International Research Journal of Advanced Engineering and Science
ISSN (Online): 2455-9024

Watermark Authentication for Secure Data Aggregation in WSN Based on Secure Hash Algorithm and Node Identifier

Dr. Jamal Mohammed Kadhim[1], Enas Faris Yahya[2]
[1, 2]Department of Computer Science, College of Science, AL-Nahrain University, Baghdad, Iraq
Email address:[1]drjamal.cs@gmail.com, [2]enasfaris2007@gmail.com

Abstract— *Data integrity and authentication are an important security challenge in Wireless Sensor Networks (WSN). The main required in WSNs is to check the integrity of the received data at BS. We detected the malicious modifications of the data and also we detected the malicious sensor node. in this work, the delivering sensor information to the neighboring node was assured by the legitimate node and not altered by the attacker or for other reasons. In addition, we propose a method to verify the integrity of the sensor data, which is based on a fragile watermarking scheme (FWS).*

In watermarking, each sensor node embeds a unique watermark to sensor data and BS can verify the data integrity. Node Authentication is one of the important topics in WSNs. In this paper, there is three types of devices have been used in building monitoring and tracking system, these devices are an Ultrasonic sensor, Arduino Uno, NodeMCU esp2866 and RFID RC522.

Keywords— *Data integrity, fragile watermarking scheme, Node authentication, Monitoring and tracking system.*

I. Introduction

There are two types of architecture in WSNs, client-server and peer-to-peer networks. In a peer-to-peer network, there is no central server and each user has his/her own data storage. In a client-server network, the data is stored on a central server [1]. Node authentication enables a sensor node to ensure the identity of the neighbor node it is communicating with [2]. The security requirement is to reject any message has been altered by malicious node this is called verification process.

The authentication process is useful in showing that no tampering has occurred during transmission and the message received from authorized source [3]. Data integrity assures that message contained is not altered while sending data. During the transmission, The attacker may add false data and try to change the message [4]. To ensure the integrity of data in WSN, many researchers proposed watermarking techniques. Most of the existing watermarking schemes generate the watermark by embedding a unique watermark to sensor data and BS can verify the data integrity. Fragile watermarking is used to detect unauthorized alterations in WSN.

To achieve data integrity is done by embedding watermark into original data. Watermarking schemes based on adding unique watermark to the message to check any modifications in the sensor data. Although, these schemes provide the data integrity assurance and complete restoration of original data in case of modification attack [5].

Authentication Scheme is a critical challenge in WSNs. In Wireless Sensor networks the original data can change by editing or injecting packet in the receiver sensor node needs to ensure that the received data is from the correct sensor node. Node authentication enables a sensor node to ensure the identity of the neighbor node it is communicating with [2]. The security requirement is to reject any message has been altered by malicious node this is called verification process. The authentication process is useful in showing that no tampering has occurred during transmission and the message received from authorized source [3]. Integrity means detect unauthorized user and detect any modification on text easily. If the attacker modifies data by unauthorized user, the hash process will be detect any alteration and deletion that occur on the data [6].

To solve this issue, many researchers suggested many solutions to achieve authentication in WSN, Encryption, and the watermark is commonly used in WSN. Information hiding can be useful in authentication mechanism into WSNs. The information hiding is generated and combine with the original data to obtain the final data and send to the neighbor node [2].

The sensor enables to determine the motion of the obstacle by computing the obstacle difference between the trig and echo. The distance was determined by computing the formula given as

$$\text{Distance} = \text{Speed} \times \text{Time}/\ 2 \quad (1)$$

The product of speed was divided by 2 because the total time it took when the waves arrive to the target and return back to the echo pin[7].

II. Literature Review

- Djallel Eddine Boubiche et al., in 2015 [8] introduced lightweight fragile watermarking technique in which any data alteration is detected by the verification node to ensure data integrity, authentication and guarantees secure communication between the aggregation nodes and the base station while saving energy. The watermark is generated first in a fixed space for each received data and then integrated into each packet. The simulation results introduced very low energy consumption over the network and a good level of security.

