Journal Homepage: -www.journalijar.com
INTERNATIONAL JOURNAL OF ADVANCED RESEARCH (IJAR)
Article DOI: 10.21474/IJAR01/2309
DOI URL: http://dx.doi.org/10.21474/IJAR01/2309

RESEARCH ARTICLE

DISCRETE EVENT MODELING AND SIMULATION OF A MODIFIED DYNAMIC EN-ROUTE FILTERING TO IMPROVE THE ENERGY EFFICIENCY IN WIRELESS SENSOR NETWORKS.

Dong Jin Park[1] and *Tae Ho Cho[2].
College of Software, SungkyunkwanUniversity, Suwon 16419, Republic of Korea.

Manuscript Info

Manuscript History

Received: 29 September 2016
Final Accepted: 30 October 2016
Published: November 2016

Key words:-
Wireless sensor networks,
Modeling and simulation,
Dynamic en-route filtering scheme

Abstract

Because wireless sensor networks are exposed to various attacks, security protocols are essential. Among the representative attacks and countermeasures, there are a false report injection attacks and a dynamic en-route filtering scheme. Dynamic en-route filtering scheme effectively defends using message authentication code, but there is room for improvement in terms of energy. So, we use a key re-distribution algorithm of dynamic en-route filtering to improve energy efficiency. In this paper, we propose a discrete event modeling and simulation of modified dynamic en-route filtering. More accurate measurement is possible through the same system designed as the actual wireless sensor network. The experimental results demonstrate the validity of our scheme, showing an energy efficiency of up to 42.45% and a filtering capacity of up to 18.71% compared to the existing scheme.

Introduction:-

Wireless sensor networks (WSNs) have been used in various applications that require real-time monitoring [1-3]. WSNs consist of numerous small sensor nodes for detecting an event, as well as a base station (BS) that is used to collect detected event data and send it to a user. Typically, large-scale WSNs are comprised of thousands of sensor nodes. Additionally, several cluster heads (CHs) in a sensor field are divided into cluster units [4]. A CH receives data about an event from member nodes belonging to its own cluster. The event data are then converted into a report and forwarded to the next node.The sensor node of a low-cost product has resource constraints, such as limited processing power, memory capacity, and energy, and it also communicates wirelessly in an open environment [5]. Therefore, the node is easily exposed to various attacks, such as environmental damage, as well as physical capture or false report injection attacks and selective forwarding attacks from malicious attackers. Hence, security research that considers the limited resources of these nodes is needed and has been investigated recently [6-8].

In a WSN, an attacker can attempt to cause false alarms at the BS by injecting false event data and depleting the limited energy resources of the nodes by forcing them to forward false reports. Therefore, it is very important to detect a false report injection attack as early as possible. To detect such attacks, a dynamic en-route filtering (DEF) scheme was proposed by Yu and Guan [9]. This scheme can detect a false report in the transfer process using the authentication key and secret keys. In DEF, however, when a network is exposed to a continuous attack from adversaries, unnecessary energy loss occurs. DEF has no suitable measures to address this problem because this method does not consider the current attack situation.

Corresponding Author:-Tae Ho Cho.
Address:-College of Software, Sungkyunkwan University, Suwon 16419, Republic of Korea.

In this paper, we use the key re-distribution algorithm of the dynamic en-route filtering scheme[10].For the experiment, we apply the discrete event modeling and simulation. A proposed model represents a simplified version of WSN and its structure. The model is built considering the conditions of experimentation of the WSN system, including the work conditions of the modified DEF system.Experimental results show that compared to the existing scheme, our scheme achieves energy savings of up to 42.45% and improves the detection performance by up to 18.71%.

The rest of this paper is organized as follows. We introduce related work in Section 2 and then present the problem statement of the existing scheme in Section 3. In Section 4, the proposed scheme is provided and the performance evaluation is discussed in Section 5. Finally, we summarize our conclusions and future work in Section 6.

