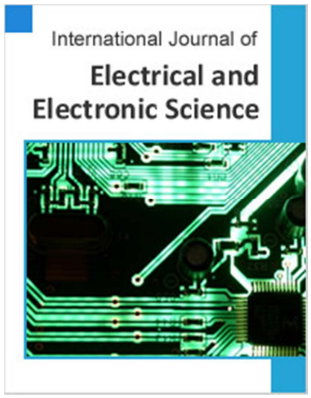
International Journal of Electrical and Electronic Science
2017; 4(2): 16-28
http://www.aascit.org/journal/ijees
ISSN: 2375-2998

AASCIT American Association for Science and Technology



Keywords
Internet of Things Gateway,
Wireless Sensor Network Security,
Intrusion Detection System,
Network Processor

Received: March 13, 2017
Accepted: May 3, 2017
Published: August 8, 2017

An Embedded Security Center for Internet of Things (IoT) Infrastructure

Qutaiba Ibrahim Ali

Computer Engineering Department, College of Engineering, Mosul University, Mosul, Iraq

Email address
Qut1974@Gmail.com

Citation
Qutaiba Ibrahim Ali. An Embedded Security Center for Internet of Things (IoT) Infrastructure. *International Journal of Electrical and Electronic Science*. Vol. 4, No. 2, 2017, pp. 16-28.

Abstract
Internet of Things (IoT) is a promising technology which has been widely used in different application fields such as warehouses, Smart Grids, public security, intelligent buildings and so on, following the progress in sensors, wireless communication and embedded systems. IOT gateway plays a significant responsibility in IoT as it can be considered as the bridge between sensor network and traditional communication network with their Internet applications. This paper gives an insight view to the design and realization challenges to localize an embedded security center into the IoT gateway nodes which connects Wireless Sensor Networks (WSN) to the Internet. The main goal is to protect the data of IoT gateway (needed for its operation) as well as its functionality and accessibility. The suggested embedded security center must responds to many objectives, it should ensure that the administrative information exchanged is correct and undiscoverable, the source is who he claims to and the system is robust and available. The proposed gateway security center consists of two ciphering methods (Advanced Encryption Standards (AES) & Rivest Cipher 4 (RC4)) to provide data encryption to the whole path from the WSN nodes to the server, a Hashed Message Authentication Code (HMAC) function to provide message integrity and authentication between the gateway and the server, a keys generation module, bidirectional entity authentication and an embedded Intrusion Detection System (IDS) to defend against internet attacks. The proposed defense strategies took into account the embedded nature of a IoT gateway and hence the recommended solutions make a compromise between highly secured and good performed system.

1. Introduction

Wireless sensor networks (WSNs) are a rising technology that allows monitoring and control of the environment through limited devices called sensor nodes [1]. Their objective is to perform scattered sensing over an area and send their results to base stations through multi-hop wireless communication [2]. One of the challenges in this field is connecting these restrained devices directly to the Internet [3]. The future Internet, designed as an "Internet of Things" is foreseen to be "a world-wide network of interconnected objects uniquely addressable, based on standard communication protocols" [1]. Identified by a distinctive address, any object including computers, sensors, Radio Frequency Identification (RFID) tags or mobile phones will be able to join the network, collaborate and cooperate efficiently to achieve different responsibilities [1]. Connecting WSNs to the Internet is possible through a *Gateway*, which considers the existence of a device that acts as an application layer gateway, responsible for translating the lower layer protocols from both networks (e.g. Transport

Control Protocol/Internet Protocol (TCP/IP) and proprietary) and routing the information from one point to another [4]. As a result, Internet hosts and sensor nodes can be able to address each other and exchange information without establishing a truly direct connection [4]. In this solution, the WSN is still independent from the Internet, and all queries still need to pass through a gateway device. However, sensor nodes is still able to provide web service interfaces to external entities while maintaining their lower layer protocols [5], see Figure 1.

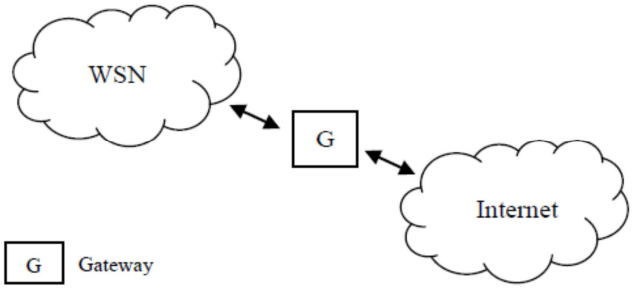


Figure 1. *Connecting WSN to Internet via Gateway.*

Deploying WSNs configured to access the Internet raises security challenges, which need to be considered before taking advantage of the many benefits of such integration.

There were many papers that consider some of the most significant WSN security problems. Most research on security in sensor networks has focused on prevention techniques, such as secure routing protocols [6-9], cryptography [10-13], key management [14, 15] and authentication techniques [16-18]. Other papers study WSN attacks which were classified based on various criteria, such as the domain of the attackers, or the techniques used in attacks [19, 20]. Another research direction focuses on the design, architecture and implementation of Intrusion detection systems in WSN world [21-24]. Current literatures present numerous architectures on IoT Gateways. The authors of [25] have put forward an architecture comprising of sensing layer, IoT network layer and IoT application layer. I. Gronbaekin [26] proposes a solution that allows service portability across systems. It is focused on the naming and addressing issues of connected objects, network elements supporting the application portability and related gateway architecture. Many previous works [27, 28, 29] deal with networks in sensing domain and describe the features of IoT gateway. Also a model gateway for IoT applications is proposed. Web IoT is presented in [30] which outlines the use of web services in realizing IoT architecture.

The current paper differs from previous works in implementing an embedded security frame work which was added to the gateway device mentioned earlier in order to enhance its security defense against different types of threats and attacks. The remainder of this paper is organized as follows: section 2 includes the suggested gateway architecture. Section 3 explains the ciphering module of the suggested system. Section 4 explains the keys generation module of the suggested system. Section 5 presents the bidirectional entity authentication. Section 6 includes the hardware implementation of WSN gateway armed with IDS on UBICOM platform and implementation of searching algorithm on the proposed WSN gateway with the results. Section 7 contains an overall system evaluation and finally, section 8 provides conclusions.

2. Architecture of the Suggested Gateway

In this paper, the wireless/wired connection between the gateway and its internet connection was protected using various security methods. An experimental prototype model of the gateway was built using Ubicom IP2022 network processer platform which provides a fully integrated platform: real Time Operating System (RTOS), TCP/IP protocol stack, and the necessary hardware. UBICOM's IP2022 chip embeds some basic hardware, but it permits combining it with on-chip software to support the most prevalent protocols. The same device can supports Ethernet, Bluetooth wireless technology, IEEE 802.11, and so on. The key to this approach is Software System on Chip (SOC) technology [25].