- Prasad U. Malwatkar et al., in 2015 [9] proposed a reversible authentication based on the reversible watermark, this mechanism can verify the collected data and restore the original data completely. The watermark is generated and embedded in the one node using secure hash value and validate in the other node. Cryptography is the expensive traditional solutions for integrity because of the energy of the sensor node, limited storage space, and computational capacity but the watermarking techniques are much lighter and having no additional overheads. This watermark aims to reduce the delay, No communication overhead with less computation.
- Baowei Wang et al., in 2015 [10] proposed a copyright method to verify the reliability of data and can also determine the location of the node. In this method, we generated a digital watermark according to sensed data, Key, ID. Where is the Key is key, ID is the Node Identifier.

Then we generated the digital watermark according to one – way hash function. A novel digital watermark embedding algorithm of combined the MSRB with LSB.

- C.Manjula et al., in 2016 [11] proposed two categories – Fragile watermark and Reinforced fragile watermark, In this technique each node generates a watermark using the sensed data and the Media Access Control (MAC) address of the sensor node then the result is hashed with Message – Digest Algorithm (MD5) and integrates it in the data packet and transmit it to the neighboring node. Due to the use of watermarking, the authentication and data integrity will be ensured. These Watermark mechanism aims to minimize the delay and power.
- Farid Lalem et al., in 2016 [12] showed that the security based on data encryption is not suitable for WSNs due to the computation resource and energy, and proposed a new technique based on a semi-blind watermark which requires only the original watermark for the extraction mechanism , In this technique each node uses the same locally watermark.

The simulation results provide that the semi-blind watermark can ensure authenticity data and reliability and also reduction of the number of exchanged data in the network. Therefore, this mechanism is able to reduce the energy and traffic between the nodes in the network. This watermark aims to increase the watermark performance against malicious node using an extraction algorithm. Therefore, all malicious node is detected by neighbor nodes.

III. Proposed Node Authentication and Data Integrity Scheme

In this project, the proposed system consists of one wireless sensor node and one laptop have been used, the laptop was used as a sink node. The ultrasonic sensor used to detect the vehicles movement and determined the distance, and then sent the signals to intelligent controller (Arduino Uno) to produce the number of tag ID and sent the tag ID to the NodeMCU to adding the node name, MAC address and calculating the fragile watermark by applying the one-way hash function SHA1 to get a 160 bits fixed digest and then transmits the packet to the sink node by wireless device (NodeMCU ESP8266). The reason for using the hash SHA1 are guarantees data integrity, indirect authentication and required less memory. For these reasons, the SHA1 is more suitable for the sensor. Figure1 shows the block diagram of the system.

