A Secured Demand Side Management Engine for Smart Societies using Industrial IoT and Big Data Analytics

Muhammad Babar, Fazlullah Khan, Mian Ahmad Jan, Abid Yahya, Fahim Arif,
Zhiyuan Tan, Joseph Chooma

***Abstract*—Smart societies have an increasing demand for quality-oriented services and infrastructure in an Industrial Internet of Things (IIoT) paradigm. Smart urbanization faces numerous challenges. Among them, secured energy Demand Side Management (DSM) is of particular concern. The IIoT renders the industrial systems to malware, cyber attacks, and other security risks. The IIoT with the amalgamation of Big Data analytics can provide efficient solutions to such challenges. This paper proposes a secured and trusted multi-layered DSM engine for a smart social society using IIoT-based Big Data analytics. The proposed engine uses a centralized approach to achieve optimum DSM over a Home Area Network (HAN). To enhance the security of this engine, a payload-based authentication scheme is utilized that relies on a lightweight handshake mechanism. Our proposed method utilizes the lightweight features of Constrained Application Protocol (CoAP) to facilitate the clients in monitoring various resources residing over the server in an energy-efficient manner. In addition, data streams are processed using Big Data analytics with MapReduce parallel processing. The proposed authentication approach is evaluated using NetDuino Plus 2 boards that yield a lower connection overhead, memory consumption, response time and a robust defense against various malicious attacks. On the other hand, our data processing approach is tested on reliable datasets using Apache Hadoop with Apache Spark to verify the proposed DMS engine. The test results reveal that the proposed architecture offers valuable insights into the smart social societies in the context of IIoT.**

Keywords: Industrial Internet of Things, Demand Side Management, Home Area Network, Smart Societies, Security, Trust.

I. Introduction

We are living in a time on this planet when 54% population is living in urban areas; with a prediction of an increase up to 66% by 2050 [1], [2]. This urbanization will bring numerous challenges for the decision makers in providing various services and fulfillment of infrastructure needs. The Internet of Things (IoT) covers the everyday objects of physical world by enabling them to network with other objects [3]. It was predicted that the connected things will increase 31% in the year 2017, in comparison to 2016, by reaching 8.4 billion interconnected devices and will cross 20 billion by year 2020 [4]. With the advent of IoT, all the dumb devices at homes will be able to hear, listen and communicate with each other by forming a Home Area Network (HAN). Moreover, this information will be shared over the Internet to make it ubiquitous [5]. With this information sharing, the dumb devices will transform into smart devices to reform the human life style. IoT is a mesh network of home/everyday objects, and these objects may be equipped with pervasive intelligence [6].

The critical sensors and devices in essential industry infrastructure with existing IoT are interconnected to form the Industrial IoT (IIoT). Usually, IIoT exploitation permits businesses and clients into industrial practices and accomplish sky-scraping production by reducing cost. Meanwhile, IIoT renders the industrial systems to malware, cyber attacks, and a number of vulnerabilities. The incorporation of objects within the Internet needs a variety of communication models. This necessity will likely add a number of novel and ingenious adversarial models to the IIoT [7]. In IIoT, security provisioning is a cumbersome task because each object has its own unique features. The distinctiveness of each object, person, and system linked to the Internet needs to be verified. The products and solutions available in the market are lacking secured features and they are vulnerable to a wide range of security breaches.

The huge number of smart devices in IIoT environment will produce data of high volume, high velocity, and/or high variety information (3V) assets, that require new forms of processing to enable enhanced decision making, insight discovery and process optimization called Big Data. Sharda, et al. [8], [9] added veracity, variability, and value to these three Vs. The application of advanced analytic techniques on large data sets is known as Big Bata analytics that consists of two parts, i.e., the Big Data set and analytics. Advancement in IIoT is resulting in a large amount of valueable data. With the help of Big Data technologies and efficient machine learning algorithms, there is a great prospective of analytical services to the urban citizens and decision makers [10], [11]. A smart society aims to raise the quality of service (QoS) and optimize these services for its residents [12].

With the advancement of technology and increased urbanization, power management has a vital role in the development of the nations. The increased energy consumption with inability to meet the rising demand, efficient energy utilization and load management is the main point of focus in many countries around the globe. Energy management aims to optimize the generation and distribution of energy [14]. The traditional perception about electric energy system is one way and is top-down oriented, i.e., the power is generated in power plants and transmitted to the industries and homes for usage through grids. Power plants generate electric energy and feeds it to the grid that try to equalize the demand and supply balance at all time. Balancing the demand and supply of energy is a critical aspect in operating an electric energy system. Demand side management (DSM) is used to change the consumer behavior towards the energy usage. In [14], the author divided the DSM depending upon the time and impact, into following categories: 1) efficient usage of energy, 2) time of energy usage, and 3) response to the demand.

In view of the above discussion, we propose a centralized DSM engine for smart social society, using IIoT and Big Data analytics. The major contributions of this paper are as follow.

1) The proposed DSM engine has two parts. The first part is responsible for the authentication of HAN, using a novel payload-based mutual authentication mechanism. Using the lightweight CoAP protocol of IoT, requests and responses are exchanged among the clients and server for secured session establishment within HAN.
2) The second part is responsible for data stream processing and decision making for the DMS. In addition, data stream processing is performed using Big Data analytics with MapReduce parallel processing mechanism. The selection of running devices is made with the help of 0/1 Knapsack Algorithm in the decision-making stage.

The remaining sections are organized as follows. A review of related works is presented in Section II. Prospective of Demand Side Management in Smart Society: A Birds Eye view is given in Section III. The proposed Demand Side Management Engine and technology stack is presented in Section IV. Section V elaborates the results. Finally, section VI presents the conclusion of the proposed work.

II. Related Work

Internet of Things (IoT) is a merger of different technologies to connect and make use of smart devices that would change the daily life behaviors. Riggins F and Wamba F. [15] proposed a framework that has its basis on the idea of evolution of IoT. According to them, the evolution will take place from monitored things to the network of things. Sun et al. [16] endorses an innovative concept of smart connected communities (abbreviated as SCC). The concept evolved from the smart cities to enhance the present living standards and meet the future demands in a technical and innovative way. A case study is also presented for smart tourism using the integration of IoT and Big Data analytics. Rathore et al. [17] proposed four tier architecture to empower the decision making of the societies, so that the right decision can be taken at the right time without delay. The functionality includes the 1) collection, 2) aggregation, 3) communication, 4) processing, and 5) interpretation. The implementation is performed using Hadoop and Spark to provide the real-time data processing. The system performance is tested for processing time and throughput.