Wireless sensor network integration with the internet is composed of four parts [5], see Figure 2a:

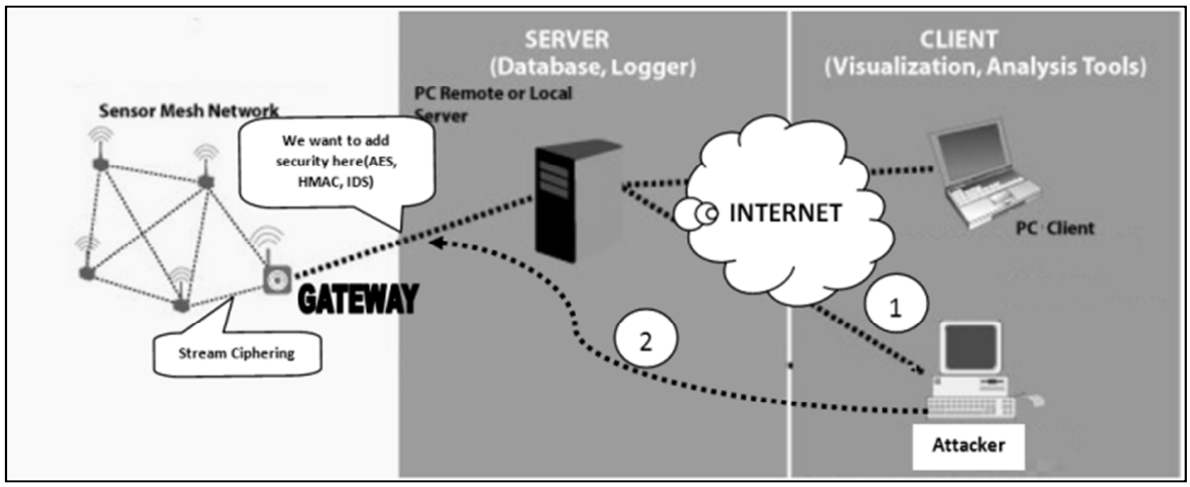
a) The Client: which provides the web access account for the different users and includes user visualization software and graphical interface for managing and monitoring the network.
b) The Server: which provides the Internet Service Providing (ISP) to the WSN nodes.
c) The Gateway: which is an always-on facility that handles translation and buffering of data from the wireless network and provides the bridge between the WSN nodes and the internet.
d) The Sensor nodes: where different nodes are connecting together to form the wireless sensor network. The wired or wireless connection between these nodes and the server is achieved via the gateway nodes.

From security point of view, the attacker can strike the system in many ways (sniffing, unauthorized access, Denial of Service. etc.) and positions as shown in Figure 2a. For example, the server could be attacked directly (attack 1 in Figure 2a) or the link between the server and the gateway is also vulnerable to different types of attacks (attack 2 in Figure 2a).

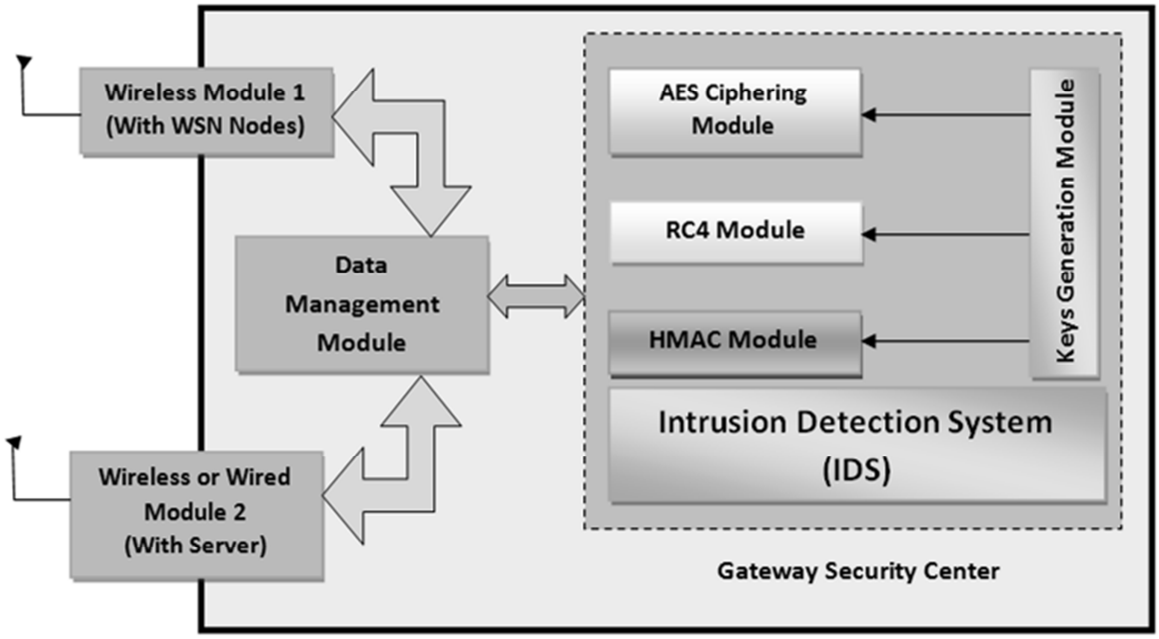
While the server could be protected using conventional security methods which afford Privacy, Integrity, Authentication and Non repudiation, the gateway still needs to be secured against different types of attacks. As mentioned earlier, the gateway node performs as a bridge between the WSN nodes and the server, so it needs to be protected in both directions. In order to protect a message, confidentiality, integrity and message authentication must be added. In addition, a well-organized Intrusion Detection System (IDS)

is needed to defend the system against the diverse types of attacks. Lastly, an appropriate and secured keys exchange method is required to reassign the keys of the different security algorithms.

Building on the above, the gateway architecture must be customized to comprise variety of security methods to assemble what is called a *gateway security center*, see Figure 2b. The recommended gateway security center consists of two ciphering systems to offer data encryption to the entire path from the WSN nodes to the server, a HMAC function to afford message integrity and authentication between the gateway and the server, a keys generation & distribution module, and an Intrusion Detection System (IDS) to guard against internet attacks.



(a)



(b)

Figure 2. *The suggested security method:(a) general view (b) the suggested gateway architecture.*

3. The Ciphering Modules

The efforts to build the ciphering engines of the gateways' security center in Ubicom's environment could be explained as follows:

3.1. AES Algorithm

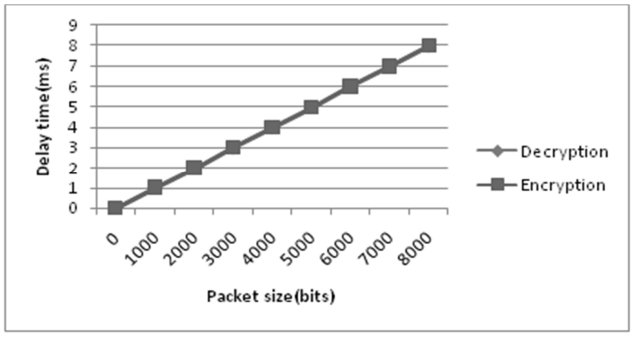
Advanced Encryption Standard (AES) is selected to afford the privacy methods to shield the transmitted packets and the digest between the gateway and the server (using two different AES keys) against unauthorized reading. The AES algorithm can use cryptographic keys of 128, 192, and 256 bits to encrypt and decrypt data in blocks of 128 bits [13]. UBICOM platform supports key size of 128 bits only, so that this length of key will be used. Figure 3a shows the time difference of AES Encryption & Decryption delay within the Ubicom platform with respect to packet size (payload) variation.

3.2. HMAC Algorithm

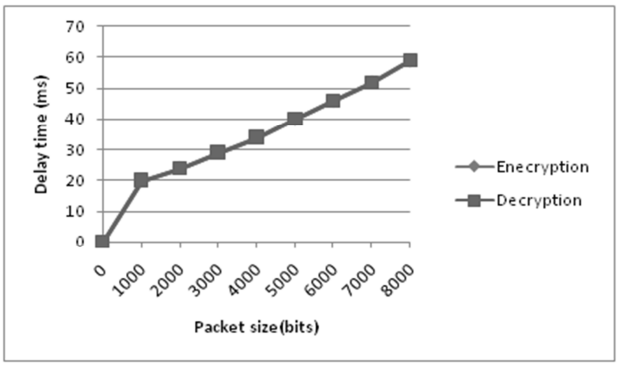
HMAC algorithm is chosen to provide message authentication and integrity between the gateway and the server. The Secure Hash Algorithm is a cryptographic hash functions. SHA-1 (Secure Hash Algorithm 1) is a revised version of SHA designed by the National Institute of Standards and Technology (NIST). A very interesting point about this algorithm and others is that they all follow the same concept. Each creates a digest of length N from a multiple-block message, each block is 512 bits in length [19]. A hash function guarantees the integrity of a message. It guarantees that the message has not been changed. A hash function, however, does not authenticate the sender of the message. The digest created by a hash function is normally called a Modification Detection Code (MDC). The code can detect any modification in the message. To provide message authentication, a modification detection code must be changed to a Message Authentication Code (MAC). This idea is a hashed MAC, called HMAC that can use any standard keyless hash function such as SHA-l or Message Digest 5 (MD5). HMAC creates a nested MAC by applying a keyless hash function to the concatenation of the message and an AES symmetric key [11]. Figure 3b shows the time variation of HMAC Encryption & Decryption delay within the Ubicom platform with respect to packet size (payload) variation.

