Dar, Zaineb and Ahmad, Adnan and Khan, Farrukh Aslam and Zeshan, Furkh and Iqbal, Razi and Sherazi, Hafiz Husnain Raza and Bashir, Ali Kashif (2019) A context-aware encryption protocol suite for edge computing-based IoT devices. The Journal of Supercomputing. ISSN 0920-8542

Publisher: Springer Science and Business Media LLC

DOI: https://doi.org/10.1007/s11227-019-03021-2

A Context-Aware Encryption Protocol Suite for Edge Computing based IoT Devices

Zaineb Dar[1], Adnan Ahmad[1, 2], Farrukh Aslam Khan[3], Furkh Zeshan[1], Razi Iqbal[4], Hafiz Husnain Raza Sherazi[5], and Ali Kashif Bashir[6]

[1] COMSATS University Islamabad, Lahore Campus, Pakistan
[2] Saint Louis University, Madrid, Spain
[3] Center of Excellence in Information Assurance (CoEIA), King Saud University, Saudi Arabia.
[4]American University in the Emirates, United Arab Emirates
[5]Department of Electrical and Information Engineering, Politecnico di Bari, Italy
[6]Department of Computing and Mathematics, Manchester Metropolitan University, United Kingdom.

Corresponding authors:
Hafiz Husnain Raza Sherazi (e-mail: sherazi@poliba.it);
Adnan Ahmad (e-mail: adnanahmad@cuilahore.edu.pk)

ABSTRACT Heterogeneous devices are connected with each other through wireless links within a Cyber Physical System. These devices undergo resource constraints such as battery, bandwidth, memory, and computing power. Moreover, the massive interconnections of these devices result in network latency and reduced speed. Edge computing offers a solution to this problem in which devices transmit the preprocessed actionable data in a formal way resulting in reduced data traffic and improved speed. However, to provide the same level of security to each piece of information is not feasible due to limited resources. In addition, not all the data generated by Internet of Things (IoT) devices require a high level of security. Context-awareness principles can be employed to select an optimal algorithm based on device specifications and required information confidentiality level. For context-awareness, it is essential to consider the dynamic requirements of data confidentiality as well as device available resources. This paper presents a context-aware encryption protocol suite that selects optimal encryption algorithm according to device specifications and the level of data confidentiality. The results presented herein clearly exhibit that the devices were able to save 79% memory consumption, 56% battery consumption and 68% execution time by employing the proposed context-aware encryption protocol suite.

INDEX TERMS Edge Computing, Cyber Physical Systems, 6LoWPAN, Context-awareness, Device Specification, Information Profiling, IoT

I. INTRODUCTION

Internet of Things (IoT) allows different devices to connect with each other through a wireless network as a part of Cyber Physical Systems (CPS) [1]. These interconnected devices are designed in such a way that they share information about their surroundings to respond to real-world events. The key features of IoT include self-configuration, smart decision making, environment sensing, event triggering, ad-hoc networking, autonomously reacting and action controlling [2]. The number of IoT devices is increasing exponentially and is estimated to reach 50 billion till 2020 [3]. For the deployment over large areas and in numerous quantities, these devices need to be highly affordable. These affordability factors give us devices with limited capabilities in terms of battery, computation power, size, and storage [4, 5]. Due to these constraints, IoT devices need a communication protocol that can work with short radio range and consume fewer resources. 6LoWPAN [5, 6] provides a promising solution to this problem with the help of edge computing by adding an adaption layer in the network protocol stack for integrating low-power network such as IEEE 802.15.4 into IPv6 [7, 8]. This adaption layer solves two major problems including a massive number of IoT devices through IPv6, and their resource constraint nature through utilizing less resouces. Moreover, with the increase of IoT devices on the network, data production ratio also increases resulting in network latency and decreased speed. In this regard, edge computing is one of the available approaches [9] in which data is collected and aggregated on the edge node. This requires securing the work nodes and edge nodes, as well as the data transferred among them [10]. Such preprocessing reduces network latency and improves speed. However, providing security (device and data) to such a large infrastructure is challenging as security requirements vary from device to device and data to data. Therefore, the encryption standards in edge computing needs to be context-aware and adjustable according to the requirements of the device as well as the data. .

The adaption layer of 6LoWPAN generally includes one encryption standard for providing security to different types of data and devices [6]. In 6LoWPAN, message encryption is usually performed through traditional cryptographic algorithms such as AES, DES, and RC4, etc. [5]. These algorithms require more computational power, time and memory that cause an overhead for lightweight IoT devices [7]. Due to these constraints, the adaption of an encryption mechanism according to the capability of the resource is an open challenge [8] [11]. To make matters worse, IoT devices use one encryption algorithm for all types of sensitive as well as non-sensitive information [6] [12]. On one hand, it is inappropriate for a resource-constrained device to provide the same security level to sensitive and non-sensitive information. While on the other hand, any modification in this protocol may result in either heavy resource utilization or compromized security. Therefore, it seems unavoidable, for long, to implement a context-aware encryption mechanism in 6LoWPAN to provide encryption to the data according to its security demands as well as device capabilities [13, 14]. The mechanism should use encryption algorithm according to the confidentiality level of the information as well as the capabilities of the IoT device.

To address this challenge, this paper proposes a Context-aware Encryption Protocol Suite (CEPS) for 6LoWPAN. CEPS is an optimal encryption mechanism that considers device capabilities and information sensitivity level. To achieve this goal, various encryption algorithms were implemented as part of the protocol suite varying in strength as well as computational requirements. The proposed context-aware encryption suite extracts the device capabilities such as battery and memory using kinetic battery model [15] and memory usage probe [16], respectively. It also classifies the information confidentiality requirements through fuzzy logic [34] and extracts the resource utilization and security strength of each encryption algorithm through simulation. It then maps the resource current status and information requirements to select an appropriate encryption algorithm optimal for both device available resources and information requirements. Experimental results showed that CEPS was able to save 79% memory consumption, 56% battery consumption and 68% execution time compared to current existing solutions. This paper extends the literature in the following three ways: a) context-awareness in encryption protocols; b) information classification based on its confidentiality for 6LoWPAN; c) mapping of encryption algorithms based on resource availability and information requirements.

The remainder of this paper is organized as follows. Section 2 presents an overview of related work. In Section 3, the proposed context-aware encryption protocol suite for 6LoWPAN is discussed. Simulation of the context-aware encryption protocol suite and its comparison with the existing techniques is explained in Section 4 while, Section 5 provides some concluding remarks.