Related Works:-

False report injection attack:-

A false report injection attack injects a false report into the network to deceive the BS and deplete the limited energy resources of the network [11]. Figure 1 shows an overview of a false report injection attack occurring through nodes that are compromised in a WSN. In the example, the attacker obtains control by seizing several nodes in the field and then creates and injects a false report (as opposed to real event data) using the compromised nodes. The false report is delivered to the BS, without being detected by the intermediate nodes, along the normal routing path. The BS that receives the false report confirms the content of the report and generates false alarms. This disrupts the appropriate measures that are used to respond to legitimate report content and results in financial- and energy-related losses. In addition, the normal nodes on the forwarding path suffer unnecessary energy losses when they deliver an attacker's malicious false reports. Because such attacks occur constantly, the lifetime of the entire network is shortened. In addition to false report injection attacks, there are also sinkhole attacks and selective forwarding attacks. The damage range of a sinkhole attack is regional and the damage range of a selective attack affects some of the nodes on a single path. However, the damage range of a false report injection attack is much larger and covers all nodes on the path and the BS. Thus, this attack is more serious in WSNs, which have limited energy.

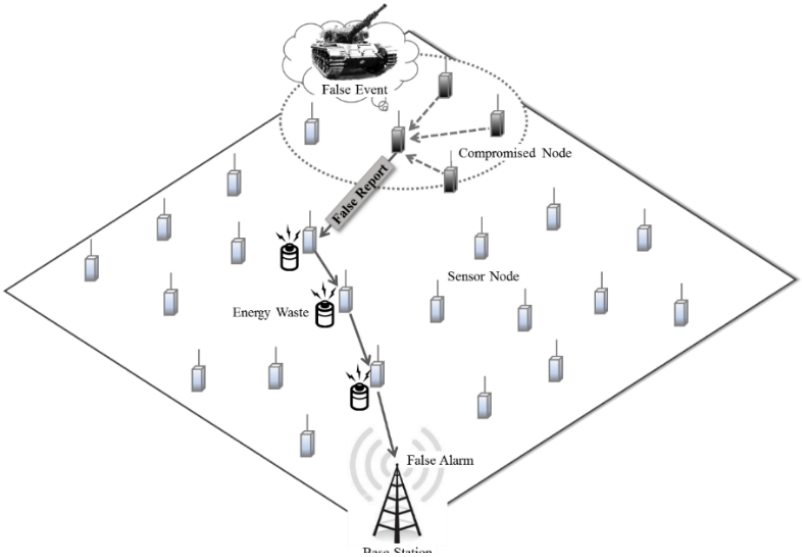

Figure 1:- False report injection attack in a WSN

Dynamic en-route filtering (DEF):-

DEF defends against false report injection attacks in cluster-based WSNs. The WSN divides the distributed nodes into clusters and each cluster elects a cluster head (CH) as a representative node. The member nodes of the cluster send event information to the cluster head. In this scheme, each node is preloaded with an authentication key and secret keys at the BS and then each node is distributed to the field. Next, each CH collects the authentication keys from their member nodes and aggregates them into a message. This message is passed on to the next node along the routing path. The forwarding node that received the message also loads the authentication key into its memory from the message. This authentication key is used to determine whether or not the event report is false. DEF has three operation phases, as shown in Figure 2.

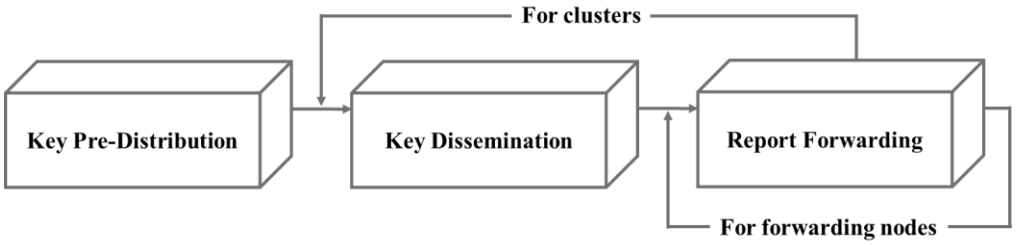

Figure 2:- Three-step operation of DEF

The key pre-distribution phase is executed only once at the BS when the sensor node is initially deployed to the field. Each node loads the different seed keys, creates an authentication key using the common hash function from the seed key, and then configures the hash chain. At this time, if the node has sufficient memory, the node generates all of the authentication keys. Otherwise, the authentication key is generated whenever it is needed. In addition, a node has l+1 secret keys. The l keys are called the y-keys and are randomly chosen from a y-key pool of size v. The last key is called the z-key and is randomly chosen from a z-key pool of size w. The z-key is distinguishable from other nodes.

