GLOBAL JOURNAL OF ENGINEERING SCIENCE AND RESEARCHES

SECURITY FOR WIRELESS SENSOR NETWORK

Dr. Qutaiba I. Ali [1], Nada I.Najim [*2]
Computer Engineering Department, College of Engineering, University of Mosul, Iraq
nadaalmaaroof@gmail.com

ABSTRACT

The integration between wireless Sensor Networks (WSN) and their server by the base station (BS) is an important step in a variety of WSN applications. Adding different security methods is an important and essential procedure to enhance the operation and safety of such integration. This paper focused on the design and implementation challenges to localize an embedded security center into base station nodes which connects WSN to the labview based automation system (server). The suggested base station security center consists of two ciphering methods (AES & RC4) to provide data encryption to the whole path from the WSN nodes to the server, an HMAC function to provide message integrity and authentication between the base station and the server, a keys generation module, and a firewall. Our design takes into account the "embedded" nature of the base station (UBICOM IP2022 network processor chip in our case) and their limited resources and suggests different methods and protocols to achieve its goals. The obtained results prove the possibility to insert firewall functionality in the system with minimum effect of its normal operation.

Keywords: Wireless Sensor Network security, automation system, firewall, Base station, Network Processor.

I. INTRODUCTION

Wireless sensor networks (WSN) can be used in industrial automation system for multiple purposes, such as monitoring synchronous or asynchronous events that require periodic data collection or detecting exceptional events, respectively [1]. For example, vibration, heat or thermal sensors can be deployed in proximity of machines to monitor their health. The analysis of the measured parameters can allow the detection of abnormal operating conditions and aids therefore in preventing potential machine failure. In addition to machine monitoring, WSNs can be deployed to measure basic physical quantities such as pressure, temperature, flow or more complex events such as process quality or automotive performance in industrial environments [1]. After an event of interest occurs, one of the surrounding sensor nodes can detect it, generate a report, and transmit the report to a base stations (BS) through multi hop wireless links[2], see Figure(1).

Security is a basic requirement and essential in the design of wireless sensor networks in order ensure the safety of operations, and the confidentiality of sensitive data. Security of WSN is a big challenge due to its limited resources such as energy,

Power supplies, small memory, computation and communication capabilities. This is the reason that traditional security techniques cannot be applied on sensor networks, indirectly rising the need to make sensor network economically feasible. There were many papers that consider some of the most significant WSN security problems.

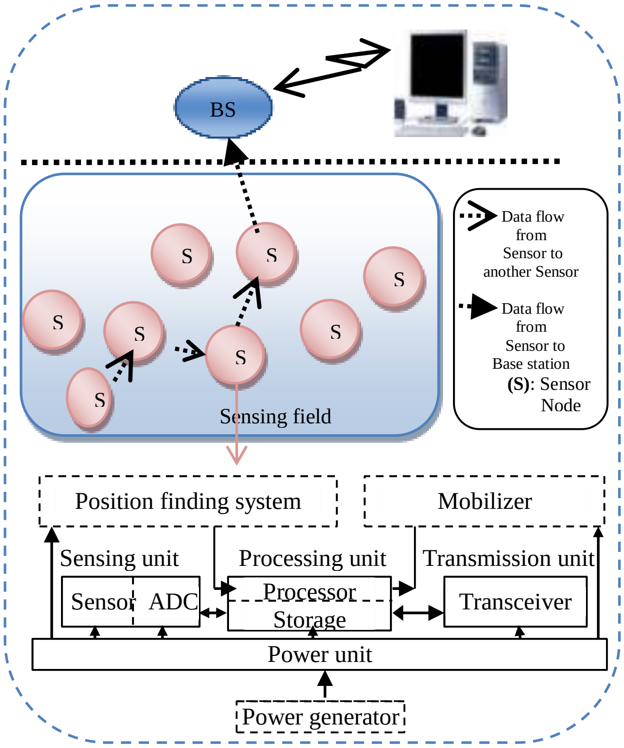


Figure (1): A typical scenario of a Wireless Sensor Network (WSN) and the components of a sensor node