II. RELATED WORK

IoT devices are not only different from each other with respect to their resources such as memory, CPU, and battery [4], but also with respect to the confidentiality levels of the information they generate [18]. Thus, to provide the same security level to every piece of information seems less efficient, as the confidentiality requirement of information and the capabilities of devices may vary. Context-awareness often provides solution to similar situations by utilizing the context to provide relevant services to the user [19]. In IoT, context-awareness is used to solve many problems in various situations including smart homes [20], smart grids [21], agriculture [22], health care [23], and automated logistics [24].

In recent times, context-awareness has also been used in IoT paradigm to provide some security solutions. The literature review for this research can be broadly categorized into four different categories including a) device profiling, b) information profiling, c) security protocol suite in IoT, and d) security in 6LoWPAN. Following is a brief overview of the research work in these areas:

A. DEVICE PROFILING

For a resource-constrained device, it is less desirable to adopt an expensive encryption technique for different pieces of information varying in security requirements, without considering its resources [25]. A context-aware system supports the acquisition, representation, delivery, and reaction [15]. Device profiling in context-awareness is to match user demands over a device by asking questions, for example, does the device has enough memory to perform this task? Does it has ample power to execute this task? For an encryption protocol suite, it is essential to find a security cipher according to the device capabilities. In the past, various architectures for IoT devices have used device profiling to perform certain tasks. A brief overview of some of these research works is as below:

Messer et al. proposed an approach for small-memory devices to execute full version of an application by offloading program's portions to a service [26]. Their system enhances the device's capability so if a device has not sufficient resources at runtime, it finds a nearby trusted computing node for offloading. However, they only considered static decisions for managing memory constraints but did not consider other resources such as network, transmission, and power. In addition, Hofer et al.

proposed a context-aware software framework to support network connections and limited computing power [27]. However, the framework allowed sharing of context among devices without assessing their security and reliability. Further, a content adaptation system for heterogeneous mobile devices was proposed by Lum et al. [28], to provide rich hypermedia according to the device capabilities. The system was implemented for optimal adaption based on various QoS attributes like device battery level, bandwidth, screen size, and network conditions. However, while the system focused on basic objects like text and images, the overall content adaption approach appeared ad hoc. Likewise, Taneja et al. [29] proposed a module mapping algorithm to minimize the IoT application latency and energy consumption. Their results provided a benchmark for IoT computation and can be used to provide QoS for various applications. However, only static network topologies were evaluated but no dynamic wireless constraints were considered. In addition, Sathyamoorthy et al. proposed a power management solution for resource constrained IoT devices [30]. Their proposed approach predicted the behavior of IoT devices by extracting the application characteristics and calculating their resource utilization. However, data confidentiality was not considered at all in their profiling, power management or analysis of data logs.

Different studies in the literature proposed interesting architectures based on the device profiling, however, the authors were unable to find any research that could suggest suitable encryption protocol according to the device profiling.

B. INFORMATION PROFILING

Massive amount of data is generated by IoT devices which vary in level of sensitivity. For a resource-constrained device, it is less feasible to use an expensive encryption operation for each piece of information regardless of its confidentiality requirements [17, 31]. Selecting an encryption technique without understanding the confidentiality level of data is not a valid technical approach [32]. Information profiling [33] can categorize the data into different classes according to its confidentiality requirements, so appropriate decisions can be observed about the optimal encryption algorithm. A brief overview of various research works present in this domain is as below:

Zardari et al. proposed an enhanced version of the KNN algorithm to improve the efficiency and accuracy of data classification into confidential and non-confidential classes [32]. Their proposed algorithm uses a subset of training file for classification contrary to the utilization of entire file in KNN. However, their algorithm did not classify data in the IoT paradigm neither it was evaluated on the execution time or resource utilization. Further, Mohammadian et al. proposed a data classification model based on the organization's privacy policies and government rules [34]. Their model used fuzzy logic to classify different attributes of the data about an organization. Primarily, the model was suggested to be used in financial organization, where data can be categorized into "very high", "high", "medium" or "low" confidentiality levels. The main drawback of this approach was users defining attributes' weights which are prone to biases, errors, and complex for large dataset. In addition, a model for IoT data classification is proposed in [33], which also encrypts the data if found sensitive. The KNN algorithm was used for classifying the data while the RSA algorithm is used if the classified data is found sensitive. The proposed approach reduced the computation of the system by not encrypting non-confidential data and hence enhanced the efficiency of the system. However, the proposed system was not context-aware and also not evaluated on network or resource utilization.

The above-mentioned proposals classify data according to its confidentiality level yet do not provide context-aware classification in IoT paradigm. Also, they were not evaluated on their computational cost or device capabilities including memory, bandwidth, network latency, and power.

C. SECURITY PROTOCOL SUITE IN IOT

IoT is a network of sensor devices which are connected through wireless network and technology to achieve overall perception of information, reliable transmission, and intelligent processing [16]. Hence, protecting privacy and security are the essential features of IoT [35]. In the recent past, various encryption mechanisms have been proposed for different IoT protocols to ensure less power consumption, memory, and network latency. A brief overview of these mechanisms is provided below:

Adrianto et al. performed a comparison among various security protocols, implemented in ETSI M2M standards, to find the most suitable algorithm for applications generating bulk of data [16]. Their selection criteria were message size, CPU usage, memory utilization and processing time. However, they did not consider battery consumption and information confidentiality level in algorithm selection. Likewise, Wu et al. [36] presented a lightweight security protocol suite for IoT which includes lightweight encryption, authentication, and key management. The security of the information is ensured by using a random single key for separate file encryption. The protocol suite was proved to be computationally efficient but does not utilize context-awareness. Further, Hamad et al. [37] identified computational requirements of renowned encryption algorithms in the cloud paradigm. Their results on power consumption and computation time were used to take decisions while selecting protocols in cloud environment. However, they did not provide any information about the computational requirements of encryption protocols. Glissa et. al [38] introduced a new security module for Omnet++ that implements the security suite for IEEE 802.15.4. The security sub-layer is responsible for data encryption and authentication according to the desired security degree. Their experimental setup addressed four performance aspects including energy consumption, transmission latency, packet delivery ratio and memory overhead. However, the security module does not provide context-awareness according to the information confidentiality level. In the similar vein, Toldinas et al. [39] performed an empirical study to evaluate the energy consumption of symmetric and asymmetric cryptography algorithms through Bouncy Castle Crypto API. They evaluated computing resources such as CPU time and memory for the reduction of power utilization by cryptography algorithms. However, both of these works do not provide context-awareness according to the information confidentiality level.

From the literature review, it is concluded that little attention has been given to context-aware encryption of IoT data with respect to device profiling and information profiling.

D. ENCRYPTION IN 6LOWPAN

While encryption mechanisms are implemented for resource-constrained sensors and actuator networks, in 6LoWPAN they experience poor performance due to the size of packets exchanged and the length of the keys [40]. Various initiatives have been taken to enhance the performance of encryption mechanisms for resource-constrained devices, e.g., TinyECC [41] and NanoECC [42]. Such initiatives improved the encryption efficiency but do not provide context-aware solution for IoT, which includes the resource availability level as well as information sensitivity requirements. Following is a brief description of the research work present for encryption in 6LoWPAN.