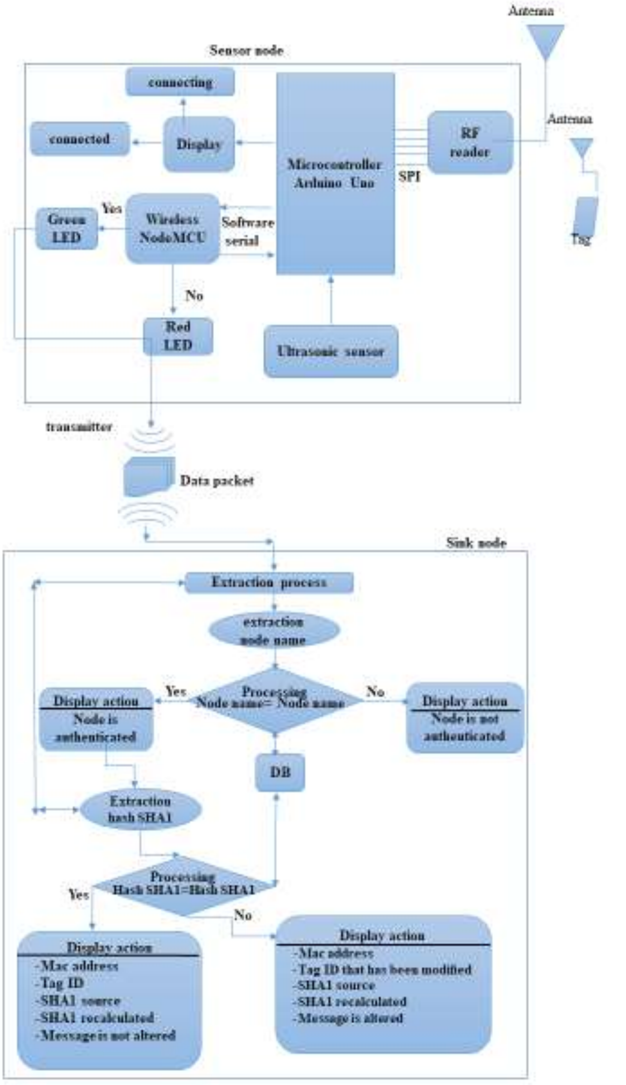
Antenna
Sensor node
connecting
connected
Display
Microcontroller Arduino Uno
SPI
RF reader
Antenna
Tag
Yes
Green LED
Wireless NodeMCU
Software serial
No
Red LED
Ultrasonic sensor
transmitter
Data packet
Sink node
Extraction process
extraction node name
Yes
Processing Node name= Node name
No
Display action Node is authenticated
Display action Node is not autheuticated
DB
Extraction hash SHA1
Processing Hash SHA1=Hash SHA1
Yes
No
Display action -Mac address -Tag ID -SHA1 source -SHA1 recalculated -Message is not altered
Display action -Mac address -Tag ID that has been modified -SHA1 source -SHA1 recalculated -Message is altered

Fig. 1. Shows the block diagram of the system

IV. Proposed Hardware Building of Sensor Node

To build a sensor node, Arduino Uno, NodeMCU, RFID reader, Ultrasonic sensor and liquid crystal display have been linked together to obtain the proposed sensor node. Arduino Uno is used to reading the signals from the sensor and processing the signals to produce the distances to the different object that passes in front it and view the distance value on the serial monitor. The Ultrasonic sensor connected to the Arduino Uno board at pins (3,4) the trig pin is configured as a pin 3 in the Arduino board and the echo pin is configured as pin 4 in the Arduino board, Vcc pin connected to 5V in Arduino board and G pin connected with ground pin in

Arduino board also the blue pin is connected to the Arduino Uno, the blue LED is blinking when the object is passing in front of the sensor.

The RFID reader also connected to the Arduino Uno through the pins from pin 9 to pin 13 at SPI protocol. The RFID reader pins are connected permanently to the SPI pins. G pin connected to ground in Arduino board and Vcc connected to 3.3V in Arduino board. The Arduino Uno is used to read the tag which is received by the RFID reader

The Arduino Uno is connected with the NodeMCU ESP 8266 to transmitting the tag ID to the base station through the wireless device. Arduino Uno is connected with NodeMCU by these pins (6,7) (3,4). Pins (6,7) from the Arduino side and pins (3,4) from the NodeMCU side. These pins are configured as a software serial between them.

Liquid crystal display connect to the Arduino Uno at the pins (A4, A5), used the liquid crystal as a display to illustrate the status of the WiFi network, when the NodeMCU connect to the WiFi the action is display connected, but if the NodeMCU not connected and stay listening to the WiFi the action is display connecting. We also connected two LED (Red and Green) to the NodeMCU. Use Red LED to indicate that the NodeMCU still listening to the WiFi and not connected, but the Green LED to indicate that the NodeMCU connected to the WiFi.