Most research on security in sensor networks has focused on prevention techniques, such as cryptography [3-4], key management [5-6] and authentication techniques [7-8]. 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 [9-10]. Recently, many papers are presented an overview of the different applications of the wireless sensor networks and security related issues due to the nature of these applications [11-12]. In this paper, a complete security frame work was added to the automation system implemented by using WSN 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 system architecture. Section 3 explains the ciphering module of the suggested system. Section 4 explains the keys generation module of the suggested system. Section 5 includes the hardware implementation of WSN base station armed with firewall on UBICOM platform with the results. Section 6 contains an overall system evaluation and finally, section 7 provides conclusions.

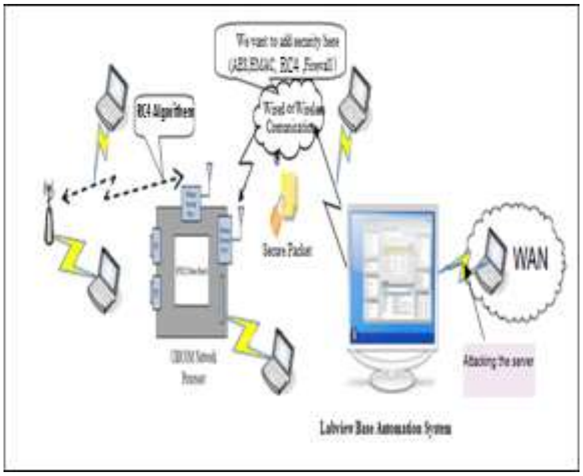
II. THE SUGGESTED SYSTEM ARCHITECTURE

The proposed system is composed of three parts. See Figure (2):

- The Server: A central host computer (sometimes called Human Machine Interface (HMI)).
- The Base station: which is an always-on facility that handles translation and buffering between the server and the WSN motes and acts as a bridge between them.
- The Sensor motes: where different nodes are connecting together to form the wireless sensor network.

In this system, sensor nodes are sensing various signals and sending them to the labview based automation system (HMI) through base station by using wireless or wired connection, this connection was protected using various security methods. An experimental prototype model of the base station was built using Ubicom IP2022 network processer platform which provides a fully integrated platform: real Time Operating System (RTOS), UDP 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 [13].

UBICOM's hardware includes the following components as shown in Figure (3) [13].


We want to add security here (AES,HMAC, RC4 ,Firewall)
RC4 Algorithm
Wired or Wireless Communication
Secure Packet
UBICOM Network Processor
WAN
Attacking the server
Labview Base Automation System

- 64KB (32K x 16) Flash program memory
- 16KB (8K x 16) SRAM data/program memory
- 4KB (4K x 8) SRAM data memory
- Can be provided with external flash memory (2M x 8)
- Two SerDes communication blocks supporting common PHYs (Ethernet, USB, UARTs, Bluetooth, Wireless IEEE 802.11, and so on.) and bridging applications
- 120MHz RISC processor
- Supports software implementation of traditional hardware functions
- In-system reprogrammable and run time self-programmable

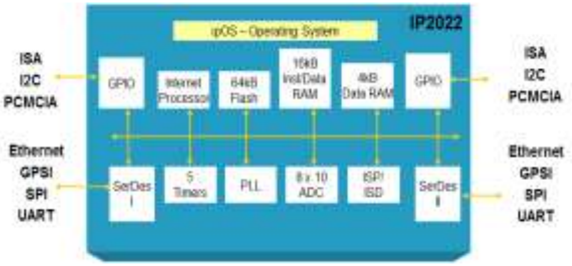


Figure. (3) IP2022 Block Diagram

The basic components for UBICOM's complete development environment can separate to three essential components [13]:

1) UBICOM's Software Development Kit (SDK): UBICOM supports many software packages, like ipOS™, ipStack™, ipHAL™, ipModule™ and etc.
2) Red Hat GNUPro tools which consist of GCC ANSI C compiler, Assembler, Linker, and GNU debugger.
3) UBICOM's Configuration Tool Integrated tool to support rapid development efforts.