Katal et al [18] defines the Big Data as the huge amount of data that need new technologies and techniques for its processing. They explained the associated properties like Variety, Volume, Velocity, Variability, Complexity and Value; they also discussed the challenges and issues related to Big Data analytics including privacy, security, data access, information sharing, analytical, skill requirement and few technical challenges. Barbato F, Capone F. [19] classifies the Demand side optimization techniques based on three point criteria 1) User interaction: this deals with the modeling of the user behavior and further categorized into individual and cooperative users 2) optimization approach: how the problems are dealt either by deterministic or stochastic approach and 3) Time scale defines the planning scope either as futuristic (a day ahead) or real time. Then they formulated a DSM optimization Model. The proposed model tries to achieve some objectives such as bill, discomfort minimization and maximization of locally generated energy; with some constraints like electric devices (fixed, shift-able, and elastic), local energy generators, and energy storage and balance and energy tariffs.

Similarly, regarding security authentication the related work is also provided here. There are various authentication schemes are found in literature for securing IoT objects to better manage communications between these objects. The HTTP-based web technology utilizes the Representational State Transfer (REST) architecture [20]. The Internet Engineering Task Force (IETF) [21] has created a particular group known as Constrained RESTful Environments (CoRE). This group is accountable for designing a lightweight protocol to offer web resources for IoT [22]. The CoAP protocol is one such product of this working group which inherits a subset of the HTTP features to meet the requirements of a resource-constrained IoT [23]. CoAP utilizes a easy request/response model of interaction for swap the resources between clients and servers. every client has the choice to register itself with a particular server for the resource observation [24]. CoAP is a perfect substitute for the current IoT protocols such as XMPP [25] and MQTT [26].

Therefore, CoAP is implemented in a variety of applications such as home automation system [27], transport logistics [28], freight supervision [29], and smart cities [30]. DTLS requires to be outlined to build it more friendlily toward the resource-constrained networks [30]. Bhattacharyya et al. [31] proposed a lightweight authentication scheme to establish a unicast communication channel. There are anumber of proposals are found in literature which are based on DTLS. For securing the communication in a CoAP-based environment of IoT, the utilization of DTLS as the underlying protocol is studied [32]. Similarly, a robust security scheme is proposed [33] and the performance of DTLS hand-shaking is investigated for the resource-starving sensors (objects) [34]. In addition, the DTLS implementation for smart phones (INDIGO) is proposed using CoAP [35]. Moreover, lightweight authentication scheme in a CoAP-based IoT environment is proposed for resource observation [36]. Even though, DTLS-based mechanism sustains a great varity of cipher suites, however, it was initially developed for those networks which have rich resources. The resource-consuming multifaceted cipher suites of DTLS do not take the message length into consideration as a critical criterion for network security. Consequently, using DTLS for an IoT implementation is an exclusive option and may not be a most favorable solution for securing the network.

III. Prospective of Demand Side Management in Smart Society: A Bird's Eye View

DMS can be also be defined as maintaining the efficient utilization of the energy by selecting the high priority and high demand devices at right time while remaining in the prescribed load and cost limits and according to the specified parameters or constrains [14], [37], [38]. The DSM illustration is depicted in **Figure 1**. The parameters which are taken into consideration in DSM include:

1) **Device Load:** This is the minimum load required to operate a specific device.
2) **Cost:** This is the most critical parameter from consumer point of view and it is the correlation of time and demand and can be different at different time of a particular day but in generally price tariffs from energy supplier are considered authority in this regard.

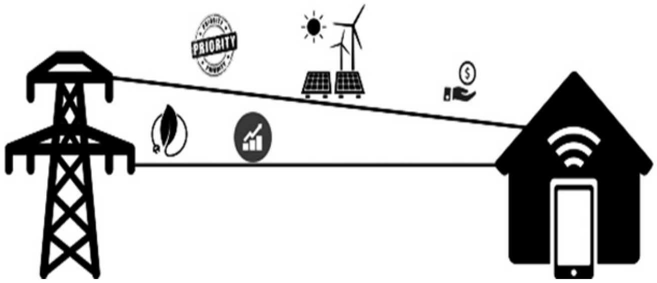


Fig. 1: Demand Side Management: An illustration

3) **Load Limit:** It is the most critical parameter from supplier point of view; generally it is a limit specified by the supplier and it may change according to consumer hours and tariffs.
4) **Time:** The time may be the deciding factor for many of the above parameters as the cost and the load is measured according to the time unit.
5) **Device Priority:** Priority can be defined as the measure of use or importance at particular time of the day. Hence, priority of a device can vary during different hours of a particular day. A specific device could be in one and only one priority of the following at a time and could change dynamically according to situation.
   a) **Real Time:** The device is absolutely required regard less of the load and cost limits or according to the trade-offs.
   b) **High:** The Device is in high demand while remaining in the load but not cost limit or vice versa but not both.
   c) **Medium:** The device is required but when it is feasible to remain in the load and cost limit. If no other constrain is specified.
   d) **Low:** The device is at low priority and can be shut down for high priority device in demand.
   e) **Not Required:** The device will remain off until the priority is changed.

DSM Techniques:

From supplier prospective load is the primary and from consumer side its the secondary point of concern. The main objective of DSM is to remain in the load and cost limits. This can be achieved with the techniques shown in **Figure 2**.

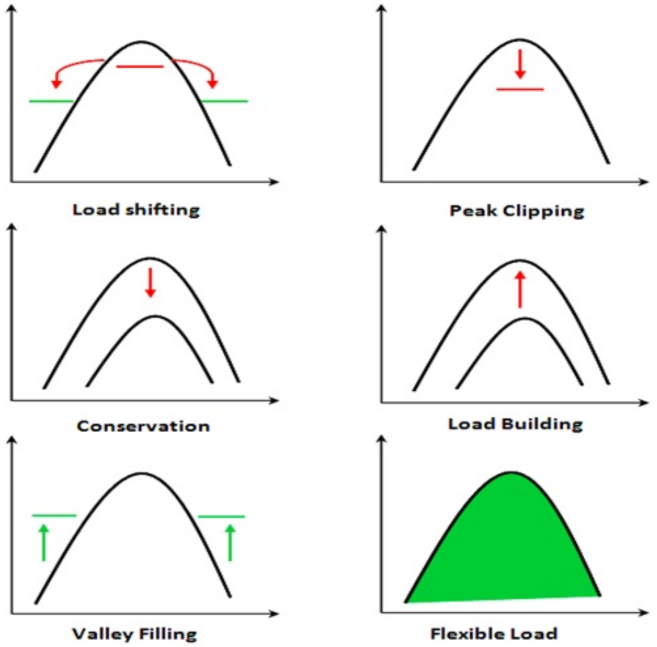


Fig. 2: DSM Techniques

1) **Load Shifting:** In this technique the load is shifted from peak hours to the off peak hours, in other words it is the way to segregate the devices based on their priority level. The low priority devices will be shifted to the other suitable time while the high priority devices can be kept on running. The main drawback of this technique is that you have to switch off some of the devices.
2) **Peak Clipping:** This is direct load control technique to reduce peak load and achieving the optimum level.
3) **Conservation:** This technique is used to reduce the energy consumption
4) **Load Balancing:** This is the technique to balance the load
5) **Valley Filling:** Valley filling is the opposite of peak clipping. It directly increases the load in off-peak hours. Other possible techniques are shown in the figure 2.
6) **Flexible Load:** It is a technique that provides the facility of the flexibility the energy.