A. The Sensor Nodes Level

The tag ID attached to the object. When the object moved in front of the sensor, the sensor sent the signal to the Arduino Uno. The Arduino Uno still listening to the RFID reader, if detected card the RFID reader sent it to the Arduino Uno, after that the Arduino Uno sent the tag ID to the NodeMCU. The Arduino Uno listen to the WiFi status for the NodeMCU, if the NodeMCU connected to the WiFi network, then the Green LED is blinking and the Arduino Uno printed connected on the liquid crystal display, but if the NodeMCU not connected to the WiFi network, then the Red LED is blinking and the Arduino Uno printed connecting on the liquid crystal display.

If the Nodemcu connected to the WiFi network, then it calculated the hash function by SHA1 and then make the final data packet by adding the node name, MAC address, the tag ID and SHA1 and then sent it to the sink node by NodeMCU ESP8266.

B. The Sink Node Level

When the data packet received from the sensor node, the sink node extracted the node name and compare it with node name stored in the database, if the two node name is equal, then the action is: the node is authenticated. After that the sink node extracted the hash function from received data packet and recalculated the hash function for the received data packet and then compare between two hash function , if the two hash are equal then data packet is not altered and the sink node extracted the information from the data packet and display it in the software window , but if the two hash function are not equal then the data packet is altered and also the sink node extracted the information from the data packet and display it in the software window. But if the two node name is not equal then the action is: the node is not authenticated and reject message.

V. Proposed Software Building of Sensor Node and Sink Node

In the building mechanism, the sensor node applied an XOR function to combine collected data from sensor, MAC address, node name, and one-way hash function. A one-way hash function applied to get 160 bits fixed size digest which represents the generated watermark. In our system, we obtain the result of watermark through this tag information as shown below:

Tag: 19 5D 2D 28

Watermark: 5ae0137ddb900dac7f4ae0564aefbb6812dce65a

In the detection and verification mechanism, when the data packet is received to the sink node, then the sink node extracted the node name from data packet received and compare it with the node name that stored in the database, if the two node name are identical, thus being considered as authenticated after that the sink node recalculates the watermark using the embedding mechanism explained before. If the two watermark values are identical, thus being considered as not altered but if the two watermark values are not identical, thus being considered as altered. But if the two node name is not identical, thus being considered as not authenticated and reject the message.

VI. Hardware and Software Experiment Results

This section covers a reviewed and presented of the experiment results for the proposed system of WSNs. Through this proposed system different parameters and properties will be provided to achieve different goals were summarized through several sections.

A. Rate Test Results for Our Sensor Node

A.1 Computation time for generating and embedding SHA1

Computation time in the sender node is the time required for the execution of some specific task. In our scheme, calculating the time was required for generating and embedding the watermark in the sending side. Table 1 shows the computational time in our proposed procedure for the SHA1 hash function.

TABLE 1. Shows the computational time for our procedure

No.	Average time for our procedure (millisecond)
Case1 (Proposed technique)	1 ms

The computational time was calculated after increasing data packet of watermark generation and embedding process. In four cases as shown in the table below, the data packet was increased by adding 8, 14, 18, 19 bytes as a keyword, as shown in the table 2.

TABLE 2. Shows the computational time of watermark generation and embedding after increasing data packet.

No.	Details	Average time (millisecond)
Case 1	Adding 8 bytes	1 ms
Case 2	Adding 14 bytes	1 ms
Case 3	Adding 18 bytes	2 ms
Case 4	Adding 19 bytes	2 ms

The time required to generating and embedding the SHA1 as shown in table 1 is lower than the time required to generating and embedding the SHA1 when the data packet was increased as shown in table 2. The computational time in our scheme is more acceptable in WSNs and cause less effect on the energy of the sensor node. An increased in the time of generating and embedding the SHA1 was observed from 1 ms to 2 ms after adding 18 bytes as a keyword.

A.2 Computation total time in our sensor node

The total time of operation was calculated from sensing the motion until sending the final message to the BS with SHA1. Second, the total time was calculated from sensing the motion until sending the final message to the BS after adding 8 bytes as a keyword with SHA1. Third, the total time was calculated from sensing the motion until sending the final message to the BS after adding 14 bytes as a keyword with SHA1. And fourth, the total time was calculated from sensing the motion until sending the final message to the BS after adding 19 bytes as a keyword with SHA1, and as shown in the table 3.