UBICOM's Unity Integrated Development Environment (IDE) contains Editor, project manager, graphic user interface to GNU debugger, device programmer.

From security point of view, the attacker can hit the system in many ways and locations. For example, the server could be attacked directly and the link between the server and the base station is also vulnerable to different types of attacks as shown in Figure (2). While the server could be protected using traditional security methods, the base station still needs to be secured against different types of attacks. As mentioned earlier, the base station node acts as a bridge between the WSN motes and the server, therefore it needs to be protected in both directions. In order to secure a message, confidentiality, integrity and message authentication must be added. Also, an efficient firewall is needed to protect the base station against the different types of attacks. Finally, a suitable and secured keys exchange procedure is required to transfer the keys of the different security algorithms. Building on the above, we suggest that the base station architecture must be modified to include various security methods to construct what we called a base station security center, see Figure (4). The suggested base station security center consists of two ciphering methods to provide data encryption to the whole path from the WSN nodes to the server, an HMAC function to provide message integrity and authentication between the base station and the server.

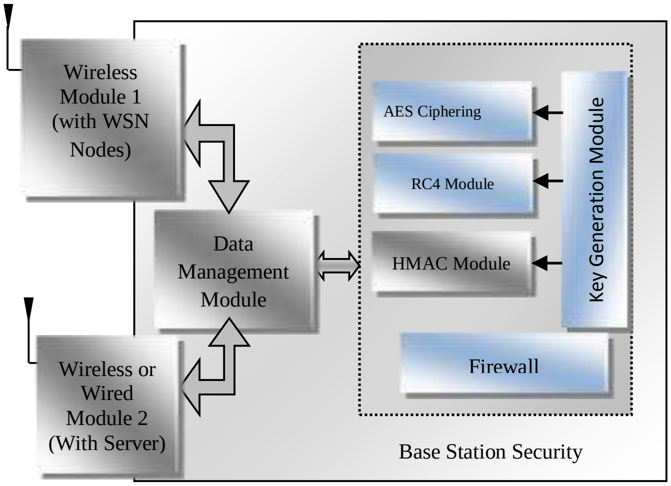


Figure (4): The Suggested Base station Architecture

III.THE CIPHERING MODULES

Our effort to build the ciphering engines of the base stations' security center in Ubicom's environment could be explained as follows:

A. AES algorithm (Advanced Encryption Standard)

Which is chosen to provide the privacy or confidentiality mechanisms to protect the transmitted packets between the base station and the server against unauthorized reading. The AES algorithm is capable of using cryptographic keys of 128, 192, and 256 bits to encrypt and decrypt data in blocks of 128 bits [14]. UBICOM platform supports key size of 128 bits only, so that this length of key will be used. The functions that are used to execute the AES algorithm in order to encrypt the transmitted packet file are as follow:

- The function that expands the AES key into the round keys array which must be passed to the encryption and decryption functions, not the key itself, is as follows:

void aes_compute_round_keys(aes_block k, aes_round_keysrk);
aes_block k: *a pointer to the 16 byte AES key.*
aes_round_keysrk: *a pointer to a 176 byte array that will contain the round keys.*

void aes_decrypt(aes_block a, aes_round_keysrk);
aes_block a: *a pointer to an array of 16 bytes of ciphertext to decrypt.*
aes_round_keysrk: *a pointer to a 176 byte array that contains the round keys.*

- The following function encrypts the 16 bytes of plaintext pointed to by a, leaving the resulting ciphertext in the same array. The round key array must be derived from the 128 bit key using the first function.