Trade-offs:

To achieve the desire results consumer needs to make several trade-offs between different desires. Some of them are as follow:

1) **Availability vs. Cost:** It's totally a consumer decision what is most important for him; the 24/7 availability or the cost optimization and more availability means high cost.

$$\text{Availability } \alpha \text{ Cost}$$

2) **Availability vs. Load:** High availability means increase in load. So if high availability is chosen its the responsibility of supplier to provide as much load as needed at the consumer end. This can be achieved through alternate sources of energy
3) **Cost vs. Load:** Cost and load are directly proportional to each other if more load is needed the cost will increase but not in all cases e-g., the cost may be decrease in off-peak hours. If these trade-offs are not chosen by the consumer then DSM technique should be capable enough to choose optimum values.

Combinatorial Optimization

Combinatorial optimization is about finding the optimal set based on some criteria given a finite set of objects. In the proposed engine, the choice of devices is based on the parameters defined above. In addition, 0/1 Knapsack algorithm is used in the proposed engine for the selection of devices which can be described as:

$$Max \sum_{i=1}^{n} DeviceCost_i \tag{1}$$

subject to

$$\sum_{i=1}^{n} DeviceLoad_i \leq Loadlimit \tag{2}$$

IV. Proposed Methodology

In this section the comprehensive and detail description of the proposed architecture is discussed. The proposed DMS engine is a multilayered centralized DMS engine which is composed of 1) Data/Message Receiving and Pre-Processing, 2) Data Stream Processing, and Decision-Making and User Interface Layers. The position of the DSM engine in the smart social society is shown in **Figure 3**. The HANs shown in the area are connected to this engine through LAN or MAN. Before going into the detail description of each layer of the proposed DSM engine, the overview of the architecture is given first.

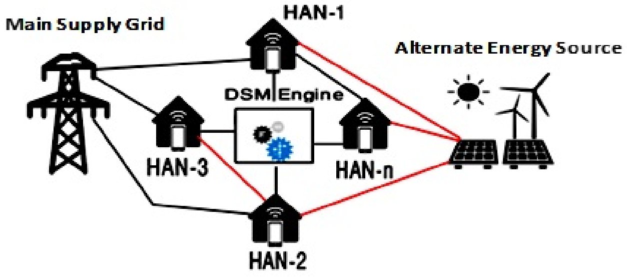


Fig. 3: DSM Engine for Smart Social Society

A. *Overview of Proposed DSM Engine*

The overview of the multilayered engine is given in this section. A high level architecture (overview) of the proposed DSM engine is shown in the **Figure 4**. The Architecture consists of multiple layers including a number of components. First of all the data produced by different HANs connected to the proposed engine are collected by the message receiver. The message identification and authentication is performed at this stage.

The proposed architecture utilizes the payload based mutual authentication scheme in order to identify the message and provide secure connection. The payload authentication uses the CoAP as underlying protocol to meet the requirements. The message receiver is responsible for keys exchange and overall authentication. Afterwards, this data is forwarded to the pre-processing for removing the anomalies from the raw data as the data received by message receiver is the raw data acquisition from the outer IoT data sources. These anomalies may include the missing values, invalid data, unreliable data etc. the message identification process is also performed at this stage. Afterward, the filtration and extraction is performed in order to filter the data and remove the noise. Once the pre-processing is done, the data is ready to be processed.

B. *Proposed Demand Side Management Engine*

The proposed DMS engine is a multilayered centralized DMS engine for smart social societies which is composed of 1) Data/Message Security and Pre-Processing Layer, 2) Data Stream Processing Layer, and Decision-Making and User Interface Layer which are is graphically depicted in **Figure 5**. The exhaustive explanation of every layer of the proposed engine is given in the forthcoming segment.

1) Data/Message Security and Pre-processing Layer: This is the first layer of the proposed DSM engine. This layer is directly connected to the IoT data sources (HANs). The data is first collected and authenticated at this stage. The data from HANs is received in the form of messages which can be one of two types: 1) HAN configuration message, and 2) device message. HAN configuration message contains the setting/ preferences information of a particular HAN. This type of message may contain HAN id, Initial priorities of all devices, max load limit, and max cost limit. Device message contains the setting information of a particular device containing the information about a particular device such as device id, load, priority, and status. Therefore, first of all when the data is collected from data sources, the message identification process is carried out to provide secure and trusted data.

Payload based Mutual Authentication Scheme:: The payload authentication scheme is similar to CoAP because it uses the CoAP as underlying protocol to meet the requirements. However, the CoAP-based implementations for IoT are dependent on DTLS for the protected transfer of resources among the objects. Though, the DTLS-enabled CoAP stack produces an additional protocol layer for security provisioning which increases the computational and communication cost. There is no addition of the extra protocol layer in the proposed scheme which does not compromise the security of data messages

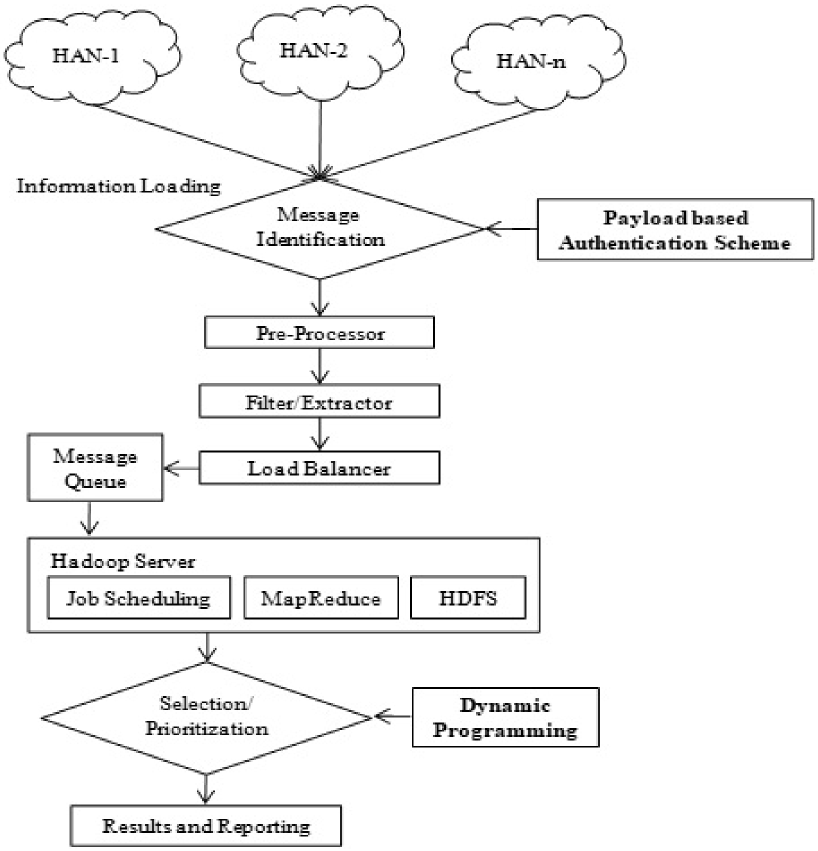


Fig. 4: Overview of DSM Engine

transferred among the clients and server. Authentication is offered at the time of request-response communication among the clients and server while the session key is exchanged within the payload of transmitted data (messages). This scenario is depicted in **Figure 5** in the security layer. For the authentication, the security features into CoAP are added in proposed scheme to make it more efficient, robust, and secure against various malicious activities. Contrasting the encryption techniques based on DTLS, the proposed method offers authentication using the payload of messages exchanged among the clients and the server. Both the server and client confront each other during the process of authentication. The whole process is carried out using 4 handshake messages where, the payload of each message is kept a maximum of 256 bits as shown in **Figure 6**. The proposed scheme is accomplished using the four steps which are 1) Session launching, 2) Server challenge, 3) Client reply and challenge, and 4) Server reply.