TABLE 3. Illustrates the time that message taken in some cases

No.	Average total time	Average time
Case 1	Total time with calculating SHA1	39 ms
Case 2	Total time after adding 8 bytes	49 ms
Case 3	Total time after adding 14 bytes	54 ms
Case 4	Total time after adding 19 bytes	59 ms

A.3 Message length

Message size means that the total number of bits in each message, in our experiments we compare among four cases through the number of bits. In our scheme, the original message is case 1, it consists of 166 bytes after calculating the SHA1 hash function, these bytes divided between node name, MAC address, tag number and SHA1 bits, and as shown in table 4. In cases 2, 3,4 and 5 the message length increasing after adding the keyword to the message, that keyword selected randomly from the message and then calculating the watermark.

TABLE 4. Shows the message size in our experiments

No.	Details	No. of bytes	Time generation SHA1
Case1	After calculating SHA1	166	1 ms
Case2	After adding8 bytes	189	1ms
Case3	After adding14 bytes	195	1 ms
Case4	After adding18 bytes	199	2 ms
Case5	After adding19 bytes	200	2 ms

B. Rate Test Results for Our Sink Node

B.1 Scenario1 for tested the accuracy of an extracted watermark to achieve data integrity in our beneficiary node

After the beneficiary node received messages, beneficiary node extracted a watermark and starts to generate the new watermark and compare it with the received watermark. In our experiment Watermark Accuracy Rates (WAR) was used. WAR was used to analyze the probability of detector making right decisions about data integrity. The values of WAR range between (0,1), and the values of WDR range between (1,0), if the WAR is equal to 0 then the data is not altered, but when the WAR is not 0 then the data is altered, we calculating the WAR by equation 2 as shown below.

$$\text{WAR} = \frac{\text{Number of characters correctly detected}}{\text{Number of watermark characters}} \qquad (2)$$

The Watermark Distortion Rates (WDR) – is used to analyze the probability of detector distortion in the message. To calculate the WDR**,** the equation 3 was used, and as shown below:

$$\text{WDR} = 1\text{- WAR} \qquad (3)$$

Depending on the result of the WAR, the result of detection alteration was obtained or not to check the data integrity for the packet received, and as shown in figure 2, figure 3.