- The following function decrypts the 16 bytes of ciphertext pointed to by ***a***, leaving the resulting plaintext in the same array. The round key array must be derived from the 128 bit key using the first function. The speed and space requirements of this function can be adjusted using the configuration tool of ipAES package that is specified for this algorithm on UBICOM platform. Figure (5a) shows the time variation of AES Encryption & Decryption delay within the Ubicom platform with respect to packet size (payload) variation at transport layer with UDP protocol.

B. SHA512

SHA512 is one member of a family of cryptographic hash functions that together are known as SHA-2. The basic computation for the algorithm takes as input a block of input data that is 1024 bits (128 bytes) and a state vector that is 512 bits (64 bytes) in size, and it produces a modified state vector. It is a follow-on to the earlier hash algorithms MD5 and SHA-1, and it is becoming increasingly important for secure internet traffic and other authentication problems. The algorithm operates on 64-bit QWORDs, so the state is viewed as 8 QWORDs (commonly called A…H) and the input data is viewed as 16 QWORDs. The algorithm consists of two steps. The first step is a "message scheduler" that takes the input 16 QWORDs and computes 64 new QWORDs. Together with the original 16 QWORDs, these form a vector of 80 QWORDs that is the input to the second step. This second step consists of 80 "rounds" where the form of the calculations in each round is the same. Each round takes as input the 8 state QWORDs, the corresponding input QWORD (after scheduling), and a round-specific constant, generating updated state QWORDS. After all rounds have executed, the resulting state vector is added to the original state vector, and this results in the new state vector. If the input consists of multiple blocks, this process is repeated for each block [15].

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 Purpose. The function was used to initializes an encryption context that will be used by the encryption functions at Ubicom platform was shown below. During context initialization we take the key details provided to determine the initial encryption state.

Void aes_encrypt(aes_block a, aes_round_keysrk);
aes_block a*: a pointer to an array of 16 bytes of plaintext to encrypt.*
aes_round_keysrk*: a pointer to a 176 byte array that contains the round keys.*

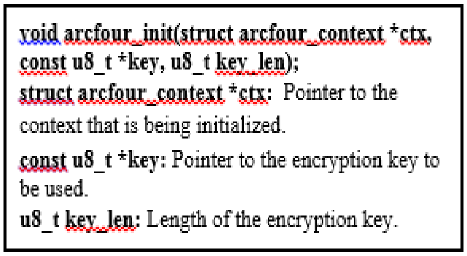
hash function such as SHA-l or SHA-2. HMAC creates a nested MAC by applying a keyless hash function to the concatenation of the message and an AES symmetric key [16]... The SHA512 function that is executed to computes the HMAC-SHA2 message digest of the 128 bytes data in Ubicom platform is follows:

void hmac_sha2_create_digest(u64_t *digest, u8_t *input, addr_tinput_len, u8_t *key, addr_tkey_len);
u64_t *digest: Pointer to the block of data to which the digest will be written.
u8_t *input: Pointer to the block of data over which the digest will be calculated.
addr_t input_len: Amount of data to be used to generate the digest value.
u8_t *key: Pointer to the key to be used in generating the digest.
addr_t key_len: Length of the key in bytes.

Figure (5b) shows the time variation of HMAC delay within the Ubicom platform with respect to packet size (payload) variation at transport layer with UDP protocol.

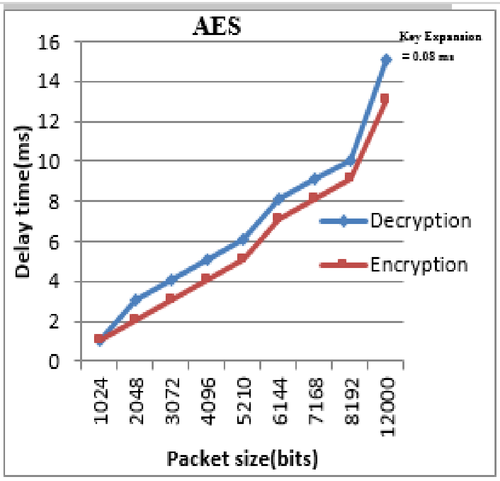
C. RC4 Module:

RC4 is a stream cipher widely used in many applications today and in wireless networks. With a unique key, a stream of pseudo-random numbers is generated. Then, the encryption of data using RC4 is simply based on XORing the pseudo-random numbers from the stream with the data [17]. RC4 is known to be fast and efficient, so that, we suggest to use 128 bit RC4 (as a light weight and fast stream ciphering algorithm) to provide a reasonable level of privacy (with minimum load on the motes processers) to protect the transmitted packets between the base station and its associated motes. In order to strength RC4 functionality, its ciphering key was changed rapidly every 30 Minute using the keys generation module mentioned later. Ubicom platform has ready to use RC4 software module which was used for this

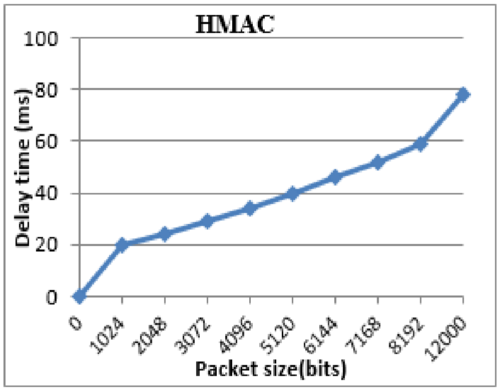


void arcfour_init(struct arcfour_context *ctx, const u8_t *key, u8_t key_len);
struct arcfour_context *ctx: Pointer to the context that is being initialized.
const u8_t *key: Pointer to the encryption key to be used.
u8_t key_len: Length of the encryption key.

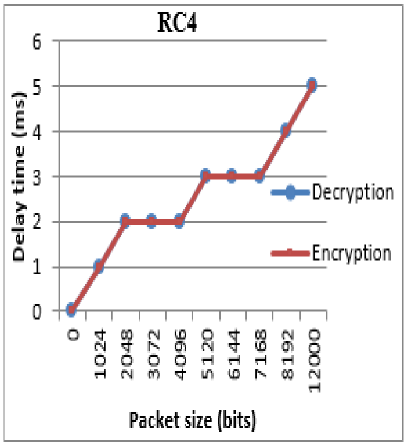
Figure (5c) shows the time variation of RC4 Encryption & Decryption delay within the Ubicom platform with respect to packet size (payload) variation at transport layer with UDP protocol. Figure (6) shows the security techniques which are executed at update sever and UBICOM Platform.



(a)

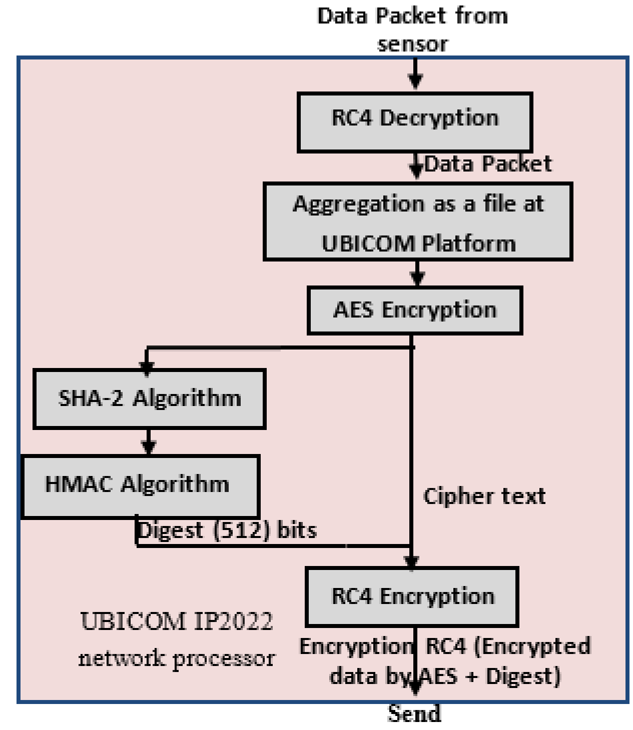


(b)