3.3. RC4 Module

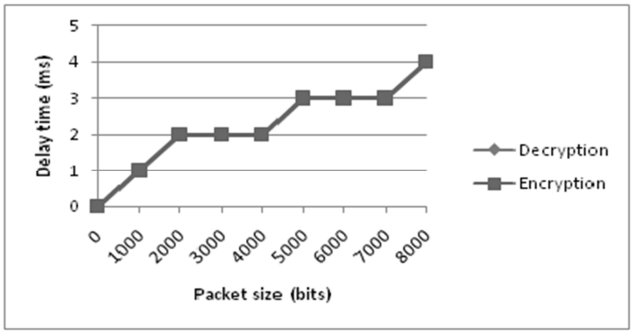
RC4 is a stream cipher extensively used in a lot of applications nowadays and in the wireless networks. With a solekey, a stream of pseudo-random numbers is generated. Then, the encryption of data using RC4 is basically based on XORing the pseudo-random numbers from the stream with the data [12]. RC4 is known to be fast and efficient, so that, a 128 bit RC4 was suggested (as a light weight and fast stream ciphering algorithm) to offer a realistic level of privacy to protect the transmitted packets between the gateway and its connected WSN nodes. In order to enforce RC4 functionality, its ciphering key was changed continuously every 15 Minute using the keys generation module mentioned later. Figure 3c shows the delay variation of RC4 Encryption & Decryption time within the Ubicom platform with respect to packet size (payload) variation.



(a)



(b)



(c)

Figure 3. *Encryption & Decryption Delay vs. Packet Length: (a) AES (b) HMAC (c) RC4.*

The major comments that could be extracted from the above performance tests is that RC4 is faster than AES when encrypting and decrypting different packet sizes. As expected, HMAC and its SHA1 hash function needs more processing to achieve its task, therefore it has the longest delay in the system.

4. Keys Generation & Distribution Module

The above security algorithms require different keys in order to carry out their functionality. Also, these keys must be altered at habitual time periods to solidify the mission of the eavesdroppers who try to crack the ciphering algorithm by determining its secured key. It is clear that keys exchanging among the server, the gateway and the WSN nodes is important to carry out keys update procedure. In this paper a new manner to achieve keys updating process without exchanging any portion of data is proposed. The proposed method presume the existence of *synchronized and equivalent pseudo random number generators* in the three parts of the system, having the same code functionality, their seeds are equivalent and produce their outputs at an identical time periods see Figure 4. In order to achieve this keys exchanging process, the administrator must first arrange the gateway to have the same timing values (i.e., Date and Time) of the server prior to place it in the field (the same procedure is repeated for the WSN nodes with respect to their gateway). This task could be performed using Global Positioning

System (GPS) receivers localized in each part of the system. The administrator must also determine the keys update period. The Ubicoms' function RND() is used as the pseudoandom number generators and fed it with the same seed value of the servers' pseudorandom number generator (another pseudorandom number generator has the nodes seeds was used to update the RC4 keys). This function was planned to generate an (128 bit AES key (data encryption)+ 128 bit AES key (Digest encryption) = 256bit) output (AES & HMAC) every one hour in synchronous with the server (PRNG1 & PRNG2 pair). On the other hand, PRNG3 & PRNG4 pair are activated every 15 Minute to change the 128 bit RC4 Keys.

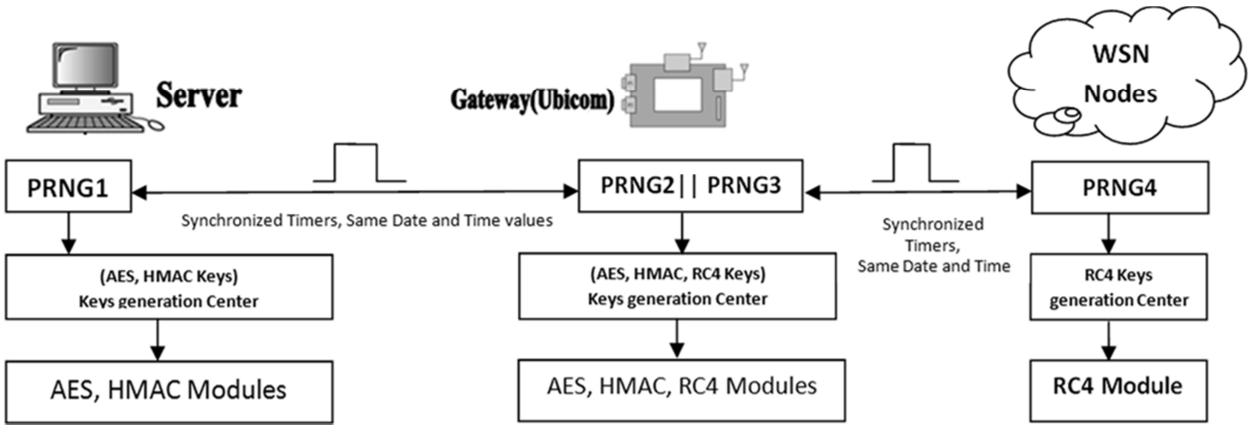


Figure 4. *The Suggested Keys Generation Algorithm.*

5. Bidirectional Entity Authentication

Prior to accepting the remote control request (i.e., accessing the reconfiguration web page of the gateway) made by the administrator, the gateway must check his identity. This can be done by adopting a particular challenge response procedure suggested in this paper, see Figure 5. The challenge is a time-varying value which is a random number and a timestamp which is sent by the server. The gateway applies a function to the challenge and sends the result, called a response, to the server. The response shows that the gateway knows the secret. This procedure is called a "bidirectional" because it confirms the administrator identity to the gateway and vice versa. This method assumes that the clocks of both sides are synchronized and they also have synchronized and equivalent pseudo random number generators (having the same code functionality, their seeds are equal and generate their outputs at the same time intervals). The challenge/response begin when the administrator sends an encrypted packet contains the username, a generated random number (RND1) and a timestamp (T1). This arrangement proves the identity of the administrator in several aspects:

1. The value of RND1 is already known by the gateway because its pseudorandom number generator is synchronized with that of the server. Only the server can generate this value at that time. The gateway checks the value of RND1 which is the first prove of the administrator identity.
2. The value of T1 is a time stamp (represents the time value in the server side) which is synchronized with the gateway clock. This arrangement prevents reply attack and can be considered as the second prove of the administrator identity.
3. This packet is encrypted using the Advanced Encryption Standard (AES) algorithm. The 128 bit key, which is called Authentication key, is known only by the two sides and is considered as the third prove of the administrator identity.