The session beginning is headed by the provisioning stage. It is a prerequisite stage (offline) during which the clients share a secret key, with the server which is only known to the server and the owner client of the key. The server preserves a record of these keys with unique identifier (ID) linked with it. Upon successful authentication, the communication of session key between both parties takes place. This mock-up assumes that the secret keys are embedded with sensors in each object at manufacturing and deployment. An alarm is generated to inform about the breach, if an intruder tries to temper with physical object.

Each client sends a request message to the server alike to a Hello Client message in the session initiation phase. The said request is sent to the server URI, /.well-known/authorize. The server gets the ID of the object from the message payload during the server challenge phase. The server carries out a table look-up for a corresponding key using this ID. The server responds back with an encrypted payload if a match is found. To generate the challenge, the server produces a nonce (pseudo-random), and a impending session key which is a temporary number that is used only once by an object. An encrypted payload is generated by the server at this stage. The client requires deciphering the encrypted payload to get back the session key in the client response and challenge phase. Finally, the server deciphers the encrypted payload of the client challenge to in the server response phase. If it is there, the server comprehends that the client has successfully authenticated itself.

Pre-processing:: As this is data is in the raw form and having a lot if discrepancies. Therefore, we need to perform the pre-processing techniques before the actual processing. In the pre-processing phase mostly validity and reliability checks are performed to make the data clean and error-free for processing. Todays real-world Big Data are extremely susceptible to inconsistencies, missing values, different formats, and noise due to their characteristically massive size and their

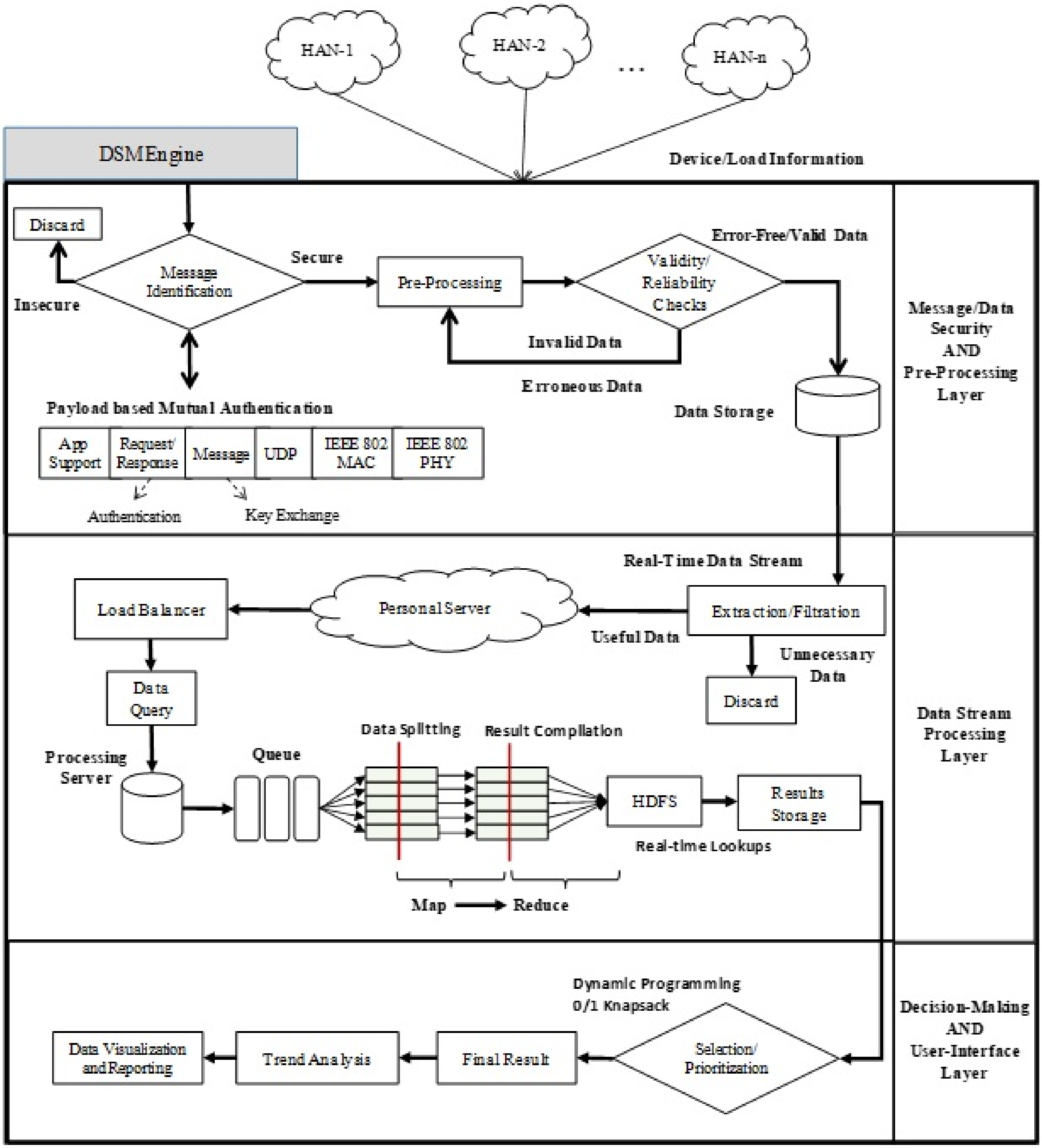
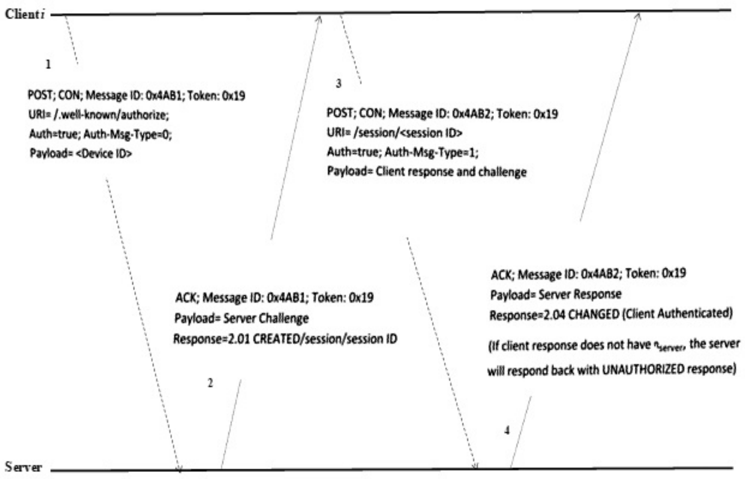