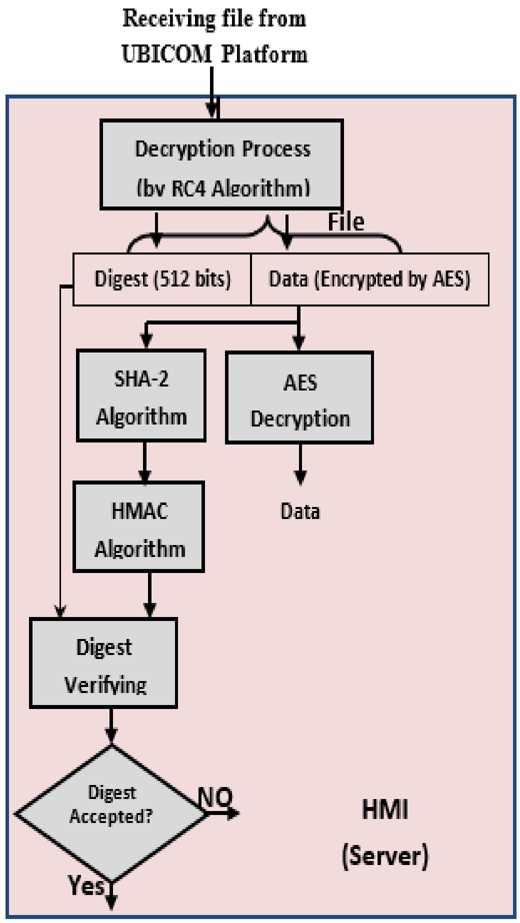


(c)

Figure (5): Encryption & Decryption Delay vs. Packet Length at transport layer: (a) AES (b) HMAC (c) RC4



(a)



(b)

Figure (6) Security Operations at (a) transmitter unit and (b) receiver unit

IV. KEYS GENERATION & DISTRIBUTION MODULE

The above security algorithms need different keys in order to perform their functionality. Also, these keys must be changed at regular time intervals to harden the task of the eavesdroppers who attempt to break the ciphering algorithm by discovering its secured key. It is obvious that keys exchanging among the server, the base station and the WSN nodes is essential to perform keys update process. In this paper we suggest a new method to perform keys updating procedure without exchanging any piece of data. Our keys updating method suppose 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 equal and generate their outputs at the same time intervals, see Figure(7). In order to perform this keys exchanging method, the administrator must first configure the base station 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 base station). Then, the administrator must also determine the keys update interval, e.g., every one hour. We used the Ubicoms' function RND ( ) as the pseudorandom number generators and fed it with the same seed value of the servers' pseudorandom number generator (another pseudorandom number generator has the motes seeds was used to update the RC4 keys). This function was programmed to produce an (128 bit AES key + 128 bit HMAC key + 128 bit RC4 Key = 384bit) output (AES & HMAC & RC4) every one hour in synchronous with the server (PRNG1 & PRNG2 pair). On the other hand, PRNG3 & PRNG4 pair are triggered every 30 Minute to change the 128 bit RC4 Keys.