A performance study of end-to-end security available for 6LoWPAN based networks is presented by Matthias et al. [43]. The performance analysis covered battery consumption, network latency and memory utilization, while did not measure the impact of information or device classification on the resources. Likewise, Raza et al. [44] explored current protocols and security solutions that can be deployed in a constrained environment. They discussed security requirement as well as security mechanisms implemented at each layer of 6LoWPAN protocol stack and addressed various challenges and limitations for a pragmatic deployment in a physical environment. However, they did not cover resource consumption for different security mechanisms. Moreover, Jung et al. [45] proposed a lightweight Secure Sockets Layer (SSL) for IP-WSN security using ECC instead of RSA for key exchange and authentication. Their results showed some improvements in resource consumption, however, they did not utilize context-awareness for device capabilities or information confidentiality level.

The literature in this domain provides some lightweight encryption techniques for resource constraint devices but it does not provide dynamic selection of algorithms based on the context of devices. Although the above-mentioned approaches provided in these four domains, addressed many limitations of the tradition encryption mechanisms, there are still various issues that can be addressed by a context-aware protocol suite for resource-constrained devices. Different architectures based on device profiling have been proposed, but they do not cover various aspects of constraint devices including memory utilization, power, bandwidth, latency, and dynamic connections. Different encryption protocols have been designed for IoT devices, yet they do not provide context-awareness according to the device available resources and information confidentiality level. This research now proposes a context-aware encryption protocol suite (CEPS) for 6LoWPAN utilizing device, information as well as encryption algorithm profiling.

III. PROPOSED SOLUTION

IoT configurations are categorized into two classes, i.e., sensor-level configuration and system-level configuration [18]. Sensor-level configurations deal with the configuration of embedded software for changing behavior of sensors such as sensing schedule, communication patterns, sampling rate, data communication and protocols [18], while system-level configurations deal with the configuration of internal software components for changing behavior of IoT middleware systems [18]. Our proposed model identifies and configures both sensors and system processing components in order to select suitable encryption algorithm. Figure 1 shows the proposed encryption protocol suite, which has four main components namely: a) device profiling, b) information profiling, c) algorithm profiling and d) mapping. CEPS first extracts the device capabilities through kinetic battery model [15] and memory usage probe [16]. It then classifies the data according to its confidentiality level through fuzzy logic and extracts the resource utilization and security strength of each encryption algorithm through simulation. CEPS then maps the security requirements of the information over the algorithm's security strength and resource utilization of algorithms over the device capabilities to get the suitable security cipher. Detail of these independent components is provided below.

A. DEVICE PROFILING

IoT devices have limited capabilities in terms of battery and memory. For a resource-constrained device, it may not be feasible to adopt an expensive encryption technique every time without considering its resources. CEPS utilizes device profiling to extract its available memory and battery, so appropriate decisions can be taken later for the selection of encryption algorithm. Following steps are performed for device profiling:

Step 1: Implement 6LoWPAN protocol stack:

IoT nodes adopt optimized functionalities of protocols which lead toward periodic sleep-wake cycles for preserving the resources. A Low Powered Personal Area Network (LoWPAN) is composed of small heterogeneous devices with limited resources capabilities in terms of energy, throughput, memory, and computation. For these

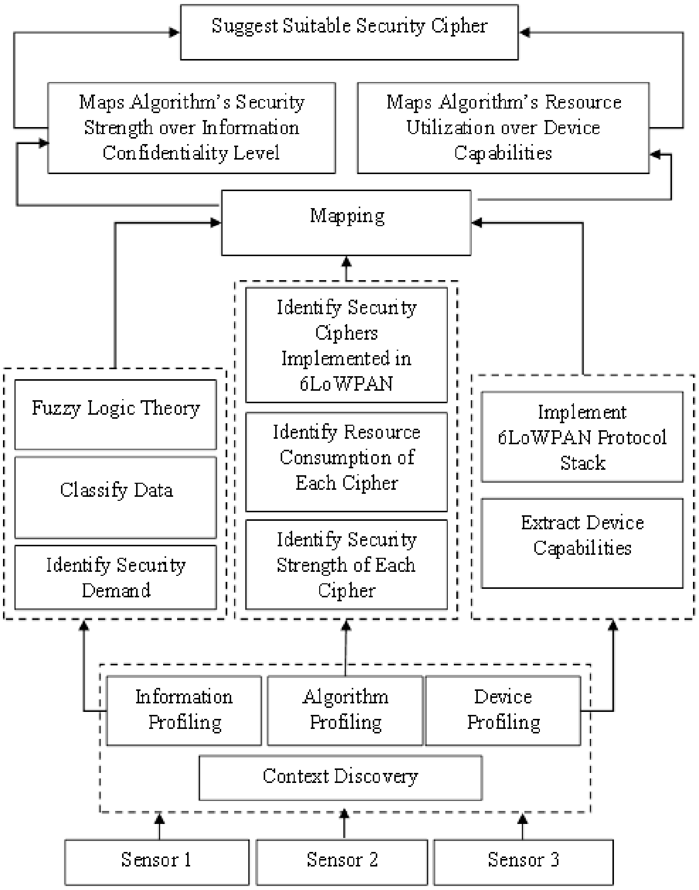
Suggest Suitable Security Cipher
Maps Algorithm's Security Strength over Information Confidentiality Level
Maps Algorithm's Resource Utilization over Device Capabilities
Mapping
Fuzzy Logic Theory
Classify Data
Identify Security Demand
Identify Security Ciphers Implemented in 6LoWPAN
Identify Resource Consumption of Each Cipher
Identify Security Strength of Each Cipher
Implement 6LoWPAN Protocol Stack
Extract Device Capabilities
Information Profiling
Algorithm Profiling
Device Profiling
Context Discovery
Sensor 1
Sensor 2
Sensor 3

FIGURE 1. Proposed solution for a context-aware protocol suite

wireless networks, the internet protocol (IP) should run on low cost, low bandwidth and low power device over IEEE 802.15.4. IPv6 over Low Powered Personal Area Network (6LoWPAN) acts as an adaption layer between IPv6 and IEEE 802.15.4 networks. This protocol was designed for small sensor devices which cannot afford expensive internet protocol. These devices transmit messages to other devices as well as receive messages over 6LoWPAN. 6LoWPAN provides an adaption layer with a code size of 12KB, requires only 4KB RAM, produces an overhead of only 2 to 11 bytes and supports 802.15.4++ as well as UDP/TCP.