If the request passed the identity checking procedure, then the gateway accepts the connection and sends a similar packet so that its identity is also proved to the server

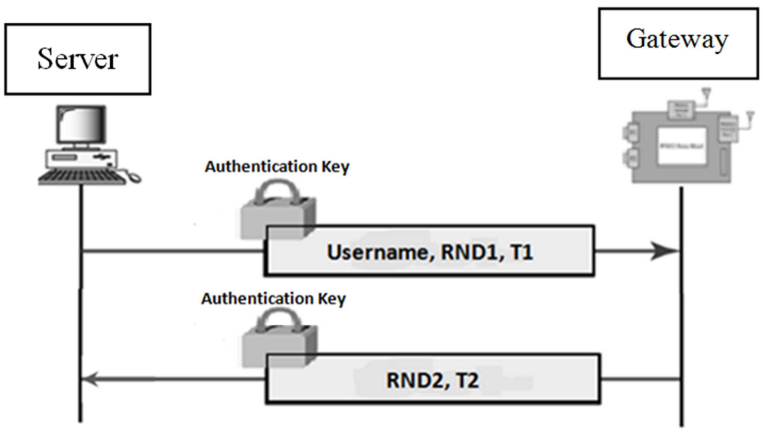


Figure 5. *The suggested bidirectional entity authentication.*

6. Embedded Intrusion Detection System (IDS)

Service availability is an important security issue which means that authorized access of data and other WSN resources is made ready when requested or demanded. This feature could be obtained by protecting the system against different types of attacks using an Intrusion Detection System (IDS). One of the major contributions in this paper is the insertion of Intrusion Detection System (IDS) functionality into the IoT gateway [23]. This was achieved by adopting a light weight, signature based IDS, based on the well-known open source SNORT IDS. *SNORT* is an open source network IDS capable of performing real-time traffic analysis and packet logging on Internet Protocol (IP) networks [22]. *SNORT* can perform protocol investigation and content searching/matching, and can be used to sense a diversity of attacks and probes [23, 24]. *SNORT* uses a simple, lightweight rules description language that is flexible and quite powerful. Snort rules are divided into two sections, the rule header and the rule options [23]. The header contains the rule's action, protocol, source and destination IP addresses including network masks, and the source and destination ports. All options are defined by keywords specifying which fields of the packet should be inspected, such as Time To Live (TTL) and content [24].

Based on the above discussion, intrusion detection can be divided into two actions: packet filtering (or classification based on header fields) and string matching over the packet payload.

Regarding gateway nodes, as they have restricted processing and energy constrains, the insertion of an extra tasks (such as an IDS program) could influence dangerously on its performance, so that, the current design takes these constrains as a main concern and the following procedure was followed:

1. The gateway nodes were loaded with specific rules set (not all rules) which correspond to the most severe attacks at that time. The determination of these rules as "significant" is achieved using IDS sensors (which are specific nodes with complete and up to date SNORT program installed on them) scattered around the network. These IDS sensors observe the network status (from security point of view) and arrange a report of the most frequent attacks at that time. These reports are sent to the Classification and Processing Server for extra processing.
2. Classification and Processing Server (could be the same ISP Server mentioned earlier) accumulates the reports from the IDS sensors and evaluates them to allocate the most risky attack(s) at that time. This server is armed with the classification and processing program which is used to categorize the SNORT rules. After that, the server will broadcast the processed rules set to all wireless gateway nodes that exist in the network. Figure 6a shows the description of the suggested system.
3. In order to maintain the efficiency and performance of the gateway nodes, a new rules processing algorithm is recommended. The core idea of this algorithm can be abstracted throughout implementing the preprocessing part of algorithm in the processing and classification server and only the searching part of algorithm is implemented in the gateway node [23, 24]. The preprocessing part will be sent as an update file from processing and classification server to the gateway nodes, see Figure 6b.

During searching phase, a match with a SNORT rule is determined if it has prefix match with the source and the destination prefixes, exact match with the protocol, and range match with the source port and the destination port [23, 24]. The searching phase in the suggested Tree algorithm is immediately finished without searching the whole trie if an input packet matches a priority rule [23, 24]. This feature will effectively improves the searching performance. Furthermore, the searching proceeds to the left or right in accordance with the sequential inspection of destination address bits starting from the most significant bit. If there is a match with all the fields in a tree, it is considered as “match” and its priority number is remembered. The searching will be stopped immediately in case if it ends with a match with a priority rule or at a leaf while it is always finished at a leaf in other trie-based algorithms.

The software structure of the proposed IoT gatewayequipped with IDS deals with processing headers phase and testing against rules phase, see Figure 6c. In processing headers phase, the IP and Transport layer headers are processed through reading header’s fields sequentially from UBICOM memory. Additionally, during headers processing, checksum will be calculated for the header or for the header and data, to warrant no errors during receiving the packet. If any error happens such as: checksum not equal zero, header length is greater than the total length or other errors, the packet is ignored. After the processing phase is completed, the headers parameters are passed to the testing phase to check them against IDS rules in order to find attacks [23, 24].

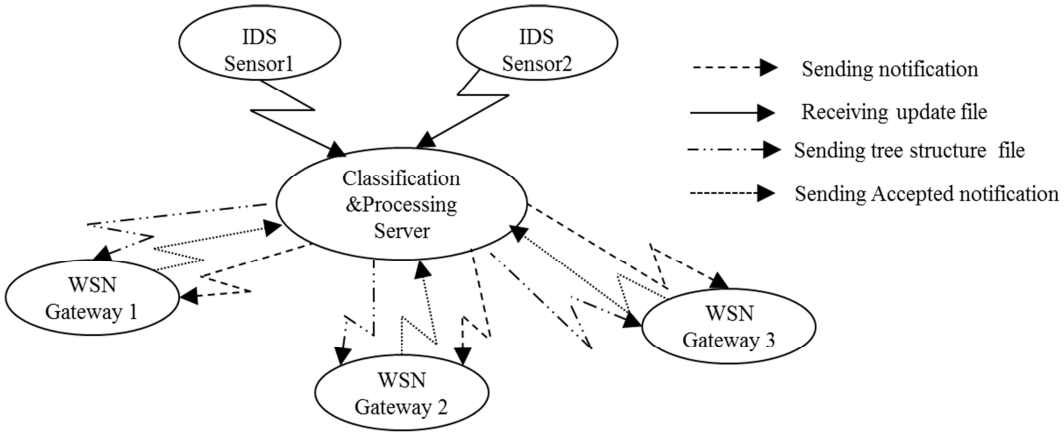
6.1. Results for Packet Filtering (Classification Phase)

The worst case is computed when considering a classifier table filled with completely different rules, without overlapping in the values of the fields. The packet filtering is based on the five headers field (as mentioned earlier), so, it will be searched for these headers fields for each incoming packet.

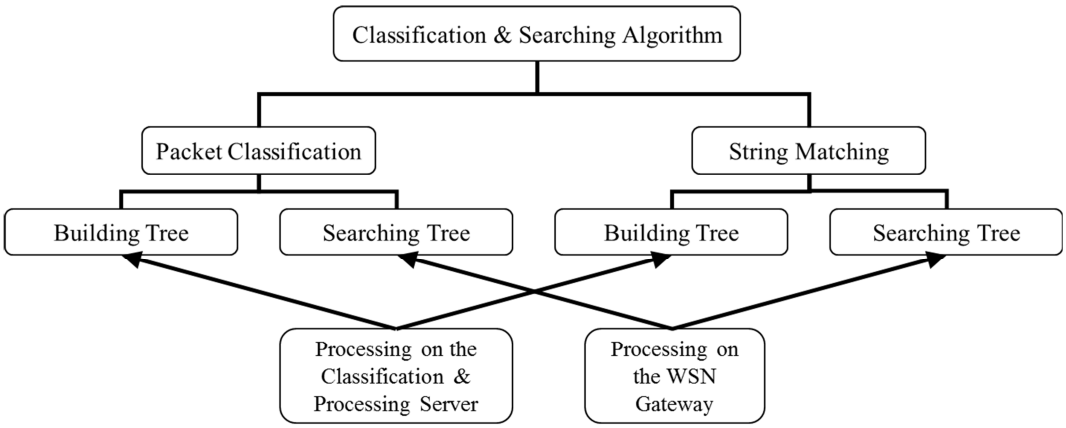
a) Storage Memory Requirements: it represents the amount of the on chip memory (Static Random Access Memory (SRAM) data memory) that is used to store the database of header. Its size depends on the number of rules, the number of header fields that are checked for filtering and the width of each header field in bits. The relationship among these variables is shown in Figure

7a, it can be noted that the memory storage increases when each variable is increased where the graph was sketched when the number of header fields was 5 with a range of number of rule sets.