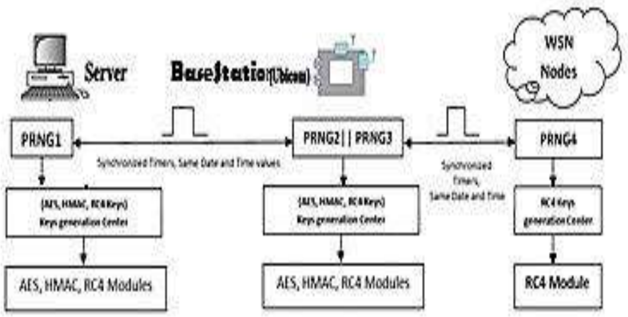


Figure (7): The Suggested Keys Generation Algorithm

V. FIREWALL

The suggested security system was also armed with an embedded firewall at the base station. A data packet that matches a rule in the rule base is allowed to pass through the firewall into the base station; the remaining packets are considered unsafe and hence, are discarded by the firewall. A packet filtering rule specifies the filtering policy for a data packet, i.e. whether to admit the packet into the system or to discard it at the firewall. The operation of a rule is based on the data packet's source and destination IP addresses the source and destination port numbers, and transport protocol of the packet. The Firewall Operation Flowchart at Ubicom platform will perform as shown in Figure (8) below.

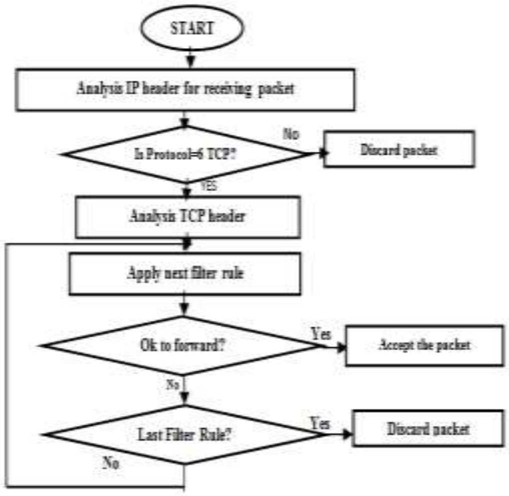


Figure (8): IP2022 Firewall Operation Flowchart

The execution time at the proposed firewall system for analysis IP and TCP headers is 30µs. Figure (6) shows the time variation of packet filtering delay with respect to variation the number of rules.

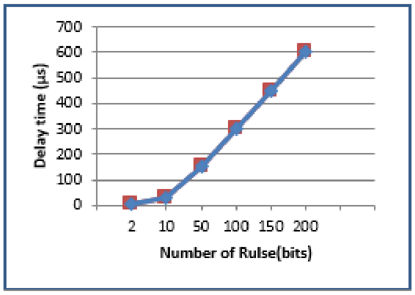


Figure (9): Filtering Delay vs. Number of Rulse

VI. SYSTEM EVALUATION

In order to evaluate the overall performance of the suggested system, two additional tests were performed. The first one is to investigate the memory resources utilization by the different security methods. Table 1 declares that our software codes were compact and consumes reasonable amount (50%) of (80 Kbyte program ram) and (less than 50%) of (4Kbyte of Data ram) at the Ubicom. This resources optimization allows the Ubicoms' platform to perform its base station functionality without reaching memory bottleneck (or fullness) state.

Table 1: Memory Utilization of the Suggested Security Algorithms

Algorithm (8000 bits packet size)	Program ram (Kbyte)	Data ram (Kbyte)
UDP protocol	32.7	1.33
AES	32.5	1.35
HMAC (SHA2)	36.77	1.96
RC4	33.4	1.85
Firewall	Neglected	
Total	40	1.7

The second test was performed to measure the total delay caused by the adoption of the base station security methods. Response time can be defined as the time needed to transfer a packet from Ubicom platform to the server.

The response time was measured between the server and the base station. See figure (10) shows the variation in the delay value with respect to different packets sizes (payload size). It is noted that the delay values could be effective when transferring relatively large packets due to the further processing needed to perform the different security tasks in various locations in the system.

