*, vol. 20, no. 8, p. 2328, Apr. 2020.

[6] M. Babar and F. Arif, "Smart urban planning using big data analytics based Internet of Things," in *Proc. ACM Int. Joint Conf. Pervas. Ubiquitous Comput., ACM Int. Symp. Wearable Comput.*, Sep. 2017, pp. 397–402.

[7] M. Babar and F. Arif, "Smart urban planning using big data analytics to contend with the interoperability in Internet of Things," *Future Gener. Comput. Syst.*, vol. 77, pp. 65–76, Dec. 2017.

[8] M. Babar and F. Arif, "Real-time data processing scheme using big data analytics in Internet of Things based smart transportation environment," *J. Ambient Intell. Hum. Comput.*, vol. 10, no. 10, pp. 4167–4177, Oct. 2019.

[9] M. Babar, A. Rahman, F. Arif, and G. Jeon, "Energy-harvesting based on Internet of Things and big data analytics for smart health monitoring," *Sustain. Comput., Informat. Syst.*, vol. 20, pp. 155–164, Dec. 2018.

[10] J. Granjal, E. Monteiro, and J. S. Silva, "Security for the Internet of Things: A survey of existing protocols and open research issues," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 3, pp. 1294–1312, 3rd Quart., 2015.

[11] S. Rajesh, V. Paul, V. Menon, and M. Khosravi, "A secure and efficient lightweight symmetric encryption scheme for transfer of text files between embedded IoT devices," *Symmetry*, vol. 11, no. 2, p. 293, Feb. 2019.

[12] M. A. Jan, F. Khan, M. Alam, and M. Usman, "A payload-based mutual authentication scheme for Internet of Things," *Future Gener. Comput. Syst.*, vol. 92, pp. 1028–1039, Mar. 2019.

[13] A. Venčkauskas, N. Morkevicius, V. Jukavičius, R. Damaševičius, J. Toldinas, and Š. Grigaliūnas, "An edge-fog secure self-authenticable data transfer protocol," *Sensors*, vol. 19, no. 16, p. 3612, Aug. 2019.

[14] S. L. Keoh, S. S. Kumar, and H. Tschofenig, "Securing the Internet of Things: A standardization perspective," *IEEE Internet Things J.*, vol. 1, no. 3, pp. 265–275, Jun. 2014.

[15] S. Raza, L. Seitz, D. Sitenkov, and G. Selander, "S3K: Scalable security with symmetric keys—DTLS key establishment for the Internet of Things," *IEEE Trans. Autom. Sci. Eng.*, vol. 13, no. 3, pp. 1270–1280, Jul. 2016.

[16] A. Bhattacharyya, A. Ukil, T. Bose, and A. Pal. *Lightweight Mutual Authentication for CoAP (WIP)*. Accessed: Mar. 3, 2014. [Online]. Available: https://draft-bhattacharyya-core-coap-lite-auth-00

[17] J. Granjal, E. Monteiro, and J. S. Silva, "On the feasibility of secure application-layer communications on the Web of things," in *Proc. 37th Annu. IEEE Conf. Local Comput. Netw.*, Oct. 2012, pp. 228–231.

[18] T. Kothmayr, C. Schmitt, W. Hu, M. Brünig, and G. Carle, "DTLS based security and two-way authentication for the Internet of Things," *Ad Hoc Netw.*, vol. 11, no. 8, pp. 2710–2723, Nov. 2013.

[19] J. Granjal, E. Monteiro, and J. S. Silva, "On the effectiveness of end-to-end security for Internet-integrated sensing applications," in *Proc. IEEE Int. Conf. Green Comput. Commun. (Green-Com)*, Nov. 2012, pp. 87–93.

[20] D. Trabalza, S. Raza, and T. Voigt, "Indigo: Secure coap for smartphones," in *Wireless Sensor Networks for Developing Countries*. Berlin, Germany: Springer, 2013, pp. 108–119.

[21] M. A. Jan, P. Nanda, X. He, Z. Tan, and R. P. Liu, "A robust authentication scheme for observing resources in the Internet of Things environment," in *Proc. IEEE 13th Int. Conf. Trust, Secur. Privacy Comput. Commun.*, Sep. 2014, pp. 205–211.