Step 2: Extract device capabilities

To select different algorithms based on the device context, its available battery and memory capacities are extracted.

a) Battery: For the powertrace, kinetic battery model [15] (KiBaM) is used to calculate the state of charge of a battery. The energy-harvesting module of KiBaM reads a data-trace that contains the amount of harvested energy per minute. This module takes the data-trace as input, processes it and feeds the model with the equivalent charging current [15]. Variable fixed_perc_energy provides the percentage of energy as the node starts.
b) Memory: To measure the memory (RAM) of an IoT device, memory usage probe was used through the /proc/self/statm [16] virtual file. This file provides information about the physical memory of a system by reading RSS from the statm file.

The algorithm for device profiling can be observed in Algorithm 1.

B. INFORMATION PROFILING

A huge amount of data is generated by IoT devices which may include general as well as sensitive information. For a real-time system in which rapid processing of data is required, using an encryption technique without understanding the confidentiality level of the data is not always a desirable approach. Therefore, it is necessary to analyze the security level of data before applying a data encryption technique. Fuzzy logic approach [34] was used in the proposed context-aware encryption protocol suite for data classification. The fuzzy set theory provides the facility to develop rule-based models which include expert knowledge along with the numerical data close to real-world scenarios. Fuzzy approaches treating uncertainties in real-world applications have several advantages as they are conceptually easy to understand, flexible, tolerant to imprecise data and can model non-linear functions of arbitrary complexity. Fuzzy logic can be implemented in systems with various sizes and capabilities ranging from

ALGORITHM 1: DEVICE PROFILING

```
Input:
Implement 6LoWPAN network topology in nodes:
    create IPv6 Internet stack
    set MTU, delay, data rate of the channel
    create csma
    install 6lowpan
    assign ipv6 address
Device capabilities:
    battery:
        Apply KibaM battery model
        read the fixed_perv_energy variable to get the available battery
    Memory:
        open proc/self/statm file
        read rss of the statm file to get the memory of the system
```

small micro-controllers to large, networked, workstation-based control systems. In CEPS, data is classified in three confidentiality classes (i.e., high, medium and low). Following steps are performed for information profiling:

Step 1: Define linguistic variables and terms

Confidentiality level is the linguistic variable representing confidentiality level of the information generated by an IoT device. The linguistic values of the confidentiality are High, Medium or Low.

Step 2: Construct membership functions for linguistic variables

A membership function is used to quantify a linguistic term. Three membership functions are developed to classify different linguistic variables. For instance, if the data generated by the IoT device is related to the user, then its confidentiality level is considered as High; if the data is related to the environment or particular task then its confidentiality level is considered as Medium; while if the data is related to the general information including routing information then its confidentiality level is considered as "Low".

ALGORITHM 2: INFORMATION PROFILING

```
Input:
Define linguistic variables and terms (High, Medium, Low)
Construct membership functions for linguistic variables
    IF (DATA_Generated_category= "User",
        then Data_confidentiality_level= "high")
    Else If (DATA_Generated_category = "enviorment" ,
        then Data_confidentiality_level = "Medium")
    Else If (DATA_Generated_category= "public information",
        then Data_confidentiality_level = "Low")
Construct knowledge base rules
    IF DATA_Confidentiality_level = "high"
        THEN Encryption_requirement= "High"
    IF DATA_Confidentiality_level = "Medium"
        THEN Encryption_requirement = "Medium"
    IF DATA_Confidentiality_level = "Low"
        THEN Encryption_requirement = "Low"
```

Step 3: Construct knowledge base rules
A fuzzy rule is a simple IF-THEN rule with a condition and a conclusion, which is constructed to control the output variable. Following fuzzy rules were defined for the linguistic variables and member functions:

IF Data_Confidentiality_level = "High" *THEN* Encryption_requirement= "High"
IF Data_Confidentiality_level = "Medium" *THEN* Encryption_requirement = "Medium"
IF Data_Confidentiality_level = "Low" *THEN* Encryption_requirement = "Low"

The algorithm for information profiling can be observed in Algorithm 2.

C. ALGORITHM PROFILING

In this step, various security ciphers, already implemented in 6LoWPAN, are profiled according to their resource consumption. Various parameters are investigated at this stage, including memory, battery and CPU time consumption for each algorithm along with their security strengths. Following steps are involved in algorithm profiling:

Step 1: Identify Security Ciphers Implemented In 6LoWPAN
Literature was consulted for the exploration of different encryption algorithms available for 6LoWPAN. For the inclusion criteria, the weighted average technique was used, where for each literature instance, the score of the algorithm was incremented by one. Also, each algorithm instance incremented the total algorithms score, which is the combined occurrences of all the algorithms.

The average was computed by dividing the total algorithms score with the total number of algorithms present in the literature for 6LoWPAN, as mentioned in eq. (1). For the final inclusion, all the algorithms that have more occurrences than the average algorithm score, were included in the encryption protocol suit.

In total, ten encryption algorithms were included in the protocol suite based on the above-mentioned criteria. Among them, AES, DES, Blowfish, Camellia, Skipjack, RC5 and RC6 are block ciphers; RSA and DSA are Public-key ciphers; while ECIES is an Elliptic curve cipher.

$$\text{AvgAlgorithmScore} = \frac{\Sigma\,\text{AlgorithmsScore}}{\Sigma\,\text{Algorithm}} \qquad \text{(Eq. 1)}$$

Step 2: Identify Resources Consumption of Security Algorithms
For the identification of resources utilized by different encryption algorithms, a text file of size 10KB was used. All the ten algorithms encrypted the file independently to get the resources utilization details of various ciphers.

- *Memory:* To extract the memory consumption of each algorithm, the RSS value from statm file was extracted before and after executing every encryption algorithm. The initial and final values were then subtracted to get the memory consumption of each encryption algorithm. The memory consumption of each algorithm to encrypt a 10KB file is shown in table 1. According to the initial footprints, memory consumption of ECC is higher than other algorithms while Camellia consumes the lowest memory.
- *Battery:* To extract the battery consumption of each algorithm, the kinetic battery model [15] was used. The power trace notifies the remaining energy of all the nodes periodically. The battery consumption of all the included encryption algorithms was computed similar to memory consumption and shown in Table 1. According to the initial footprints, the battery consumption of ECC is higher than other algorithms while Skipjack consumes the lowest battery.

ALGORITHM 3: ENCRYPTION ALGORITHM PROFILING

```
Input:
Device Resource Consumption
    battery:
        apply KibaM model to get available battery
        apply encryption algorithm
        difference of the initial and final value gives battery consumption of the algorithm
    Memory:
        read RSS before and after executing algorithm
        difference of initial and final value gives memory consumption of the algorithm
    Execution Time:
        InitialTime = clock_time();
        //rest of the code.
        FinalTime = clock_time();
        diff = FinalTime  - InitialTime ;
        num_seconds = (double) diff / CLOCK_SECOND;
```

CPU time: To measure the execution time of each algorithm, the initial clock counter was stored before code execution and final clock counter was stored after it completed execution. The initial and final values were then subtracted to get the CPU consumption of each encryption algorithm. The time required by each algorithm to encrypt a file of size 10KB is shown in table 1. According to the initial footprints, the execution time of AES is lower than other algorithms while the execution time of DSA is greatest.