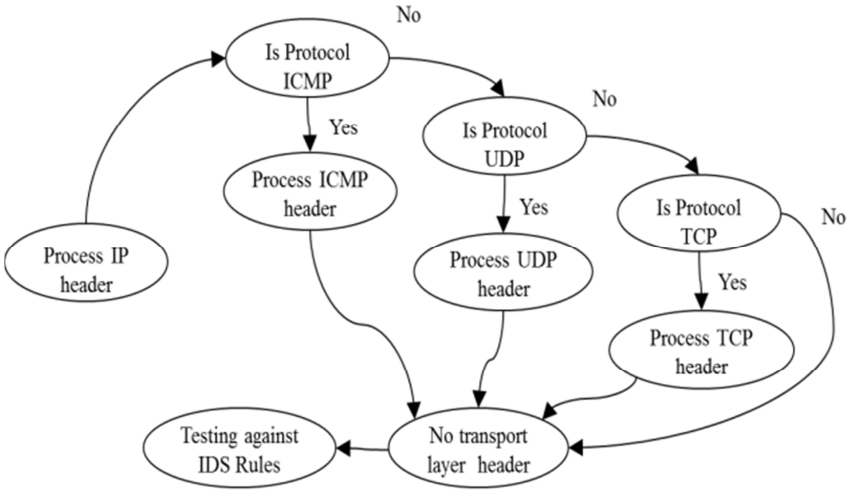
b) Searching Time of Packet Filtering: it represents the time elapsed to find a specific header fields which match the header fields for the input packet. It depends on the number of header fields that are checked for filtering and the width of each header field (in bits). It is noted from Figure 7b that the searching time increases when each variable is increased where, the graph is sketched with the number of header fields was 5 with different sizes of header width.



(a)



(b)



(c)

Figure 6. *The Suggested IDS (a) General Operation (b) The suggested classification and searching algorithm (c) State diagram of the suggested WSN for receiving packet.*

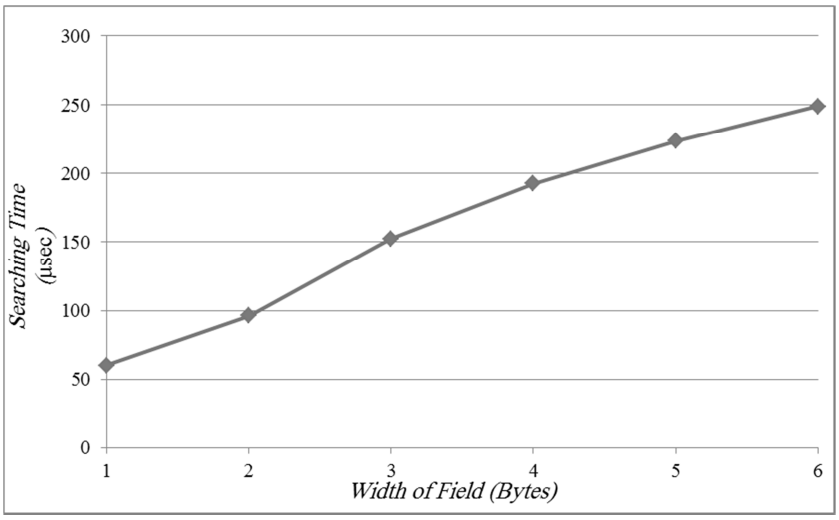
In order to assess the performance of the recommended IDS, numerous tests were implemented. Experimental measurements of the searching algorithms of each phase were performed as two steps. In the first one, results were discussed by computing the memory storage based on the number of rules that can be stored in the memory of the IoT gateway. While in the second one results were discussed by manipulating the searching time to find the specific headers rule or string. Finally, the total response time of the proposed IDS was measured using an experimental network.

6.2. Results for String Matching Phase

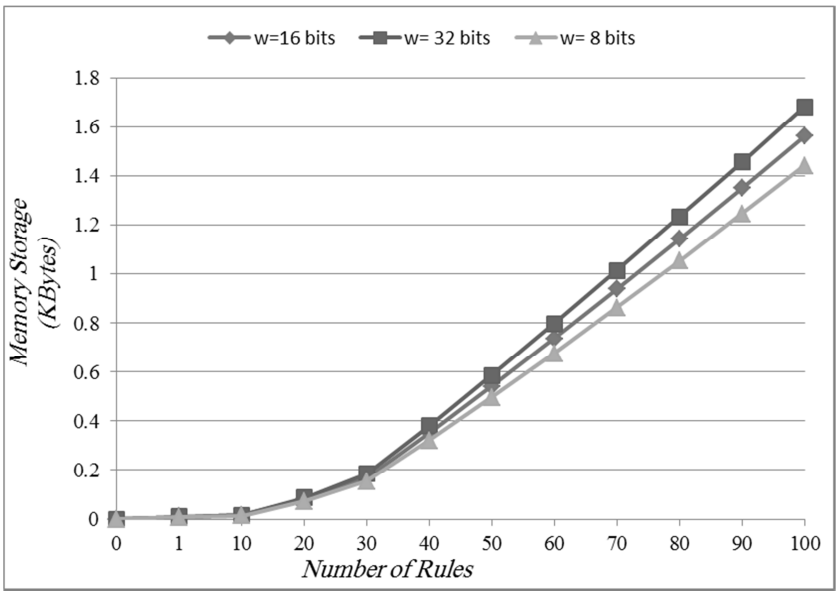
String matching performance depends on the number of matches between the packet payload and the searching arrays; hence, a scenario is considered (as the worst case) in which all patterns exists in the packet payload. After a successful rules matching is founded in the header's packet, the proposed IoT gateway will start performing string matching for the incoming packet.

a) Storage Memory Requirements: the storage requirement is another important metric to evaluate the string matching algorithms, where it represents the on chip memory that stores the searching arrays. It depends on the number of searching arrays (in bytes) and the length of each searching arrays, see Figure 8a.

b) Searching Time of the String Matching: it represents the time consumed to find a specific searching array from a set of searching arrays that matches the pattern at the packet's payload. The results are generated with packet size of 1300 byte with 100 patterns; the results are conducted with pattern length of 15 byte. From Figure 8b, it can be noted that searching time increases when the variable is increased where the graph was sketched with various number of searching arrays.

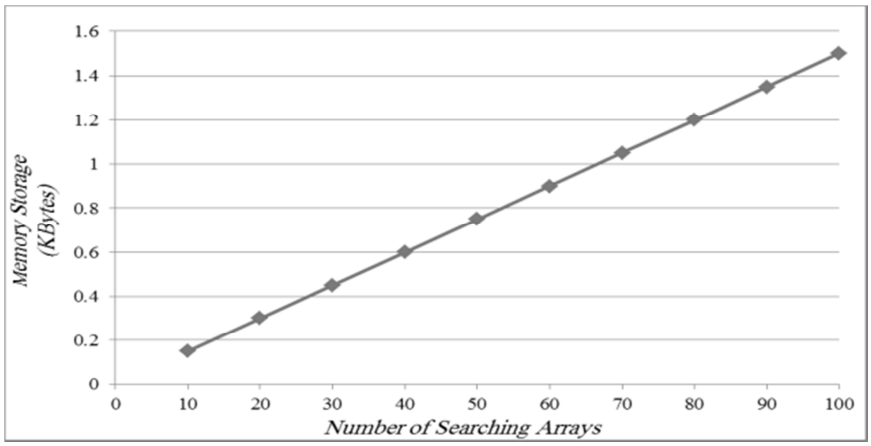


(a)