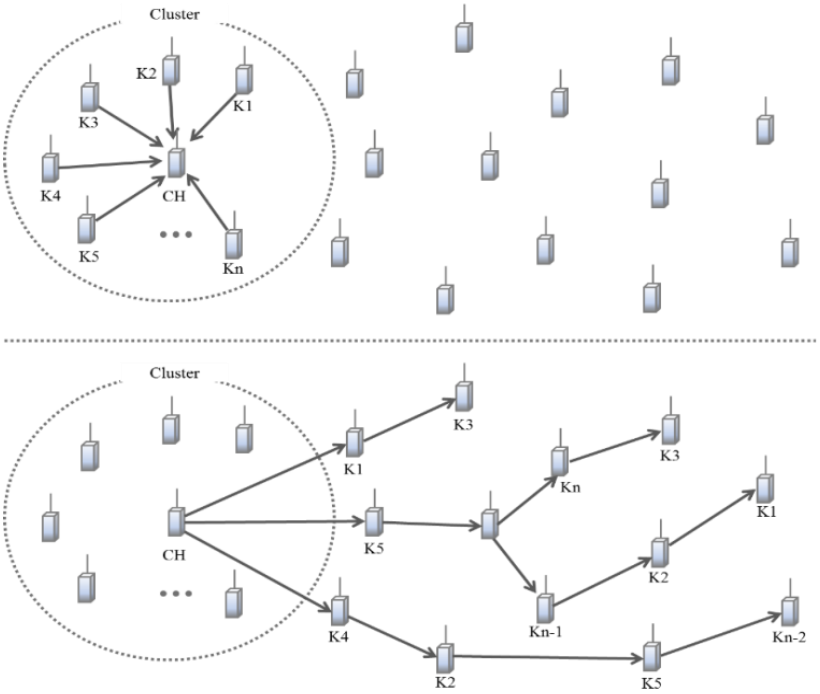

Figure 3:- Key dissemination phase of DEF

Figure 3 shows the key dissemination phases. This phase has a four-step procedure. First, each node in the cluster encrypts the current authentication key using one of the secret keys. The authentication key message structure of the node is shown in Equation (1).

$$Auth(v_i) = \{v_i,\ j_i,\ id(y_1^{v_i}),\ \{id(y_1^{v_i}),\ k_{j_i}^{v_i}\}_{y_1^{v_i}},\ \ldots, id(y_l^{v_i}),\ \{id(y_l^{v_i}),\ k_{j_i}^{v_i}\}_{y_l^{v_i}},\ id(z^{v_i}),\ \{id(z^{v_i}),\ k_{j_i}^{v_i}\}_{z^{v_i}}\} \quad (1)$$

In (1), j_i is the index of the current authentication key and $j_i = 1$ denotes the first dissemination. Additionally, $id(y_1^{v_i})$ is the index of $y_1^{v_i}$ in the y-key pool and $\{*\}_{y_1^{v_i}}$ is encrypted using the key $y_1^{v_i}$. Then, the CH collects the

authentication message from all nodes in the cluster that they belong to and aggregates them into message K(n), shown in Equation (2). In (2), $v_1, \dots, v_n$ are the nodes in the cluster.

$$K(n) = \{ Auth(v_1), \dots, Auth(v_n) \} \qquad (2)$$

Next, the CH selects q (q>1) forwarding nodes from neighbor nodes and forwards them to K(n). The reason for selecting q is so that K(n) can be delivered to another node without necessitating key re-dissemination of K(n) if the next node is compromised. When a forwarding node receives K(n), the node performs the following steps.

1) It checks whether K(n) has at least t distinct z-key indexes. If not, K(n) is determined to be false and is dropped.
2) It checks the secret key indexes of K(n). If it has the same key index, it decrypts the message and stores the authentication key in its own memory.
3) It drops K(n) if K(n) has already been forwarded by h_{max} . Each node repeats the above operation until K(n) has been forwarded by h_{max} or until K(n) has reached the BS.