Step 3: Identify Security Strengths of Algorithms

Every encryption algorithm utilizes different resources and has different security strengths. To identify the security strengths of each cipher, relevant literature review was performed. According to the literature [46-51], the encryption algorithms are categorized according to their security strength as shown in Table 1.

TABLE 1
INITIAL PROFILING OF EACH ALGORITHM

Algorithm	Memory (KB)	Battery (Joules)	Execution time (ms)	Strength
RSA	752	23.61	110	High
ECC	1608	50.55	110	High
DSA	1068	30.75	70	High
AES	720	23.09	90	High
Blowfish	844	20.99	220	High
RC6	356	18.83	80	Medium
DES	672	12.46	70	Medium
Skipjack	392	5.89	80	Low
Camellia	152	18.46	80	Low
RC5	228	13.38	80	Low

The algorithm for encryption algorithm profiling can be observed in Algorithm 3.

D. MAPPING

The device, information and algorithm profiling are used for the mapping of optimal encryption algorithm according to the context of the information and device. The device capabilities are mapped to the algorithm’s required resources and the information confidentiality level is mapped to the algorithm's strength, to find the optimal algorithm. These two steps are briefly outlined below:

Step 1: Map security strength of algorithm over information security requirement

Once the data generated by the IoT device is classified according to its security demand, such as high, medium or low, all the available encryption algorithms are sorted according to the desired security strength. For instance, if the data is highly confidential and requires strong encryption algorithm, all the ciphers will be sorted according to the security strengths they provide, as well as the resources they consume. Later, the device current capabilities will further decide the final selection of

ALGORITHM 4: MAPPING ALGORITHM

```
Input:
Sort algorithm in descending order on the basis of device profiling
Check information confidentiality level
Check device capabilities (battery, memory)
    IF (confidentiality = "High")
        THEN
            Create list of all algorithms
            Sort in descending order
            for all algorithms
                IF (battery_of_device >= list_of_Algorithm[i].battery
                AND memory_of_device >= list_of_Algorithm[i].memory)
                    RETURN list_of_Algorithm_High_Security[i];
    ELSE IF (confidentiality = "Medium")
        THEN
            Create list of algorithms having security strength medium and low
            Sort in descending order according to device resources
            for all algorithms
                IF (battery_of_device >= list_of_Algorithm[i].battery
                AND memory_of_device >= list_of_Algorithm[i].memory)
                    RETURN list_of_Algorithm_Medium_Security[i];
    ELSE IF (confidentiality = "Low")
        THEN
            Create list of algorithms having security strength low
            Sort in ascending order according to device resources
            RETURN list_of_Algorithm_Low_Security[0];
```

encryption algorithm.

Step 2: Map device capabilities over resource consumption of algorithm

Once the encryption algorithms are sorted with respect to information requirements, the device available capabilities are checked to select the final cipher for encryption. Various resource requirements of each algorithm were calculated in the previous step including memory consumption, battery consumption and execution time. These resource consumptions are mapped on device available resources. For instance, if the information demands high confidentiality level, but the available resources of the device do not satisfy requirements of any algorithm, the protocol suite will select the best possible algorithm that can be executed by the available resources. The algorithm for mapping can be observed in Algorithm 4.

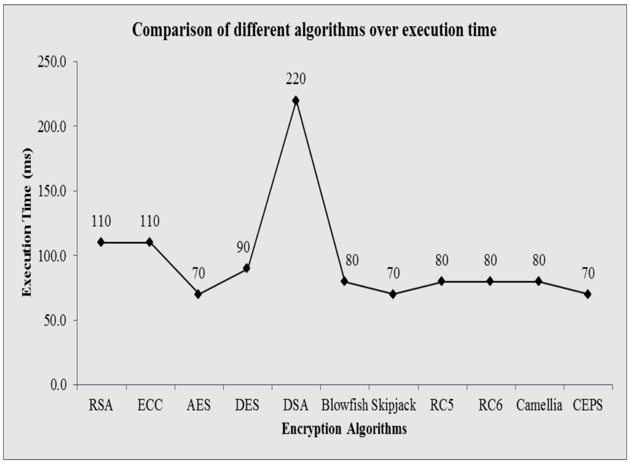

FIGURE 2. Execution time comparison of CEPS with other available algorithms

VI. RESULTS AND DISCUSSION

This section presents simulation and evaluation of our proposed context-aware protocol suite. The proposed was evaluated on the basis of memory, battery and processing time consumption and was compared with already existing solutions in 6LoWPAN.

The objective of this work is to provide a context-aware protocol suite according to the device available resources and information confidentiality demands. For this purpose, we selected ten encryption algorithms which are already implemented in 6LoWPAN including AES, RSA, DSA, ECC, DES, Blowfish, Skipjack, Camellia, RC5 and RC6.

The implementation of the protocol suite was done in ns-3.26, which is an open source simulator developed in C++ [52]. It is a network simulator for discrete events, primarily targeted to carry out research about a computer network. The ns-3 project was used because it provides a solid base for simulation, well- documented, easy to use and provides debugging facilities [53]. It has various built-in libraries and network topologies which are used for defining the simulation. ns-3 uses real clock during simulation instead of virtual clock which makes the results close to the hardware testbed results [54]. Other major reasons for its adaption in this article include its support for complete network topology for IoT devices [54] as well as IPv6 [55] [56].

Along with ns-3, Crypto++ cryptography toolkit[1] was also used to demonstrate the performance of security algorithms in 6LoWPAN. Crypto++ is a freely available open source cryptographic library developed in C++ and is widely used for research. It provides complete implementation of security ciphers, hash functions, authentication codes, and key agreement structures[2].

6LoWPAN[3] was designed for small sensor devices which cannot afford big code size, its complexity, and network overhead. The protocol provides a data packet size of 81 octets, and a data rate of up to 250 kbps. The simulation module of 6LoWPAN installs its stack on top of already existing NetDevice. NetDevice provides an interface to access and manage devices by IP and hides details of physical and MAC layers. Two different asynchronous functions (send and receive) were implemented to get notifications about data transmission. Some other functions were also written to compute the available resources including memory, battery, and execution time

To measure the performance of the context-aware encryption protocol suite, a 10KB file is encrypted with all the available options. Various encryption algorithms currently available for 6LoWPAN are compared with each other as well as with the proposed encryption protocol suite on the basis of memory utilized, battery consumed and execution time. In the following, the description of each of the modules is provided:

A. EXECUTION TIME

To measure the execution time of an encryption algorithm, the clock library present inside ns-3 was used. The clock_init() function was initialized at the start of the program, which uses hardware time along with the interrupts. Function clock_time() returns the current tick of the clock. Hence, to measure the execution time of a program, initial clock tick counter is stored before code execution and final tick counter is stored once the code completes its execution.