Fig. 5: Proposed DSM Engine

prospective origin from multiple, heterogeneous sources. The low-quality data will lead to low-quality processing results. Data pre-processing methods, when applied before the actual processing, can noticeably improve the overall quality of the data processing and/or the time required for the actual processing [39]–[41]. The proposed architecture performs the following before processing as pre-processing: 1) data reduction, 2) data cleaning, and 3) data transformation. Data reduction can be made practical to get a reduced representation of the datasets that is smaller to a great extent in quantity, yet closely preserves the integrity of the original data. In addition, processing and analysis on the reduced dataset is much more efficient yet produce almost the same analytical results. Data reduction typically applied on massive data that do not present



POST; CON; Message ID: 0x4AB1; Token: 0x19
URI= /.well-known/authorize;
Auth=true; Auth-Msg-Type=0;
Payload= <Device ID>

ACK; Message ID: 0x4AB1; Token: 0x19
Payload= Server Challenge
Response=2.01 CREATED/session/session ID

POST; CON; Message ID: 0x4AB2; Token: 0x19
URI= /session/<session ID>
Auth=true; Auth-Msg-Type=1;
Payload= Client response and challenge

ACK; Message ID: 0x4AB2; Token: 0x19
Payload= Server Response
Response=2.04 CHANGED (Client Authenticated)
(If client response does not have n_{server}, the server will respond back with UNAUTHORIZED response)

Fig. 6: Handshake process

many valued data. Likewise, data transformation is performed. The basic purpose of the transformation is to limit the value of data to a certain limited range. We preferred the Min-Max technique to transform a specific value X to another value Y. Finally, in data preprocessing, the data cleaning is carried out. The collected data could be incomplete that lacking attributes values or certain attributes of interest, noisy that is a random error or variance in a measured variable, and inconsistent that containing discrepancies in codes or names. In this research, the incompletes (missing values) and noise is taken into consideration.

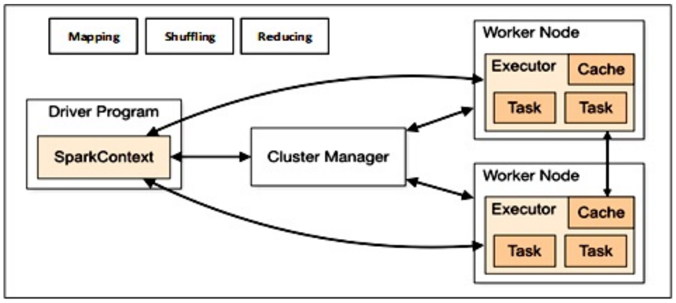


Fig. 7: Apache Spark Working

2) Data Stream Processing Layer: Stream processing layer is performed to process the data coming from different IoT sensors at real time after proper pre-processing. This processing is based on the notion of parallel processing in which various execution of processes are performed concurrently. Parallel processing formulate a program execute quicker because there are many engines (CPUs) running it. The processing of the data in proposed architecture starts with the data filtration. The filtration process is used to discard the useless data means which is not useful for the processing and does not affect the results of the processing. In the proposed architecture, the Kalman Filter (KF) is used to more speed up the data processing and separate the valuable and noisy data. KF is utilized to carry out the data filtration in order to filter the noise from the data. It is a statistical method and plays a noteworthy part in sensing of the real-world data [42], [43]. Once the filtration and extraction is done, the data is stored in a server for load balancing. Load balancing is carried out as we perform parallel processing. A load balancer is used to distribute the traffic across a number of parallel processors in order to distribute the load. It increases the reliability of processing along with the concurrent users capacity.

To be very precise in the case of data processing, the implementation is achieved using Hadoop framework with MapReduce mechanism. At this phase, the identical formation of MapReduce and HDFS is utilized. Moreover, other than HDFS, we can also use SQL supposed HBASE, and HIVE, for the administration of Database (Offline or in-memory) to store historical information. Since, a real-time stream of the data is processed. Hence, a third party tool is used for real-time stream processing to merge with the Hadoop to facilitate the real-time stream execution. To provide real-time realization, Apache Spark is used with the Hadoop which is a general-purpose engine for wide-ranging stream data processing. It provides speedy stream processing for Big Data and allows reusability across streaming applications. The Spark components are depicted in **Figure 7**.

3) Decision-Making and User-Interface Layer: After processing, results are communicated to respective HAN for compliance and data is stored in database for future trend analysis. In this phase, first of all the selection and prioritization is taken place. In selection and prioritization, the devices are selected based on the parameters provided by the consumers. The devices are selected in two different phases. In phase I, the HANs are prioritized based on their requirements and criteria provided by them and in Phase II, the devices are selected for a particular HAN. The input contains s set of finite number of devices with associate cost and load and output having a set of selected devices. The I/O can be represented as follow:

Input:

$$D = \{d_1, d_2, d_n\}$$
$$C : D \bar{a} R+$$
$$L : D \bar{a} R+$$

Output:

$$S = \{s_1, s_2, s_3\}$$
$$S \subseteq D \mid \text{Min} \sum_{s \in S}^{n} C_s$$
$$\text{and} \sum_{s \in S}^{n} L_s \leq \text{MaxLoad}$$

The selection and prioritization activity is performed using dynamic programming approach though 0/1 Knapsack algorithm in the proposed DMs engine.

Dynamic Programming: A dynamic programming (DP) is a technique (algorithm) which is based on recursion (like divide-and-conquer) with some variation. Basically, DP is the combination of recursion and some common sense. Recursion lets you to describe a functions value in the form of other value, where common sense describes the implementation of function in such a way that recursion is done in advance and results are stored to be accessed easily that eventually makes the program faster. It is also known as memorization because it memorizes the outcome of some specific states which can be accessed in later stage to solve sub-problem. In this a sub-problem is solved by a particular sub-solution which is previously constructed and it makes this technique very fast than other techniques. Most of the DP problems can be classified into two different types: 1) Optimization problems, and 2) Combinatorial problems. The optimization problems anticipate you to choose a viable result to minimize or maximize the value of the required function. On the other hand, combinatorial problems anticipate you to discover the number of possible and optimal ways to accomplish something. Each DP problem has a plan to be tracked:

- To highlight that the problem can be divided into most favorable sub-problems
- Repeatedly describe the solution by stating it in form of best possible solutions for sub-problems (smaller)
- Calculate the value of the best possible solution in bottom-up manner
- Design most favorable solution from the calculated information.