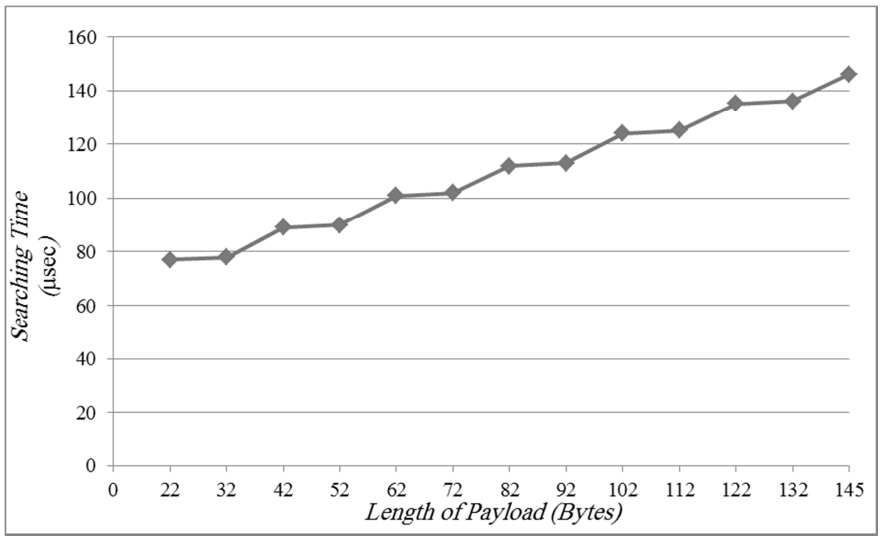


(b)

Figure 7. *Results for Packet Filtering (a) Storage Memory Requirements (b) Searching Time.*



(a)



(b)

Figure 8. *Results for String Matching (a) Storage Memory Requirements (b) Searching Time.*

6.3. IDS Response Time

In order to measure the response time of the proposed IDS, an experimental network was built. Figure 9 shows the components of the experimental environment in the Lab. PC1 is used to emulate the functionality of the ISP server. PC2 has a packet generation program installed, which allows the user to edit and send packets (to emulate different types of attacks) via network interface card. In order to measure the response time of the proposed IDS in (µsec) range, hardware connections between Ubicom I/O pins and real time digital oscilloscope from Tektronix was made.

To find the system response times, the timer which is available on the Ubicom platform was used. This timer counts the maximum number of received packets during one second. Table 1 lists ten types of attacks which were generated using the packet generator tools. These attacks have various rules placed in different locations on the tree, with different values of searching algorithm time parameters, and different packet lengths.

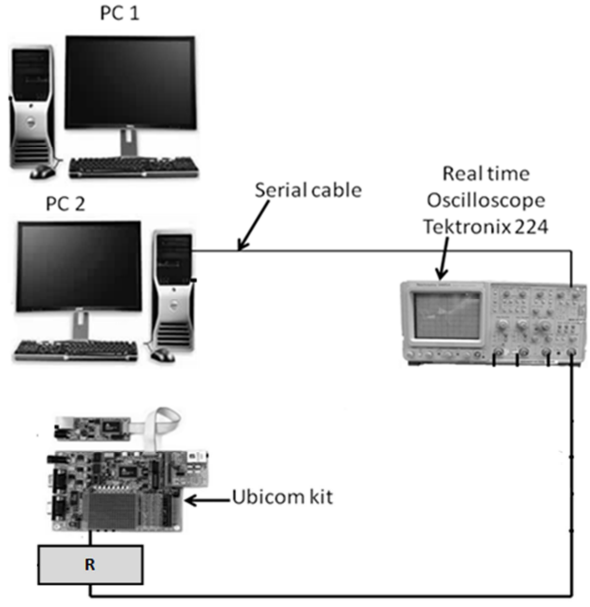


Figure 9. *The Experimental Testbed.*

As noted from Table 1, the searching time of the tree structure can be considered effective in some cases, especially when all header filter parameters of the rule are shared fields with other rules and when the rule is localized in the end of the tree. In this case, the searching time of tree structure is very long, because other signatures are searched before finding the specific signature. On the other hand, if all headers filter parameters for acertain rule are specific fields, the searching time of tree structure is relatively small. Meanwhile, the delay of the searching algorithm increases when the increment in the signatures length and depth. It can be found that Bit Rate ranges between 1.08 and 9.24 Mbps, which represent the effective range of the proposed IDS.

Table 1. *The variant times for overall system.*

Attack Type	Signature Length (Byte)	Depth (Byte)	Packet Length (Byte)	Total Time (µs)
DDOS	--	--	106	36
DOS1	6	32	540	196
DOS2	14	14	790	113
DOS3	--	--	400	33
RESPONSE	8	8	150	281
EXPLOIT1	6	15	350	98
EXPLOIT2	3	145	1200	5032
WEB CLIENT	7	122	700	2096
EXPLOIT3	8	92	1400	510
WEB COLDFUSION	20	52	900	5607

6.4. IoT Gateway Power Consumption

In order to validate the convenience of the suggested IoT gateway from power consumption point of view, several practical tests must be performed using an experimental network as shown earlier in Figure 9. The purpose of performing these experimentsta is to emulate the real IoT environment in which the enhanced IoT gateway will be installed.

The objective of the first experiment is to record the electrical current drained by the gateway according to its different modes of operation: Transmission, Reception, IDLE, CPU full load and SLEEP. Traffic generator PC2 was programmed to send and receive a 1Mbps streamed UDP traffic to and from the IP2022 Ubicom platform. The real time oscilloscope (Tektronix224) was used to measure the drained current from the batteries (according to the different network traffic conditions) by measuring the voltage across a (0.1Ω) resistor, which is proportional to the drained current. Table 2 summarize the settings of this experiment and lists the average values obtained for different data rates.

Table 2. *Network setup&measured current values.*

Experiment duration in each Case (Minute)	5
WLAN NIC	Belkin (a/b/g) Dual-Band WLAN PCMCIA Card
Supply Voltage (v)	3
RF power (W)	1 dBm
WLAN Packet length (Byte)	1500
Packet/sec.	84
Current drained in TX mode (mA)	150 (for IEEE802.11a)
Current drained in RX mode (mA)	120 (for IEEE802.11a)
Current drained in IDLE mode (mA) (WLAN NIC disconnected)	100
Current drained in CPU full load mode (mA) (WLAN NIC disconnected)	150
Current drained in SLEEP mode (mA) (for the Ubicom board only)	1

The purpose of the next experiment is to discover the network activities of a typical IoT infrastructure and hence, the power consumption of the intended Ubicom gateway under realistic road traffic conditions. In order to feed the experimental test bed with truthful values, a simulation model was built using the Network Simulation package. The goal of building this model is to generate a traffic patterns as close as possible to the real situations. The network represents an IoT Ad hoc Network infrastructure of 40 gateway covering (5 Km^2) area of a typical WSN field. The data traffic generated by the gateways (resulting from their interaction with the WSN nodes and other gateways) are forwarded using suitable routing protocol to the IoT server. It was assumed that WSN nodes broadcast their 100 byte status packets each one second [3], while gateways generate their 1000 byte summery report 10 times per minute and forward them to the IoT server [3]. According to the earlier analysis in [23], Optimized Link State Routing (OLSR) gives the best performance compared to other ad hoc routing protocols when working in non-stationary ad hoc topology, so that it was adopted in the simulation model. The OLSR mechanisms are regulated by a set of parameters predefined in the OLSR RFC 3626 [1] which was used in the simulation model. In order to simplify the simulation model, gateways' were assumed to be identical and subjected to the same network traffic conditions.

The different network traffic patterns generated from running the previous simulation model were used to fed the experimental network in order to measure the average drained current as listed in Table 3. These values represent the *baseline* IoT model, i.e., without the intervention of any attack or the functionality of the suggested security center.