To encrypt a file of size 10 KB, the highest execution time was found to be of DSA, which was 220 milliseconds, while the lowest execution time was found to be of AES, with 70 milliseconds. On average, encryption algorithms took around 100 milliseconds to encrypt a file of 10 KB. However, CEPS took 70 milliseconds to encrypt the same file, which is 68% less than DSA and 30% less than the average case, as shown in Figure 2.

B. MEMORY

To measure the memory (RAM) utilized by an encryption protocol, the memory usage probe was used through /proc/self/statm [16] virtual file. This file provides

[1] https://github.com/weidai11/cryptopp/releases/tag/CRYPTOPP_8_0_0
[2] https://www.cryptopp.com/docs/ref/
[3] https://tools.ietf.org/html/rfc4919

physical memory consumed by each of the algorithms as well as CEPS.

The memory consumption of ECC was highest with a footprint of 1608 KB, while Camellia consumed the least memory with 152 KB. On average, encryption algorithms took around 679 KB of memory to encrypt a file of 10 KB. On the other hand, the memory consumption of CEPS was 334 KB to encrypt the same file, which is 79% less than ECC and 51% less than the average case, as shown in Figure 3. Camellia and RC5 take less memory than CEPS, however, both of these algorithms provide minimum security to each type of information, whereas CEPS provides significantly higher security to confidential information, achieving an overall better performance.

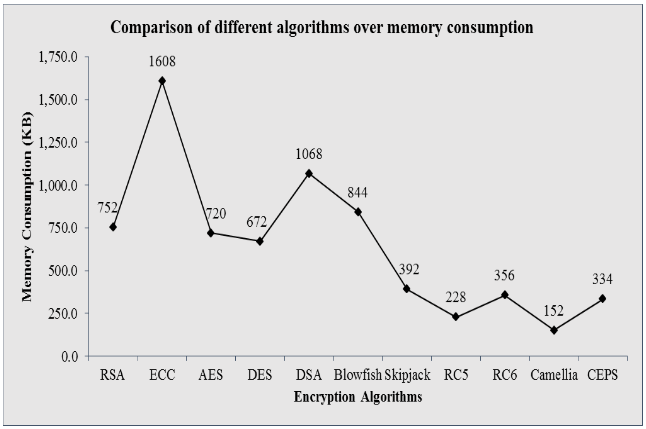

FIGURE 3. Memory comparison of CEPS with other algorithms

C. BATTERY

When a device is powered on, a counter starts measuring the estimated battery consumption, and when the device is turned off, the current value of the battery consumption is stored. The difference between the start value and the current value is then multiplied by the device power. The current values of the skymote microcontroller are given in eq. (2). The power trace can be calculated by adding the consumption states of active CPU, Low Power Mode (LPM), transmit and receive. The following equation can be used to get the energy consumption of the sensor node, where the value of skymote microcontroller voltage is given as 3V [15].

$$\text{Energy Consumed} = \frac{((1.8*\text{CPU}+0.051*\text{LPM}+21.8*\text{Receive}+19.5*\text{Tx})*3)}{\text{CLOCKS\_PER\_SEC}} \quad \text{(Eq. 2)}$$

The battery consumption of ECC was highest among the other encryption algorithms with a footprint of 50.55 joules, while skipjack consumed the least battery consumption with 5.89 joules. On average, encryption algorithms took around 22 joules of battery consumption to encrypt a file of 10 KB. However, the battery consumption of CEPS was 17 joules to encrypt the same file, which is 56% less than ECC and 21% less than the average case, as shown in Figure 4.

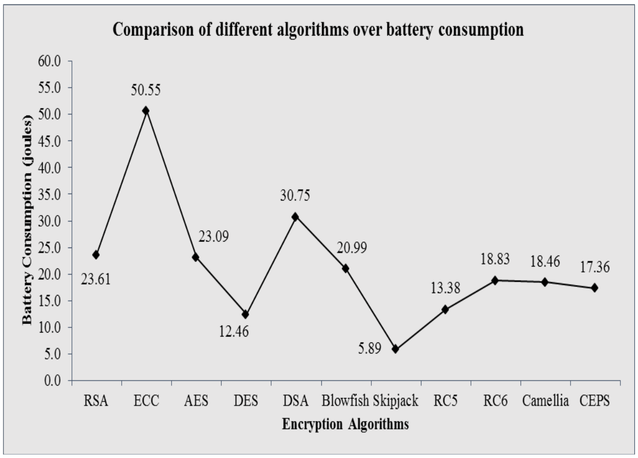

FIGURE 4. Battery comparison of CEPS with other encryption algorithms

V. CONCLUSION

IoT devices are usually resource constrained in terms of battery, memory and CPU power; facing few big challenges like stringent latency and enhanced security. Since these issues cannot be addressed effectively by the centralized computing architectures. Therefore, for IoT systems, Edge computing offers a fast response time and better computational power. However, due to lack of resources, the adaption of an encryption algorithm according to device capabilities is a challenging task. Currently, IoT devices use one encryption algorithm to encrypt data, however, the data generated by IoT devices may vary in confidentiality level and thus do not require same level of security. This paper addressed this issue and proposed a context-aware encryption protocol suit for Edge computing based IoT systems which adapts the encryption algorithm based on the information sensitivity as well as the device available resources.

This paper proposed a context-aware encryption protocol suite for IoT devices. Different modules of device profiling, information profiling, algorithm profiling, and mapping were designed for the selection of optimal encryption algorithm according to the device available resources and information requirements. For device profiling, kinetic battery model [15] and memory usage probe [16] were used for battery consumption and memory consumption, respectively, while fuzzy logic [34] was used

for information classification. For the simulation and implementation of different encryption algorithms, ns-3 and crypto++ were used. By using the proposed context-aware encryption protocol suite, we are able to save 79% memory consumption, 56% battery consumption and 68% execution time.

There are several shortcomings in the proposed research. For example, for the device profiling, only battery and memory consumptions were traced, whereas bandwidth and processing power of the device could also be considered. Also, some paid simulator could be used to get more accurate results. Moreover, in this research, encryption ciphers implemented in 6LoWPAN were used. In future, similar protocols can be designed for other IoT communication technologies like WiFi, Bluetooth, Zigbee, and WiMax.