As we need to find an optimal object/solution from finite set. This scenario is included in combinatorial optimization. In such problems the traditional exhaustive searching is not feasible because in such problems the optimal solutions are discrete or tends to be discrete. Therefore, we prefer the dynamic programming approach and the knapsack problem which is in the combinatorial optimization.

Dynamic Programming based 0/1 Knapsack Algorithm:: The one of the general problem is the 0-1 knapsack problem that limits the number xi of replicas of all type of entry to 0 or 1. For example, various items set the knapsack to find the max total value. Every item has particular weight and value. Total weight will always be less than fixed weight called W. So weights and values of items must be taken into consideration is shown in Table I.

Given a set of n items from $1 - n$, all with a value v_i and a weight w_i, with a capacity of maximum weight W, can be mathematically represented as:

$$Max \sum_{i=1}^{n} v_i w_i \tag{3}$$

TABLE I: Knapsack Problem

Item No.	Weight	Value
1	1	8
2	3	6
3	5	5

subject to,

$$\sum_{i=1}^{n} w_i x_i \leq W \ and \ x_i \in \{0, 1\} \tag{4}$$

where, x_i denotes the instances of i consist of the knapsack.

Once the optimal solutions are obtained, then the final results are produced and store in a particular storage area for event management. These results can be used to perform the trend analysis and reporting.

V. Experimental Results and Analysis

The experimental results for proposed system are provided in this section. The security authentication module is implemented using NetDuino Plus 2 boards for the client and server interaction model. The experimental work is performed using .NET Micro Framework for resource-constrained devices with at least 256 KB of flash and 64 KB of RAM. The CoAPSharp library is also used that provides basic communication. On the other hand, the stream processing module of proposed DSM engine is implemented using Apache Spark with Hadoop single node setup on UBUNTU 16.04 LTS. Input libraries are used for processing and producing Hadoop Readable form (sequence file) at collection and aggregation unit so that it can be processed by Hadoop.

A. Security Authentication using Proposed Architecture

In this section the experimental results with regard to security layer authentication scheme is provided. In **Figure 8**, the comparison of the proposed authentication scheme with the CoAP-based DTLS implementation (INDIGO) for smartphones is provided. DTLS* symbolizes the handshake between standard computer and a smart-phone where the computer operates as a server and the smart-phone acts as a client. On contrary, DTLS+ symbolizes the computer as a client and the smart-phone as a server.

Similarly, the CoAP messages are transferred asynchronously over the UDP sockets in the proposed architecture. Each client keeps record of the transferred CON requests to maintain track. When a matching an RST response or is received for such messages, the transmission is considered successful. The average response time for a single confirmable message for a payload of one byte is compared against DTLS and the CoAP protocol (with no security), as depicted in **Figure 9**.

Correspondingly, the average memory consumption of a message at the compile time is also acquired using the Microsoft.SPOT.Native assembly. The proposed scheme is compared with existing ones for a confirmable message of 500 bytes shown in **Figure 10**. A considerable quantity of memory

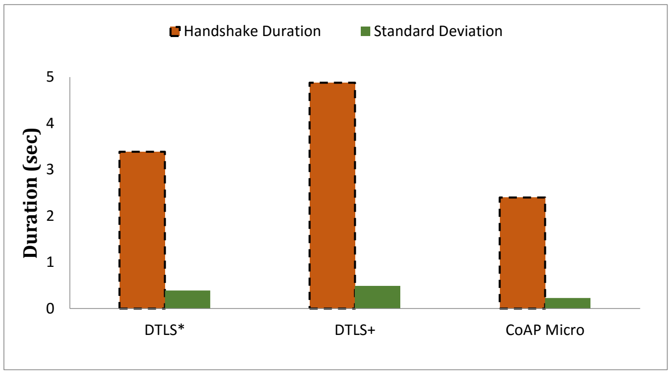


Fig. 8: Handshake Duration

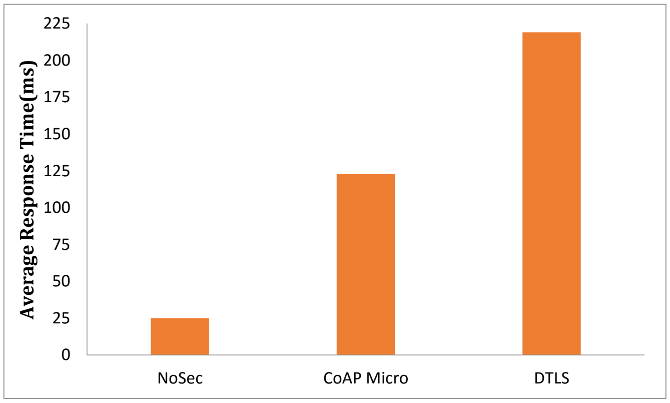


Fig. 9: Average Response Time

to the messages compile time) is allocated by CoapBlip [35] and its variation TinyCoAP [36] which is an adaptation of the standard C libraries that need TinyOS element for its installation on a sensor node. On the other hand, HTTP/UDP has a low memory foot-print as it does not provide a reliability mechanism or a request/response matching.

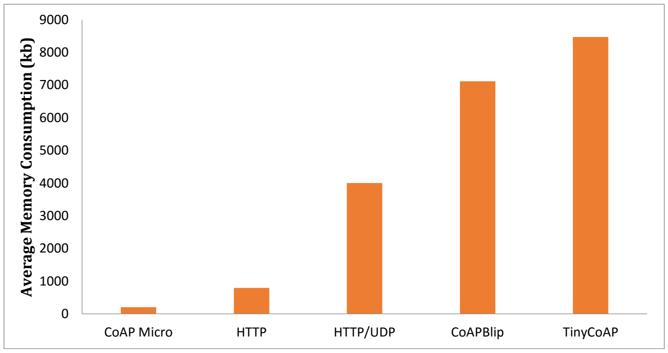


Fig. 10: Memory Consumption (500 Bytes payload)

B. Stream Processing and Decision-Making using Proposed Architecture

In this section the experimental results with regard to stream processing using big Data analytics is provided. The proposed DSM engine stream processing is experimentally validated in a smart home scenario of different sensor. The energy consumption of the sensors is tested by turning-on the devices randomly. The said smart home is composed of one kitchen, four rooms, and two toilets. Every room has 4 devices and the kitchen is allotted with 6 appliances ranging from high to lower energy consumption. Similarly, the toilet has 2 appliances. Therefore, we have 26 sensors in the whole smart home scenario. Furthermore, some of the devices that are lastingly present in the on-state, for instance refrigerator, etc. are not taken into consideration in the proposed scheme. The results disclose that when the consumers are regularly using the devices the energy consumption is noteworthy increases. On the other hand, when applying the proposed architecture to the smart home environment the energy utilization considerably decreases. **Figures 11, 12, 13**, and **14** depict the energy consumption of different appliances such as computer, air-conditioning system, television, and light. The consumption of the devices is significantly reduced on the route of a week time. Furthermore, the energy consumption is optimized as well; while in the traditional consumption is not optimized that is clearly depicted in the graph. Likewise, the unsuitable energy consumption is managed and controlled in the case of proposed architecture. Therefore, the consumer can efficiently manage the needless power consumption of the machines using the proposed system.