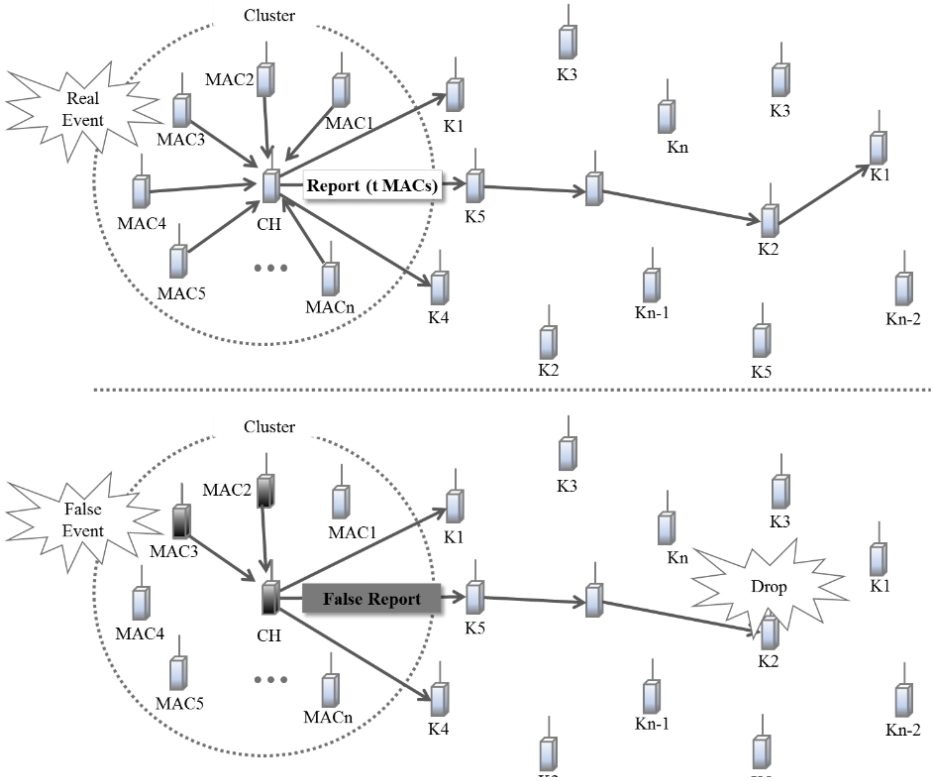

Figure 4:- Report forwarding phase of DEF

Figure4 shows the report forwarding phase. In this phase, the sensor node detects an event, generates a message authentication code (MAC) [12] for the event information using a new authentication key, and forwards the MAC to its CH. In the CH, the number of sensor nodes that participate in generating a report (t) is pre-determined before deployment and the CH forwards the report to q neighbor nodes. When the next node sends an ok message to indicate that the report has been received correctly, the CH exposes the authentication key of the sensor nodes and verifies the report. It then informs the next hop node of the verified result. This operation is repeated until the report arrives at the BS or until the report is determined to be false.

Discrete event system specification (DEVS):-

The DEVS formalism[13] introduced by Zeigler provides a means of specifying a mathematical object that is referred to as a system. Basically, a system has a time base, inputs, states, and outputs, as well as functions that determine subsequent states and outputs given current states and inputs. DEVS derives an accurate result through a realistic test using the time value. DEVS has the following characteristics.

1) Easy reusability of the model
2) Contains information on the each port and function for the external and internal event.
3) Contains information on the time and state values of the model for the next event
4) Time values are used to control the function timing

DEVS is largely composed of an atomic model and a coupled model, which combines atomic models. The atomic model employs irreducible model definitions that specify the behavior for any modeled entity. The structure of the atomic model is defined by Equation (3).

$$M = \langle X, S, Y, \delta_{int}, \delta_{ext}, \lambda, ta \rangle \quad (3)$$

Here, X is the set of external input event types, S is the sequential state set, Y is the set of external event types generated as the output, δ_{int} (δ_{ext}) is the internal (external) transition function that dictates the state transitions caused by internal (external input) events, λ is the output function that generates external events at the output, and ta is the time-advance function.
The coupled model is an aggregation/composition of two or more atomic models connected by explicit couplings. The structure of the coupled model is defined by Equation (4).

$$DN = \langle D, \{M_i\}, \{I_i\}, \{Z_{i,j}\}, select \rangle \quad (4)$$

Here, D is a set of component names for each I in D; M_i is a component basic model;$\{I_i\}$ is the set of influences of Ifor each j, $Z_{i,j}$ is a function, specifically the i-to-j output translation; and select is a function used as the tie-breaking selector. Detailed descriptions about DEVS simulator, experimental frame and of both atomic and coupled models can be found in [13, 14].

Problem Statement:-
This section provides background information about DEF and introduces improvements. DEF is an effective scheme for defending against false report injection attacks in a dynamic environment; however, it has several problems in terms of its energy efficiency.