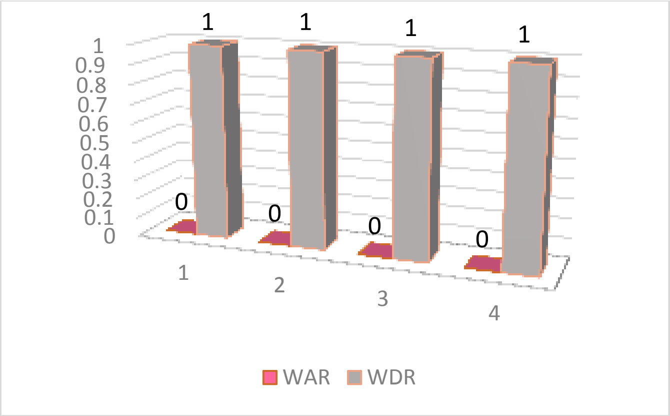

Fig. 2. Shows the data packet was not altered

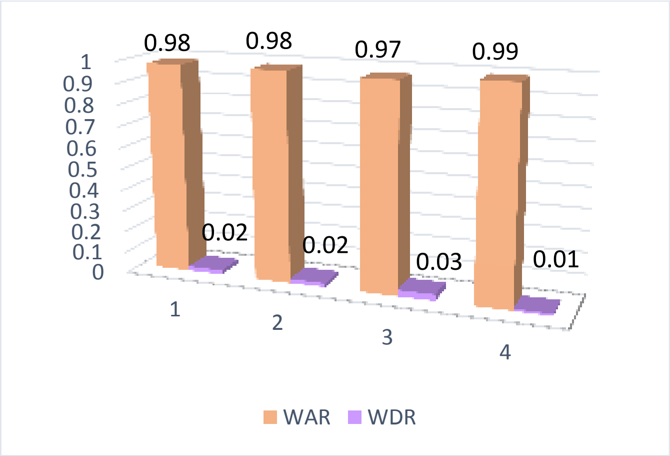

Fig. 3. Shows the data packet was altered

The time that required to extract the watermark from the message and verified their contents to achieve the data integrity, the process was called extraction and verification time.

TABLE 5. Shows the computational time for extraction and verification process to achieve the data is not altered.

No.	Average time to check data integrity
	Not altered
1-	00:00:00:.0109921

B.2 Scenario1 for tested the accuracy of the extracted watermark to achieve node authentication in our sink-node

After the beneficiary node received messages, beneficiary node extracted a node name (node 1) - in our experiment the node name (our node) was named as a node1- and starts to compare the received node name with the node name stored in the database, if the two node name are equal then the node is authenticated. The computational time that required for extraction and verification process to achieve node authentication as shown in the table 6.

TABLE 6. Shows the computational time for extraction and verification process for node authentication

No.	Average time to check node authentication
1-	00:00:00.0161871

B.3 Scenario 2 for tested the accuracy of the extracted watermark to achieve node authentication in the beneficiary node

After the beneficiary node received messages, beneficiary node extracted the node name (node 1) - in our experiment the node name (illegal node) was named as a node 2 and starts to compare the received node name with the node name stored in the database The computational time that required for extraction and verification process to achieve the node is not authenticated as shown in the table 7.

TABLE 7. Shows the computational time for extraction and verification process for non- authentication process

No.	Average time to check non- authentication process
1-	00:00:00:.0137570

VII. Safety Analysis About Experiment Three Type of Attacks

In several attempts, three types of attacks were tried in the message by copy, change and add. Two nodes were used to perform three attacks respectively: Packets Forgery, Selective Forwarding, Packets Tampering. The total experiments for these three attacks are 45 experiments divided among these three types of attacks, and as shown in the table 8.

TABLE 8. Illustrates the three types of attacks

No.	Attacks	No. of experiments	Rates (%)
1	Packets Forgery	20	100
2	Selective Forwarding	10	100
3	Packets Tampering	15	100

In the first type of attacks, the Tag ID was changed and applied 20 experiments to verified the data integrity, in these experiments the rate was 100%. In the second type of attacks, the node name that related to our node was copied and embedded to the false message and then sent the data packet to our sink node and applied 10 experiments to verified node authentication, in these experiments the rate was 100%. In the third type of attacks, the simple modification on the Tag was changed and applied 15 experiments to verified the data integrity and experience how effective the SHA1 hash function and detection of sensitivity to a simple change of data, in these experiments the rate was 100%.

In the figure 4 shows the success of the total experiment in our scheme to achieved node authentication and data integrity at the same time, a compared with the previous experiment we obtained the same rates.