REFERENCES

[1] K. Ashton, "That 'internet of things' thing," RFID journal, vol. 22, no. 7, pp. 97--114, 2009.

[2] J. Gubbi, R. Buyya, S. Marusic and M. Palaniswami, "Internet of Things (IoT): A vision, architectural elements, and future directions," Future generation computer systems, vol. 29, no. 7, pp. 1645--1660, 2013.

[3] L. Hou, S. Zhao, X. Xiong, K. Zheng, P. Chatzimisios , M. S. Hossain and W. Xiang, "Internet of things cloud: architecture and implementation," IEEE Communications Magazine, vol. 54, no. 12, pp. 32--39, 2016.

[4] R. Khan, . S. . U. Khan, R. Zaheer and S. Khan, "Future internet: the internet of things architecture, possible applications and key challenges," in cFrontiers of Information Technology (FIT), 2012 10th International Conference on, IEEE, 2012, pp. 257--260.

[5] C. Hennebert and S. J. Dos, "Security protocols and privacy issues into 6LoWPAN stack: A synthesis," IEEE Internet of Things Journal, vol. 1, no. 5, pp. 384--398, 2014.

[6] C. H. Liu, Z. Sheng, V. C. Leung, W. Moreno, K. K. Leung and Ö. Yürür, "Context-awareness for mobile sensing: A survey and future directions," IEEE Communications Surveys \& Tutorials, vol. 18, no. 1, pp. 68--93, 2016.

[7] A. G. Roselin, P. Nanda and S. Nepal, "Lightweight Authentication Protocol (LAUP) for 6LoWPAN Wireless Sensor Networks," in Trustcom/BigDataSE/ICESS, 2017 IEEE, IEEE, 2017, pp. 371--378.

[8] C. Kolias, A. Stavrou, J. Voas, I. Bojanova and R. Kuhn, "Learning internet-of-things security" hands-on"," IEEE Security \& Privacy, vol. 14, no. 1, pp. 37--46, 2016.

[9] X. Su, P. Li, Y. Li, H. Flores, J. Riekki, and C. Prehofer, "Towards semantic reasoning on the edge of IoT systems," in Proc. the 6th International Conference on the Internet of Things, 2016, pp. 171–172.

[10] B. Gu, Z. Zhou, S. Mumtaz, V. Frascolla, and A. K. Bashir. Context-Aware Task Offloading for Multi-Access Edge Computing: Matching with Externalities. IEEE Globecom. 2018.

[11] I. Yaqoob, E. Ahmed, M. H. ur Rehman, A. I. A. Ahmed, M. A. Al-garadi, M. Imran and M. Guizani, "The rise of ransomware and emerging security challenges in the Internet of Things," Computer Networks, vol. 129, pp. 444--458, 2017.

[12] M. Marjani, F. Nasaruddin, A. Gani, A. Karim, I. A. T. Hashem, A. Siddiqa and I. Yaqoob, "Big IoT data analytics: architecture, opportunities, and open research challenges," IEEE Access, vol. 5, pp. 5247--5261, 2017.

[13] G. Bansod, A. Patil, S. Sutar and N. Pisharoty, "ANU: an ultra-lightweight cipher design for security in IoT," Security and Communication Networks, vol. 9, no. 18, pp. 5238--5251, 2016.

[14] I. F. Siddiqui, N. M. F. Qureshi, M. A. Shaikh, B. S. Chowdhry, A. Abbas, A. K. Bashir, and S. U. J. Lee. Stuck-at Fault Analytics of IoT Devices Using Knowledge-based Data Processing Strategy in Smart Grid. Wireless Personal Communications, Springer, 2018.

[15] A. Riker, M. Curado and E. Monteiro, "Neutral Operation of the Minimum Energy Node in Energy-Harvesting Environments," in 2017 IEEE Symposium on Computers and Communication (ISCC), 2017, pp. 1-6.

[16] D. Adrianto and F. J. Lin, "Analysis of security protocols and corresponding cipher suites in ETSI M2M standards," in Internet of Things (WF-IoT), 2015 IEEE 2nd World Forum on, IEEE, 2015, pp. 777—782.

[17] H. Suo, J. Wan, C. Zou and J. Liu, "Security in the internet of things: a review," in Computer Science and Electronics Engineering (ICCSEE), 2012 international conference on, vol. 3, IEEE, 2012, pp. 648--651.

[18] C. Perera and A. V. Vasilakos, "A knowledge-based resource discovery for Internet of Things," {Knowledge-Based Systems, vol. 109, pp. 122--136, 2016.

[19] C. Perera, A. Zaslavsky, P. Christen and D. Georgakopoulos, "Context aware computing for the internet of things: A survey," *IEEE Communications Surveys \& Tutorials,* vol. 16, pp. 414--454, 2014.

[20] Y.-W. Kao and S.-M. Yuan, "User-configurable semantic home automation," Computer Standards & Interfaces, vol. 34, no. 1, pp. 171--188, 2012.

[21] Q. Zhou, S. Natarajan, Y. Simmhan and V. Prasanna, "Semantic information modeling for emerging applications in smart grid," in Information Technology: New Generations (ITNG), 2012 Ninth International Conference on, IEEE, 2012, pp. 775--782.

[22] K. Taylor, C. Griffith, L. Lefort, R. Gaire, M. Compton, T. Wark, D. Lamb, G. Falzon and M. Trotter, "Farming the web of things," IEEE Intelligent Systems, vol. 28, no. 6, pp. 12--19, 2013.

[23] A. Hristoskova, V. Sakkalis, G. Zacharioudakis, M. Tsiknakis and F. De Turck, "Ontology-driven monitoring of patient's vital signs enabling personalized medical detection and alert," Sensors, vol. 14, no. 1, pp. 1598--1628.

[24] C. Preist, J. Esplugas-Cuadrado, S. A. Battle, S. Grimm and S. K. Williams, "Automated business-to-business integration of a logistics supply chain using semantic web services technology," in International Semantic Web Conference, Springer, 2005, pp. 987--1001.

[25] C. M. Medaglia and A. Serbanati, "An overview of privacy and security issues in the internet of things," in The Internet of Things, Springer, 2010, pp. 389--395.

[26] A. Messer, I. Greenberg, P. Bernadat, D. Milojicic, D. Chen, T. J. Giuli and X. Gu, "Towards a distributed platform for resource-constrained devices," in Distributed Computing Systems, 2002. Proceedings. 22nd International Conference on, IEEE, 2002, pp. 43--51.

[27] T. Hofer, W. Schwinger, M. Pichler, G. Leonhartsberger, J. Altmann and W. Retschitzegger, "Context-awareness on mobile devices-the hydrogen approach," in System Sciences, 2003. Proceedings of the 36th Annual Hawaii International Conference on, IEEE, 2003, p. 10.