[22] S. Din, H. Ghayvat, A. Paul, A. Ahmad, M. M. Rathore, and I. Shafi, "An architecture to analyze big data in the Internet of Things," in *Proc. 9th Int. Conf. Sens. Technol. (ICST)*, Dec. 2015, pp. 677–682.

[23] M. M. Rathore, A. Ahmad, A. Paul, and S. Rho, "Urban planning and building smart cities based on the Internet of Things using big data analytics," *Comput. Netw.*, vol. 101, pp. 63–80, Jun. 2016.

[24] M. M. Rathore, A. Paul, A. Ahmad, M. Anisetti, and G. Jeon, "Hadoop-based intelligent care system (HICS): Analytical approach for big data in IoT," *ACM Trans. Internet Technol.*, vol. 18, no. 1, p. 8, Dec. 2017.

[25] B. N. Silva, M. Khan, C. Jung, J. Seo, Y. Yoon, J. Kim, S. Jin, J. Kang, and K. Han, "Planning of smart cities: Performance improvement using big data analytics approach," in *Proc. 4th Int. Conf. Adv. Comput., Electron. Commun. Inst. Res. Eng. Doctors*, 2016, pp. 51–55.

[26] Y. Sun, H. Song, A. J. Jara, and R. Bie, "Internet of Things and big data analytics for smart and connected communities," *IEEE Access*, vol. 4, pp. 766–773, 2016.

[27] R. Tönjes, M. I. Ali, P. Barnaghi, S. Ganea, F. Ganz, M. Haushwirth, B. Kjærgaard, D. Kümper, A. Mileo, S. Nechifor, A. Sheth, V. Tsiatsis, and L. Vestergaard, "Real time iot stream processing and large-scale data analytics for smart city applications," in *Proc. Eur. Conf. Netw. Commun.*, 2014, pp. 1–5.

[28] B. Cheng, S. Longo, F. Cirillo, M. Bauer, and E. Kovacs, "Building a big data platform for smart cities: Experience and lessons from santander," in *Proc. IEEE Int. Congr. Big Data*, Jun. 2015, pp. 592–599.

[29] M. M. Rathore, A. Paul, A. Ahmad, and G. Jeon, "IoT-based big data: From smart city towards next generation super city planning," *Int. J. Semantic Web Inf. Syst.*, vol. 13, no. 1, pp. 28–47, 2017.

[30] V. G. Menon and J. Prathap, "Vehicular fog computing: Challenges applications and future directions," *Int. J. Veh. Telematics Inf. Syst.*, vol. 1, no. 2, pp. 15–23, 2017.

[31] M. M. Rathore, A. Ahmad, A. Paul, and G. Jeon, "Efficient graph-oriented smart transportation using Internet of Things generated big data," in *Proc. 11th Int. Conf. Signal-Image Technol. Internet-Based Syst. (SITIS)*, Nov. 2015, pp. 512–519.

[32] Z. Shelby, K. Hartke, and C. Bormann, *The Constrained Application Protocol (CoAP)*, document RFC 7252, 2014.

[33] K. Kuladinithi, O. Bergmann, T. Pötsch, M. Becker, and C. Görg, "Implementation of coap and its application in transport logistics," in *Proc. IP+SN*, Chicago, IL, USA, 2011, pp. 1–6.

[34] A. Ludovici, P. Moreno, and A. Calveras, "TinyCoAP: A novel constrained application protocol (CoAP) implementation for embedding RESTful Web services in wireless sensor networks based on TinyOS," *J. Sens. Actuator Netw.*, vol. 2, no. 2, pp. 288–315, May 2013.

HUI ZHANG received the B.Sc. degree in communication engineering from Henan Polytechnic University, China, in 2007, the M.Sc. degree from the School of Energy Science and Engineering, Henan Polytechnic University, in 2010, and the Ph.D. degree from the School of Energy and Mining Engineering, China University of Mining and Technology, in 2013. He is currently an Associate Professor with the School of Energy Science and Engineering, Henan Polytechnic University. He has authored several articles in journal and conferences, and holds several patents. His research interests include mining communication and smart mine.