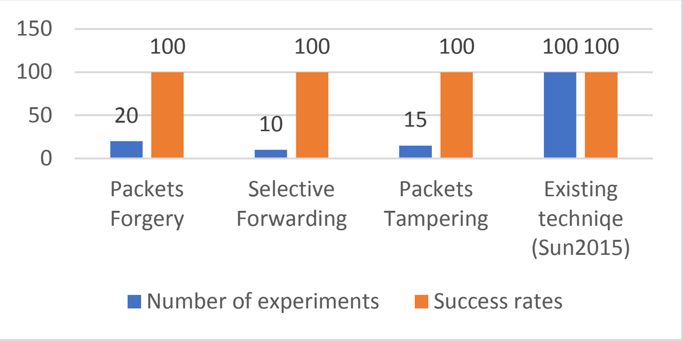

Fig. 4. Shows the success of the total experiment in our scheme

VIII. Conclusion

In this work, specifications were introduced to preserve the data integrity and node authentication in WSNs. A set of requirements has been described to make data transfer process safe and was implemented using lightweight methods to provide speed and good protection to the sensor node**.** The hardware results explained that the lightweight message is better and successful through the speed of generating and embedding than increasing the data packet or adding a keyword. Further works focus on using of lightweight message was successful in our proposed system, the hardware results explained that our proposed process was successful in terms of reducing the process time in the sink node. When increasing the bits of the message, the time that is required to the extraction and verification process was increased.

Acknowledgment

I special thanks to my supervisor Dr. Jamal Mohammed Kadhim who gave me all the supporting, assistance, helpfulness for giving me the major steps to go on to explore the subject. Grateful thanks are due to the Head of Computer Science Department and the staff of the Department at College of Sciences of AL- Nahrain University for their kind attention.

References

[1] Sabah, N., M Kadhim, J., and Dhannoon, B.N.: 'Developing an End-to-End Secure Chat Application', International Journal of Computer Science and Network Security, Vol. 17 , no. 11, november 2017.

[2] Makhoul, J.M.B.-C.G.-A.: 'Two Security Layers for Hierarchical Data Aggregation in Sensor Networks', Int. J. Autonomous and Communication System, Vol. 7 , no. 3, 2016.

[3] Al-Jaber, R.S.A.-A.: 'AFragile Watermarking algorithm for content authentication ', International Journal of Computering and information sciences, Vol. 2, no. 1, April 2004.

[4] Dr.P.M.Pawar, J.C.-. 'Secure Data Aggregation Using Watermarking Technique For Wireless Sensor Network – A Review. ', International Journal of Computer Application, Vol. 7, no. 3, May-June 2017.

[5] Javaid, K.H.-M.S.K.-I.A.-Z.U.A.-A.K.-A.H.-N.: 'A Zero Watermarking Scheme for Data Integrity in Wireless Sensor Networks', 19th International Conference on Network-Based Information Systems (NBiS) 2016.

[6] A K Albermany, S., Amer, D., and Kamal, S.: 'S-RADG: A Stream Cipher RADG Cryptography', International Journal of Scientific & Engineering Research, Vol. 9, no. 3 ,2018.

[7] Mohammed Sahib Mahdi Altaei, L.A.-A., Qudama Khamis: ‘Simulation of Moving Obstacle Avoidance for Auto Guided Land Vehicle’, Al-Nahrain Journal of Science, Vol. 21, no. 4 2018.

[8] Athmani, D.E.B.-S.B.-H.T.-C.-A.-S.K.P.-A.B.-S.: ‘SDAW: secure data aggregation watermarking-based scheme in homogeneous WSNs’, Telecommunication Systems, Vol. 62, no. 2 2015.

[9] Rathod, P.U.M.-S.G.C.-S.R.J.-D.B.C.-P.S.G.: ‘ Survey Paper On Reversible Watermark Authentication Scheme For Wsn ’, International Journal of Technical Research and Applications,Vol. 3, no. 6, 2015.

[10] Sun, B.W.-J.S.-Y.Z.-B.W.-J.S.-Q.D.-X.: ‘A Copyright Protection Method for Wireless Sensor Networks Based on Digital Watermarking’, International Journal of Hybrid Information Technology, Vol. 8, no. 6, 2015.

[11] V.Saranya, C.M.-R.P.-K.R.-. ‘Data Aggregation Integrity in Wireless Sensor Networks Using Cross Layer Watermarking’, International Journal of Research in Electronics, Vol. 3, no. 1, 2016.

[12] Pascu, F.L.-M.A.-A.e.B.-R.E.-L.L.-L.N.-A.: ‘Data Authenticity and Integrity in Wireless Sensor Networks Based on a Watermarking Approach’, Florida Artificial Intelligence Research Society Conference, 2016.