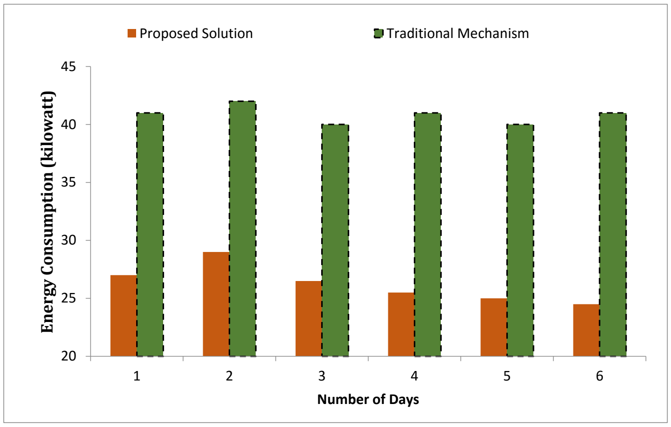


Fig. 11: Energy Consumption of Air-Condition

VI. Conclusion

This paper proposes a secure and trusted multilayered DSM engine for a smart social society using IIoT and Big Data analytics. The proposed engine is a centralized approach in order to achieve optimum DSM over a network of Home Area Network (HAN). Proposed architecture of centralized Demand Side Management Engine takes Input from IoT sensors of HANs and performs authentication and Big Data analytics for the selection and prioritization of devices of a particular HAN. The payload based authentication scheme is utilized in proposed DSM engine to get the security. The payload-based encryption method utilizes an easy 4-way handshake mean to authenticate the participating objects. In addition, data stream

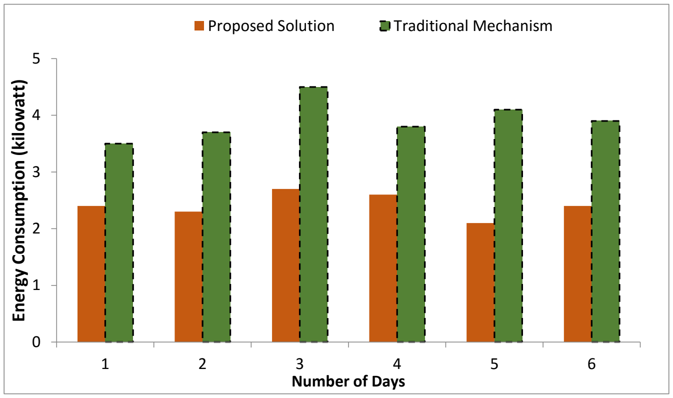


Fig. 12: Energy Consumption of Computer

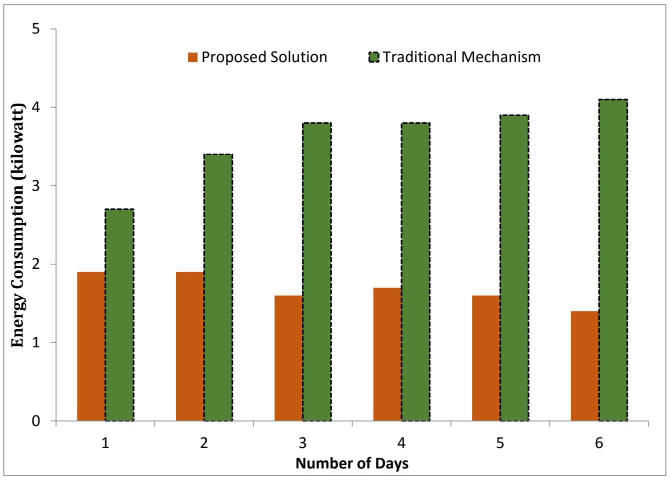


Fig. 13: Energy Consumption of Light Source

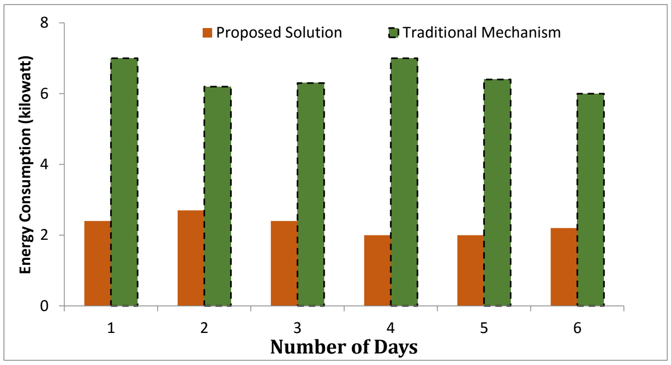


Fig. 14: Energy Consumption of Light Television

processing is performed using MapReduce parallel processing mechanism. The security perspective of proposed system is evaluated using NetDuino Plus 2 boards which reveals that the proposed authentication is incurs less connection overhead, computationally efficient, and offers a robust defense against various cyber attacks. On the other hand, the stream processing is performed using Apache Hadoop with Apache Spark to verify the proposed DMS engine and the examination reveals that the proposed architecture offers valuable insights into the smart social societies in the context of IIoT.

References

[1] Philippe Bocquier. World urbanization prospects: an alternative to the un model of projection compatible with the mobility transition theory. *Demographic Research*, 12:197–236, 2005.

[2] Barney Cohen. Urbanization in developing countries: Current trends, future projections, and key challenges for sustainability. *Technology in society*, 28(1-2):63–80, 2006.

[3] Ala Al-Fuqaha, Mohsen Guizani, Mehdi Mohammadi, Mohammed Aledhari, and Moussa Ayyash. Internet of things: A survey on enabling technologies, protocols, and applications. *IEEE Communications Surveys & Tutorials*, 17(4):2347–2376, 2015.

[4] Gartner Incorporation. Gartner press release. http://www.gartner.com/newsroom/id/3598917, Last accessed on April 7, 2018.

[5] Deze Zeng, Song Guo, and Zixue Cheng. The web of things: A survey. *Journal of Communication*, 6(6):424–438, 2011.

[6] Feng Xia, Laurence T Yang, Lizhe Wang, and Alexey Vinel. Internet of things. *International Journal of Communication Systems*, 25(9):1101, 2012.

[7] Jorge Granjal, Edmundo Monteiro, and Jorge Sá Silva. Security for the internet of things: a survey of existing protocols and open research issues. *IEEE Communications Surveys & Tutorials*, 17(3):1294–1312, 2015.

[8] Gartner Incorporation. The importance of big data: A definition. https://www.gartner.com/doc/2057415/importance-big-data-definition, Last accessed on May 8, 2018.

[9] Ramesh Sharda, Dursun Delen, and Efraim Turban. *Business intelligence: a managerial perspective on analytics*. Prentice Hall Press, 2013.

[10] Philip Russom et al. Big data analytics. *TDWI best practices report, fourth quarter*, 19(4):1–34, 2011.