Table 3. *Network traffic & average drained current values.*

Average Traffic Sent from each Gateway (kbps)	79
Average Traffic Received from each Gateway (kbps)	401
Total Average Traffic (kbps)	480
Average Drained Current (mA)	104
Battery Life (Hours) for fully charged 2800 mAh AA Battery	27

The purpose of this experiment is to measure the effect of the cooperative signature (Snort based) IDS functionality on the network traffic and hence, gateways' power consumption. The test bed was fed with the simulation model outcomes while changing both the rules update file size (number of rules) and the signatures update interval. The results obtained from performing these tests can be shown in Figure 10. It is clear that increasing the file size while decreasing the update interval creates more network load and hence more power is consumed due to the increment in the transmission/reception operations. It is worth to mention that when using a fully charged 2800 mAh AA battery, gateway can works for 27 hour

under IoT baseline traffic pattern, however, battery life was decreased to 26.5 hour when the update file size was chosen to be 40 Kbyte with 10 Minute update interval (highest extra traffic case). In real world implementation, the file size to be (20 Kbyte) with (30 Minute) update interval is recommended which is a good compromise between gateway invulnerability and its power consumption. Finally, It is worth to mention that an additional gateways' CPU utilization (due to the additional IDSs' tasks) was observed to be ranged between (5%-15%) according to the update file size.

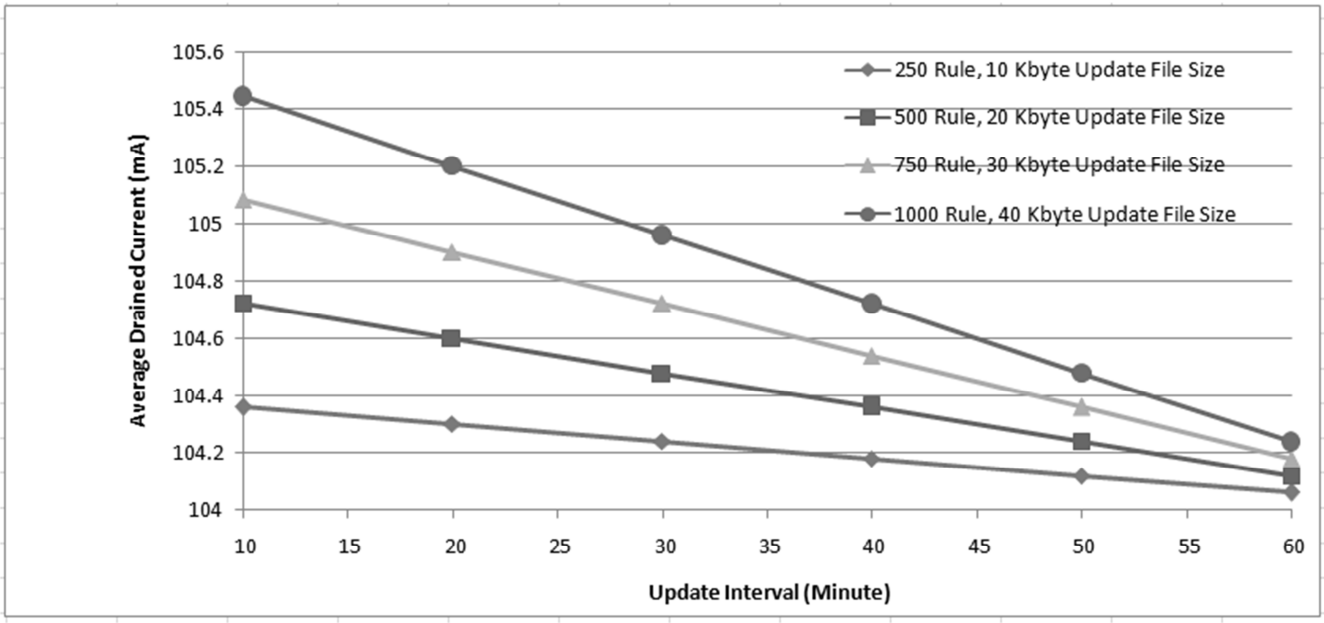


Figure 10. *Effect of varying update file size and update interval on gateway drained current.*

7. System Evaluation & Security Assessment

In the last section of this paper, different evaluation metrics [31, 32] are used to assess the overall performance of the suggested IoT gateway. The system was loaded with 250 different SNORT rules while activating the different security methods suggested in this paper. Table 4 lists the different characteristics of the proposed system in terms of its resources utilization, system performance and security features.

The main remark could be extracted from system resources utilization statistics is that the embedded security center was integrated successfully and efficiently into the IoT gateway platform. It is evident that the suggested security center consumes a reasonable amount of system resources with minimum effect on the gateway original tasks. However, the observed utilization is prone to be variant according to the requirements of the installed IDS strategies.

On the other hand, the design ensures that the insertion of the additional security tasks will not cause a sever degradation in the nodes or network performance and hence, IoT system services (which require delay values to be in the range of tens of milliseconds [1, 2]).

Lastly, regarding IDS security and management features, the proposed security center supports wide range of known attack patterns (e.g., SNORT rules) and it can be developed to detect sophisticated Ad hoc WSN attacks. It is also important to mention that UBICOM platform has a ready to use SNMP client which is very important to perform IoT gateway remote management and reconfiguration tasks.

To complete the picture, an extensive security assessment was made though considering the probable attack vectors and risk sources while suggesting the appropriate countermeasures, see Table 5. It can be concluded that the suggested security center is capable of detecting and defending against a wide range of security attacks in the different network layers which enhances IoT invulnerability against security threats using a pre-managed and transparent fashion.

Table 4. *IoT Gateway Evaluation.*

Hardware Resources Utilization	Additional Memory Utilization	10%
	Extra - Average CPU% Due to Different Gateway Security Functionality	25%
	Extra -Average power Consumption Due to Different Gateway Security Functionality	17%
IoT Infrastructure Performance	Maximal Throughput with Zero Loss	9 Mbps
	Average Induced Traffic Latency (s)	15 ms
	MaximumPacket Processing Rate (Packet/s)	1800
Gateway Security Features	False Positive Ratio	2%
	False Negative Ratio	1.2%
	Depth of System's Detection Capability	(+3500 SNORT Attack Signatures)

	Firewall Interaction	Supported & Integrated
	Router Interaction	Supported & Integrated
	Simple Network Management Protocol (SNMP) Interaction	Supported & Integrated
	Multi-sensor Support (i.e., Cooperative IDS)	Supported & Integrated
	Distributed Management	Supported
	Ease of Configuration	Supported

Table 5. *Security assessment of the suggested security center.*

Attack Type	Attacks' Target	Defense Strategy
Known Attack Patterns (more than 3500 as defined by SNORT IDS developers)	Gateways' hardware, software, services and energy resources Functionality of network, transport and application layers IoT Network Infrastructure	Embedded SNORT (Signature) Based IDS
Denial of Service Attacks	Gateways' hardware, software, services, communication and energy resources IoT Network Infrastructure	Embedded SNORT (Signature) Based IDS
Distributed Denial of Service Attacks	Gateways' hardware, software, services, communication and energy resources IoT Network Infrastructure	Embedded SNORT (Signature) Based IDS
Sybil attack	Gateway services	Entity Authentication
Timing Attack	Gateway services	Embedded SNORT (Signature) Based IDS
Application Attack	Gateway services	Embedded SNORT (Signature) Based IDS
Administrative Impersonation Attack	Gateway data & services IoT Functionality	Entity and Message Authentication
Monitoring Attack	Gateway data & services	Packet Encryption
Illegal Access Attack	Gateway data & services	Entity Authentication
Data Sniffing and modification	Gateway data & services IoT Functionality	Message Authentication and Integrity

8. Conclusions

The Integration between Wireless Sensor Networks and the Internet is another face of Internet of Things (IoT) revolution. However, this combination adds additional security challenges which need to be tackled before gaining IoT rewards. Connecting WSN to the Internet could be accomplished in many ways and *Gateway based* is one of them. In this paper, a novel secured gateway architecture to immunize the path between the gateway and its ISP server is suggested. Compared to other IoT gateway implementations [33-35], it can be seen that the current system enjoys a rich security features with realistic resources utilization. The most important part of the suggested security center is the embedded IDS which strengths the IoT gateway resistance against different types of attacks originates from different threats sources. The experimental tests prove the usefulness of this proposal in terms of the enhanced security, reasonable resource utilization and an adequate system performance.