[28] W. Y. Lum and F. C. Lau, "A context-aware decision engine for content adaptation," IEEE Pervasive computing, vol. 1, no. 3, pp. 41--49, 2002.

[29] M. Taneja and A. Davy, "Resource aware placement of IoT application modules in Fog-Cloud Computing Paradigm," in Integrated Network and Service Management (IM), 2017 IFIP/IEEE Symposium on, IEEE, 2017, pp. 1222--1228.

[30] P. Sathyamoorthy, E. C.-H. Ngai, X. Hu and V. Leung, "Profiling Energy Efficiency and Data Communications for Mobile Internet of Things," Wireless Communications and Mobile Computing, 2017.

[31] A. Musaddiq, Y. B. Zikriya, O. Hahm, H.J. Yu, A. K. Bashir, and S.W. Kim. A Survey on Resource Management in IoT Operating Systems. IEEE Access. vol. 6, pp. 8459-8482. 2018.

[32] M. A. Zardari and L. T. Jung, "Data security rules/regulations based classification of file data using TsF-kNN algorithm," Cluster Computing, vol. 19, no. 1, pp. 349--368, 2016.

[33] M. A. Zardari, L. . T. Jung and M. N. B. Zakaria , "Data Classification Based on Confidentiality in Virtual Cloud Environment," Research Journal of Applied Sciences, Engineering and Technology, vol. 8, no. 13, pp. 1498--1509, 2014.

[34] M. Mohammadian and D. Hatzinakos, "Data classification process for security and privacy based on a fuzzy logic classifier," International Journal of Electronic Finance, vol. 3, no. 4, pp. 374--386, 2009.

[35] A. Safi, "Improving the Security of Internet of Things Using Encryption Algorithms," World Academy of Science, Engineering and Technology, International Journal of Computer, Electrical, Automation, Control and Information Engineering, vol. 11, no. 5, pp. 546--549, 2017.

[36] X.-W. Wu, E.-H. Yang and J. Wang, "Lightweight security protocols for the Internet of Things," in Personal, Indoor, and Mobile Radio Communications (PIMRC), 2017 IEEE 28th Annual International Symposium on, IEEE, 2017, pp. 1--7.

[37] F. Hamad, L. Smalov and A. James, "Energy-aware Security in M-Commerce and the Internet of Things," IETE Technical review, vol. 26, no. 5, pp. 357--362, 2009.

[38] G. Glissa and A. Meddeb, IEEE 802.15. 4 security sublayer for OMNET++, IEEE, 2017, pp. 1891--1896.

[39] J. Toldinas, R. Damasevicius, A. Venckauskas , T. Blazauskas and J. Ceponis, "Energy consumption of cryptographic algorithms in mobile devices," Elektronika ir Elektrotechnika, vol. 20, no. 5, pp. 158--161, 2014.

[40] S. U. Khan, C. Pastrone, L. Lavagno and M. A. Spirito, "An authentication and key establishment scheme for the IP-based wireless sensor networks," Procedia Computer Science, vol. 10, pp. 1039--1045, 2012.

[41] A. Liu and P. Ning, "TinyECC: A configurable library for elliptic curve cryptography in wireless sensor networks," in Proceedings of the 7th international conference on Information processing in sensor networks, IEEE Computer Society, 2008, pp. 245--256.

[42] P. Szczechowiak,, L. B. Oliveira, M. Scott, M. Collier and R. Dahab, "NanoECC: Testing the limits of elliptic curve cryptography in sensor networks," in Wireless sensor networks, Springer, 2008, pp. 305--320.

[43] C. Matthias, S. Kris, B. An, S. Ruben, M. Nele and A. Kris, "Study on impact of adding security in a 6LoWPAN based network," in Communications and Network Security (CNS), 2015 IEEE Conference on, IEEE, 2015, pp. 577-584.

[44] S. Raza, S. Duquennoy, J. Höglund, U. Roedig and T. Voigt, "Secure communication for the Internet of Things—a comparison of link-layer security and IPsec for 6LoWPAN," Security and Communication Networks, vol. 7, pp. 2654--2668, 2014.

[45] W. Jung, S. Hong, M. Ha, Y.-J. Kim and D. Kim, "SSL-based lightweight security of IP-based wireless sensor networks," in Advanced Information Networking and Applications Workshops, 2009. WAINA'09. International Conference on, IEEE, 2009, pp. 1112-1117.

[46] M. Mathur and A. Kesarwani, "Comparison between Des, 3des, Rc2, Rc6, Blowfish And Aes," in Proceedings of National Conference on New Horizons in IT-NCNHIT, 2013, pp. 143--148.

[47] M. Ebrahim, S. Khan and U. B. Khalid, "Symmetric algorithm survey: a comparative analysis," 2014.

[48] R. Tripathi and S. Agrawal, "Comparative study of symmetric and asymmetric cryptography techniques," International Journal of Advance Foundation and Research in Computer (IJAFRC), vol. 1, no. 6, pp. 68--76, 2014.

[49] T. Nie, C. Song and X. Zhi, "Performance evaluation of DES and Blowfish algorithms," in Biomedical Engineering and Computer Science (ICBECS), 2010 International Conference on, IEEE, 2014, pp. 1--4.

[50] H. K. Verma and R. K. Singh, "Performance analysis of RC5, Blowfish and DES block cipher algorithms," International Journal of Computer Applications (0975--8887) Volume, 2012.

[51] M. Agrawal and P. Mishra, "A comparative survey on symmetric key encryption techniques," International Journal on Computer Science and Engineering, vol. 4, no. 5, p. 877, 2012.

[52] A. Kumar, K. Gopal and A. Aggarwal, "Simulation and analysis of authentication protocols for mobile Internet of Things (MIoT)," in PDGC, 2014 International Conference on, IEEE, 2014, pp. 423—428.

[53] G. Brambilla, M. Picone, S. Cirani, M. Amoretti and F. Zanichelli, "A simulation platform for large-scale internet of things scenarios in urban environments," in Proceedings of the First International Conference on IoT in Urban Space, ICST, 2014, pp. 50--55.

[54] S. R. Prasad, R. Vivek and J. Mungara, NS3 simulation studies for optimized neighbour discovery in 6LoWPAN networks, IEEE, 2016, pp. 15—18.

[55] G. D'Angelo, S. Ferretti and V. Ghini, "Simulation of the Internet of Things," in High Performance Computing \& Simulation (HPCS), 2016 International Conference on, IEEE, 2016, pp. 1—8.

[56] M. S. H. Talpur, M. Z. A. Bhuiyan and G. Wang, "Shared--node IoT network architecture with ubiquitous homomorphic encryption for healthcare monitoring," International Journal of Embedded Systems, vol. 7, no. 1, pp. 43--54, 2014.