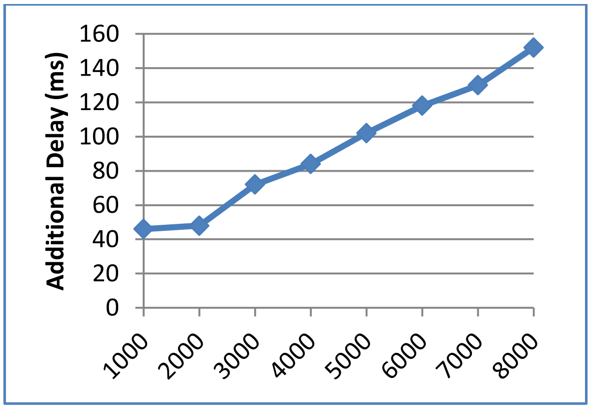


Figure (10): Response Delay Variation vs. Packet Size

VII. CONCLUSIONS

WSNs can be used for multiple purposes, such as monitoring synchronous or asynchronous events that require periodic data collection or detecting exceptional events, respectively. So, the need for security becomes vital. However, in this paper, we suggest the adoption of low cost embedded network processor to support the security. We believe that the current method has several advantages which make it as an attractive, flexible, reasonable resource utilization and strong security solution. This method can work on any UBICOM platform and can be upgraded at any time easily.

VIII. REFERENCES

[1] Christin, Delphine, Parag S. Mogre, and Matthias Hollick. "Survey on wireless sensor network technologies for industrial automation: the security and quality of service perspectives." Future Internet 2.2 (2010): 96-125.

[2] Kavitha T., Sridharan D. "Security Vulnerabilities in Wireless Sensor Networks: A Survey." Journal of Information Assurance and Security 5 (2010) 031-044.

[3] Didla S., Ault A., and Bagchi S., "Optimizing AES for embedded devices and wireless sensor networks," in Proceedings of the 4th International Conference on Testbeds and research infrastructures for the development of networks & communities, Belgium, 2008.

[4] Vivaksha Jariwala, Devesh Jinwala. "A Novel Approach for Secure Data Aggregation in Wireless Sensor Networks." arXiv preprint arXiv: 1203.4698 (2012).

[5] Kausar F., Masood F. and Hussain S... "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, 2008, pp. 85-98.

[6] Gawdan I., Chow C., Zia T., Sarhan Q., "A Novel Secure Key Management for Hierarchical Wireless Sensor Networks," In Proceeding of2011 Third Conference on Computational Intelligence, Modeling and Simulation (CIMSiM), 2011 , pp. 312 - 316.

[7] He, D.; Gao, Y.; Chan, S.; Chen, C.; Bu, J. An enhanced two-factor user authentication scheme in wireless sensor networks', Int. J. Ad-Hoc Sensor Wireless Network. 2010, pp1-11.

[8] Pardeep Kumar, Sang-Gon Lee, Hoon-Jae Lee, "E-SAP: Efficient-Strong Authentication Protocol for Healthcare Applications Using Wireless Medical Sensor Networks," Sensors 2012, 12, 1625-1647; doi: 10.3390/s120201625.

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

[10]Alazemi, Abdulaziz Rashid. "Defending WSNs against jamming attacks."American Journal of Networks and Communications 2.2 (2013): 28-39.

[11]Alcaraz, Cristina, and Javier Lopez. "A security analysis for wireless sensor mesh networks in highly critical systems." Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on 40.4 (2010): 419-428.

[12]Prasanna, S., and Srinivasa Rao. "An Overview of Wireless Sensor Networks Applications and Security." International Journal of Soft Computing and Engineering (IJSCE), ISSN (2012): 2231-2307.

[13]IP2022 Wireless Network Processor Features and Performance Optimized for Network Connectivity IP2022 Data Sheet", UBICOM, Inc., 22 Jan. 2009, Web Site: http//www.ubicom.com.

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

[15]Gallagher, Patrick. "Secure Hash Standard (SHS)." (2008).

[16] Forouzan, Behrouz, Catherine Coombs, and Sophia Chung Fegan. Introduction to data Communications and networking. McGraw-Hill, Inc., 1997.

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