References

[1] A. Castellani, " Architecture and Protocols for the Internet of Things: A Case Study", In Proceedings of First International Workshop on the Web of Things (WoT), 2010.

[2] Z. Shelby, "Embedded Web Services", IEEE Wireless Communications, pp. 52-57, December 2010.

[3] L. Atzori, "The Internet of Things: A survey", Computer Networks Journal, Vol. 4, No. 5, pp. 2787-2805, October 2010.

[4] C. P. Mayer, "Security and Privacy Challenges in the Internet of Things", KiVS Workshop on Global Sensor Network, 2009.

[5] R. Roman, J. Lopez, "Integrating Wireless Sensor Networks and the Internet: a Security Analysis", Internet Research Journal, Vol. 19, no. 2, pp. 246-259, 2009.

[6] A. Modirkhazeni, N. Ithnin and Q. Ibrahim, "Empirical Study on Secure Routing Protocols in Wireless Sensor Networks," International Journal of Advancements in Computing Technology, Vol. 2, No. 5, 2010, pp. 25-41.

[7] K. Zhang, C. Wang, and C. Wang, "A secure routing protocol for cluster-based wireless sensor networks using group key management," In Proc. 4th IEEE International conference on Wireless Communications, Networking and Mobile Computing (WiCOM'08), 2008, pp. 1-5.

[8] T. Lalitha and R. Umarani, "Energy efficient Cluster Based Key Management Technique for Wireless Sensor Network," International Journal of Advances in Engineering & Technology (IJAET), Vol. 3 No. 1, 2012, pp. 186-190.

[9] S. Sharma and S. Jena, "A Survey on Secure Hierarchical Routing Protocols in Wireless Sensor Networks," In Proceedings of International Conference on Communication, Computing (ICCCS'11), 2011, pp. 146-151.

[10] R. Doriguzzi, G. Russello, and E. Salvadori, "Tinykey: A light-weight architecture for wireless sensor networks securing real-world applications," in Wireless On-Demand Network Systems and Services (WONS) International Conference, 2011.

[11] L. Casado and P. Tsigas, "Contikisec: A secure network layer for wireless sensor networks under the contiki operating system," in Proceedings of the 14th Nordic Conference on Secure IT Systems, Berlin, 2009.

[12] F. Büsching, U. Kulau and L. Wolf, "Architecture and Evaluation of INGA - An Inexpensive Node for General Applications," in IEEE Sensors Conference, Taiwan, 2012, pp. 842-845.

[13] Q. Ali, N. Najim, “Security for Wireless Sensor Network”, Global Journal of Engineering Science and Researches, Vol. 1, No. 3, 2014.

[14] I. Gawdan, C. Chow, T. Zia, Q. Sarhan, “A Novel Secure Key Management for Hierarchical Wireless Sensor Networks,” In Proceeding of 2011 Third Conference on Computational Intelligence, Modeling and Simulation (CIMSiM), 2011, pp. 312-316.

[15] F. Kausar, F. Masood and S. Hussain, “An Authenticated Key Management Scheme for Hierarchical Wireless Sensor Networks,” In Advances in Communication Systems and Electrical Engineering, Lecture Notes in Electrical Engineering, Vol. 4, No. 2, 2008, pp. 85-98.

[16] B. Vaidya, J. Rodrigues, J. Park, "User authentication schemes with pseudonymity for ubiquitous sensor network", in NGN. Int. J. Commun. Syst. 2010, Vol. 23, No 1, pp 1201-1222.

[17] M. Das, "Two-factor user authentication in wireless sensor networks", IEEE Trans. Wireless Comm. 2009, vol. 8, No. 2, pp 1086-1090.

[18] D. He, Y. Gao, S. Chan, C. Chen, J. Bu, “An enhanced two-factor user authentication scheme in wireless sensor networks”, Int. J. Ad-Hoc Sensor Wireless Network., Vol. 2, No. 3, 2010, pp 1-11.

[19] A. Giannetsos, “Intrusion Detection in Wireless Sensor Networks”, Master thesis, Mellon University, April 8, 2008.

[20] V. C. Manju., “Study of Security Issues in Wireless Sensor Network”, International Journal of Engineering Science and Technology (IJEST), Vol. 3, No. 10, October 2011.

[21] S. Hichem, F. Mohamed, “Novel Hybrid Intrusion Detection System for Clustered Wireless Sensor Network”, International Journal of Network Security & Its Applications (IJNSA), Vol. 3, No. 4, July 2011.

[22] S. Andriy, F. Lukas, M. Vaclav, “Neighbor-Based Intrusion Detection for Wireless Sensor Networks”, Faculty of Informatics Masaryk University (FI MU), Report Series FIMU, FIMU May 2010.

[23] Q. Ali, S. Lazim, "Design and Implementation of an Embedded Intrusion Detection System (IDS) for Wireless Applications", IET Information Security Journal, Vol. 6, Issue 3, 2012.

[24] Q. Ali, S. Lazim, E. Fathi, “Securing Wireless Sensor Network (WSN) Using Embedded Intrusion Detection Systems”, IJEEE Journal, Vol. 8 No. 1, 2012.

[25] Z. Handong, L. Zhu, "Internet of Things: Key technology, architecture and challenging problems", 2011 IEEE International Conference on Computer Science and Automation Engineering (CSAE), vol. 4, no. 4, pp. 507, 512, 10-12 June 2011.

[26] I. Gronbaek, "Architecture for the Internet of Things (IoT): API and Interconnect”, SENSORCOMM '08. Second International Conference on Sensor Technologies and Applications, pp. 802, 807, 25-31 Aug. 2008.

[27] H. Chen; X. Jia; H. Li, "A brief introduction to IoT gateway", IET International Conference on Communication Technology and Application (ICCTA 2011), pp. 610, 613, 14-16 Oct. 2011.

[28] Q. Zhu; R. Wang, Q. Chen, Y. Liu; W. Qin, "IOT Gateway: Bridging Wireless Sensor Networks into Internet of Things," 8th IEEE/IFIP International Conference on Embedded and Ubiquitous Computing (EUC), pp. 347, 352, 11-13 Dec. 2010.

[29] L. Costantino, N. Buonaccorsi, C. Cicconetti, R. Mambrini, "Performance analysis of an LTE gateway for the IoT," 2012 IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM), pp. 1, 6, 25-28 June 2012.

[30] Q. Ali: ‘Event driven duty cycling: an efficient power management scheme for a solar-energy harvested road side unit, IET Electrical Systems in Transportation, 2016, Available online at: www.ietdl.org, 2016.

[31] Q. Ali: ‘Enhanced power management scheme for embedded road side units’, IET Computers & Digital Techniques, 2016, 10 (4), pp. 174-185.

[32] Q. Ali, Realization of a Self Powered Road Side Unit Using Network Processor Technology, Recent Patents on Computer Science, 2017, Vol. 10, No. 4.

[33] MXE-200i, Fanless Embedded Computer-IoT Gateway, ADLINK Technology, Available at: www.adlinktech.com.

[34] NIO 100, Intel® Quark Processor X1021 IoT Gateway System, available at: www.nexcom.com.

[35] DKxxxIoT Gateway Series, Available at: www.Intel.com/iotgateways.

Biography

Qutaiba Ibrahim Ali: Was born In Mosul city/Iraq in 1974. He acquired his BSC and MSC (with honor) in Electrical Engineering in 1996 and 1999. He obtained PHD in computer Engineering (with honor) in 2006. Since 2000, he joined Mosul University/Iraq as a faculty member and still there. His research interests include: network simulation & modeling, real time and embedded systems. Dr. Qutaiba published 5 international books and more than 85 papers (some of them are ISI indexed Journals) in his fields of interest.