[11] Martin Strohbach, Holger Ziekow, Vangelis Gazis, and Navot Akiva. Towards a big data analytics framework for iot and smart city applications. In *Modeling and processing for next-generation big-data technologies*, pages 257–282. Springer, 2015.

[12] Muhammad Babar and Fahim Arif. Real-time data processing scheme using big data analytics in internet of things based smart transportation environment. *Journal of Ambient Intelligence and Humanized Computing*, pages 1–11, 2018.

[13] Cyril Cecchinel, Matthieu Jimenez, Sébastien Mosser, and Michel Riveill. An architecture to support the collection of big data in the internet of things. In *Services (SERVICES), 2014 IEEE World Congress on*, pages 442–449. IEEE, 2014.

[14] Peter Palensky and Dietmar Dietrich. Demand side management: Demand response, intelligent energy systems, and smart loads. *IEEE transactions on industrial informatics*, 7(3):381–388, 2011.

[15] Frederick J Riggins and Samuel Fosso Wamba. Research directions on the adoption, usage, and impact of the internet of things through the use of big data analytics. In *System Sciences (HICSS), 2015 48th Hawaii International Conference on*, pages 1531–1540. IEEE, 2015.

[16] Yunchuan Sun, Houbing Song, Antonio J Jara, and Rongfang Bie. Internet of things and big data analytics for smart and connected communities. *IEEE Access*, 4:766–773, 2016.

[17] M Mazhar Rathore, Awais Ahmad, Anand Paul, and Seungmin Rho. Urban planning and building smart cities based on the internet of things using big data analytics. *Computer Networks*, 101:63–80, 2016.

[18] Avita Katal, Mohammad Wazid, and RH Goudar. Big data: issues, challenges, tools and good practices. In *Contemporary Computing (IC3), 2013 Sixth International Conference on*, pages 404–409. IEEE, 2013.

[19] Antimo Barbato and Antonio Capone. Optimization models and methods for demand-side management of residential users: A survey. *Energies*, 7(9):5787–5824, 2014.

[20] Roy T Fielding and Richard N Taylor. Principled design of the modern web architecture. *ACM Transactions on Internet Technology (TOIT)*, 2(2):115–150, 2002.

[21] IETF. Internet engineering task force. ttps://www.ietf.org, Last accessed on May 27, 2018.

[22] Jaewoo Kim, Jaiyong Lee, Jaeho Kim, and Jaeseok Yun. M2m service platforms: Survey, issues, and enabling technologies. *IEEE Communications Surveys and Tutorials*, 16(1):61–76, 2014.

[23] Zach Shelby, Klaus Hartke, and Carsten Bormann. The constrained application protocol (coap). 2014.

[24] Klaus Hartke. Observing resources in the constrained application protocol (coap). 2015.

[25] Peter Saint-Andre. Extensible messaging and presence protocol (xmpp): Core. 2011.

[26] Dinesh Thangavel, Xiaoping Ma, Alvin Valera, Hwee-Xian Tan, and Colin Keng-Yan Tan. Performance evaluation of mqtt and coap via a common middleware. In *Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP), 2014 IEEE Ninth International Conference on*, pages 1–6. IEEE, 2014.

[27] Seung-Chul Son, Nak-Woo Kim, Byung-Tak Lee, Chae Ho Cho, and Jo Woon Chong. A time synchronization technique for coap-based home automation systems. *IEEE Transactions on Consumer Electronics*, 62(1):10–16, 2016.

[28] Markus Becker, Thomas Pötsch, Koojana Kuladinithi, and Carmelita Goerg. Deployment of coap in transport logistics. In *Proceedings of the 36th IEEE Conference on Local Computer Networks (LCN). Bonn Germany 2011*, 2011.

[29] Markus Becker, Koojana Kuladinithi, Thomas Pötsch, and Carmelita Görg. Wireless freight supervision using open standards. In *5 th International Workshop Cold Chain Management. June*, pages 10–11, 2013.

[30] Marco Centenaro, Lorenzo Vangelista, Andrea Zanella, and Michele Zorzi. Long-range communications in unlicensed bands: The rising stars in the iot and smart city scenarios. *IEEE Wireless Communications*, 23(5):60–67, 2016.

[31] A Bhattacharyya, A Ukil, T Bose, and A Pal. Lightweight mutual authentication for coap (wip), 2014.

[32] Jorge Granjal, Edmundo Monteiro, and Silva Jorga Sa. On the feasibility of secure application-layer communications on the web of things. In *Local Computer Networks (LCN), 2012 IEEE 37th Conference on*, pages 228–231. IEEE, 2012.

[33] Shahid Raza, Ludwig Seitz, Denis Sitenkov, and Göran Selander. S3k: Scalable security with symmetric keysdtls key establishment for the internet of things. *IEEE Transactions on Automation Science and Engineering*, 13(3):1270–1280, 2016.

[34] Thomas Kothmayr, Corinna Schmitt, Wen Hu, Michael Brünig, and Georg Carle. Dtls based security and two-way authentication for the internet of things. *Ad Hoc Networks*, 11(8):2710–2723, 2013.

[35] Daniele Trabalza, Shahid Raza, and Thiemo Voigt. Indigo: Secure coap for smartphones. In *Wireless Sensor Networks for Developing Countries*, pages 108–119. Springer, 2013.

[36] Mian Ahmad Jan, Priyadarsi Nanda, Xiangjian He, Zhiyuan Tan, and Ren Ping Liu. A robust authentication scheme for observing resources in the internet of things environment. In *Trust, Security and Privacy in Computing and Communications (TrustCom), 2014 IEEE 13th International Conference on*, pages 205–211. IEEE, 2014.

[37] Goran Strbac. Demand side management: Benefits and challenges. *Energy policy*, 36(12):4419–4426, 2008.

[38] Thillainathan Logenthiran, Dipti Srinivasan, and Tan Zong Shun. Demand side management in smart grid using heuristic optimization. *IEEE transactions on smart grid*, 3(3):1244–1252, 2012.

[39] Muhammad Babar and Fahim Arif. Smart urban planning using big data analytics to contend with the interoperability in internet of things. *Future Generation Computer Systems*, 77:65–76, 2017.

[40] Muhammad Babar, Ataur Rahman, Fahim Arif, and Gwanggil Jeon. Energy-harvesting based on internet of things and big data analytics for smart health monitoring. *Sustainable Computing: Informatics and Systems*, 2017.

[41] Muhammad Babar and Fahim Arif. Smart urban planning using big data analytics based internet of things. In *Proceedings of the 2017 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2017 ACM International Symposium on Wearable Computers*, pages 397–402. ACM, 2017.

[42] Dan Simon. Kalman filtering with state constraints: a survey of linear and nonlinear algorithms. *IET Control Theory & Applications*, 4(8):1303–1318, 2010.

[43] Di Li, Soummya Kar, José MF Moura, H Vincent Poor, and Shuguang Cui. Distributed kalman filtering over massive data sets: analysis through large deviations of random riccati equations. *IEEE Transactions on Information Theory*, 61(3):1351–1372, 2015.