MUHAMMAD BABAR received the bachelor's degree (Hons.) in computer sciences from the University of Peshawar, Pakistan, in 2008, and the Master of Science and Ph.D. degrees in computer software engineering from National University of Sciences and Technology (NUST), Islamabad, Pakistan, in 2012. He is currently with Iqra University, Islamabad. He has published his research work in various IEEE and ACM/Springer international conferences and journals. His research interests include big data analytics, the Internet of Things (IoT), smart city design and planning, and Social Web of Things (SWOT). He is an active Reviewer and a guest editor in the reputed journals.

MUHAMMAD USMAN TARIQ received the bachelor's and Master of Science degrees in computing, with a specialization in software engineering, and the Ph.D. degree (Hons.) in management from Calsouthern, USA. He has more than 13 years' experience in industry and academia. He has a passion for learning and development, project management, and training that made him achieve four patents. His research interests include management, the IoT, six sigma, knowledge management, information technology, economics, organizational change, facial recognition, biomedical devices, and computer science.

MIAN AHMAD JAN received the Ph.D. degree in computer systems from University of Technology Sydney (UTS), Australia, in 2016. He is currently a Researcher with Ton Duc Thang University, Vietnam. His research has been published in various prestigious the IEEE TRANSACTIONS and Elsevier Journals. His research interests include security and privacy in the Internet of Things, and wireless sensor networks. He was a recipient of various prestigious scholarship during his studies, notably the International Research Scholarship (IRS) at the UTS, and the Commonwealth Scientific Industrial Research Organization (CSIRO) scholarships. He has been received the Best Researcher awarded for the year 2014 at the UTS, Australia. He has been the general Co-Chair of Springer/EAI 2nd International Conference on Future Intelligent Vehicular Technologies, in 2017. He has been a guest editor of numerous special issues in various prestigious journals, such as the IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, *Future Generation Computer Systems* (Elsevier), *Mobile Networks and Applications* (MONET) (Springer), *Ad Hoc & Sensor Wireless Networks*, and *MDPI Information*.

VARUN G. MENON (Senior Member, IEEE) is currently an Associate Professor with the Department of Computer Science and Engineering, SCMS School of Engineering and Technology, India. His research interests include the Internet of Things, fog computing and networking, underwater acoustic sensor networks, cyber psychology, hijacked journals, ad-hoc networks, and wireless sensor networks. He is a Distinguished Speaker of ACM Distinguished Speaker. He is currently a Guest Editor of the IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, the IEEE SENSORS JOURNAL, the *IEEE Internet of Things Magazine*, and the *Journal of Supercomputing*. He is an Associate Editor of *IET Quantum Communications*. He is also an Editorial Board Member of the IEEE Future Directions: Technology Policy and Ethics.

XINGWANG LI (Senior Member, IEEE) received the B.Sc. degree from Henan Polytechnic University, Jiaozuo, China, in 2007, the M.Sc. degree from the University of Electronic Science and Technology of China, in 2010, and the Ph.D. degree from the Beijing University of Posts and Telecommunications, in 2015.

From 2010 to 2012, he was working as an Engineer with Comba Telecom Ltd., Guangzhou, China. From 2016 to 2018, he was also a Visiting Scholar with the State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications. From 2017 to 2018, he was a Visiting Scholar with Queen's University Belfast, Belfast, U.K. He is currently an Associate Professor with the School of Physics and Electronic Information Engineering, Henan Polytechnic University. His research interests include MIMO communication, cooperative communication, hardware constrained communication, non-orthogonal multiple access, physical layer security, unmanned aerial vehicles, and the Internet-of-Things. He has served as many TPC members, such as the IEEE/CIC International Conference on Communications in China (ICCC'2019) and the IEEE Global Communications Conference 2018 (Globecom'18). He is also an Editor on the Editorial Board of IEEE ACCESS, *Computer Communications*, and *KSII Transactions on Internet and Information Systems*. He is also the Lead Guest Editor for the Special Issue on Recent Advances in Physical Layer Technologies for the 5G-Enabled Internet of Things of Wireless Communications and Mobile Computing and the Lead Guest Editor for the Special Issue on Recent Advances in Multiple Access for 5G-enabled IoT.