- DEF quickly detects a false report by using keys that are distributed randomly. However, when a DEF-based network is exposed to continuous attacks, unnecessary energy consumption results from forwarding a report detailing a non-existent event; this occurs until the false report is detected.
- In DEF, the key-dissemination phase that redistributes the entire key may be performed arbitrarily in order to reduce the energy loss that is caused by continuous attacks, but it is very important to determine when this phase should be executed because it consumes a lot of energy. Also, even if the conditions for the entire key distribution phase execution are met (topology change) and the phase is executed, energy loss can still occur. This is caused by the fact that the defence scheme does not grasp or consider the attack situation.

To address these problems, we propose the following improvements.

- In order to understand that there is a constant attack on the current network, the forwarding node counts the number of detected false reports corresponding to each source CH. If the count value exceeds the threshold value, the forwarding node requests key re-distribution for the BS. After key re-distribution, if a false report is generated using the same false key, the amount of unnecessary energy consumption is minimized by detecting this at the next hop node.
- Because an entire key re-distribution process consumes a lot of energy, it is executed locally based on the network state. Also, the BS specifies several suspected nodes as potential compromised nodes, allowing the BS to be free from continuous attacks by eliminating suspected nodes along the routing path.

In this paper, we propose an energy-efficient key re-distribution scheme that considers the state of the network for early detection of false reports when exposed to constant attacks. Through this scheme, we expect to minimize unnecessary energy consumption that is associated with the existing scheme.

Proposed Scheme:-

Overview:-

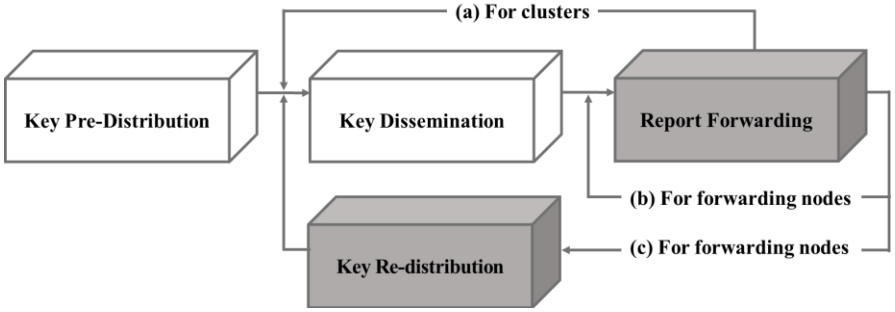

Figure 5:- Operation phases of the proposed scheme

In DEF, there is no change in the secret key that is loaded in the key pre-distribution phase. Using this property, the proposed scheme modified the report forwarding phase of DEF and added the key re-distribution phase. Figure 5 shows the operation procedure of this proposed scheme. When the false report count value is less than the threshold, Fig. 5(a) operates in the same manner as DEF at the report forwarding phase. In the report forwarding phase, Fig. 5(b) shows that if the report is detected to be false, then the forwarding node identifies a source CH and increases the false report count value. After this point, the behavior is the same as the existing DEF. In Fig. 5(c), the key re-distribution request message is sent to the BS when the count value is more than the threshold value of the report forwarding phase.

Assumptions:-

This paper has the following assumptions. Since the key pre-distribution phase is operated before dissemination in the field, the BS knows the key information for all of the nodes. Since the BS has all of the keys (through the global pool key), when a false report is delivered to the BS the BS knows that the false report exists. The BS also knows the node energy along the re-distribution request message path. There is no change in the network topology. The forwarding node has space to store the number of false reports that is detected. When the node stores the authentication key from K(n) in its memory, it also stores the decrypted secret key. The structure of the key re-distribution request message sent to the BS from the forwarding node is defined as follows:

$$FN \rightarrow BS : S \| D \| CH_{ID} \| KI_S \| MAC$$

Here, FN is the forwarding node, S is the source, and D is the destination. $CH_{ID}$is the identity of the source CH that generated the false report. $KI_S$is the index of the secret key that corresponds to the authentication key used for detecting a false report. The notation || indicates consecutive concatenation.

Detailed procedure:-

This section describes the operation of the proposed scheme in detail. Figure 6 shows the entire execution process of the proposed scheme. Compromised nodes in the cluster and CH generate a false report and then send it to neighbor nodes (Fig. 6[a]). The next forwarding node, u_i, which received the report, verifies the report. It cannot detect the false report and sends the report to the next neighbor nodes (Fig. 6[b]). $u_{i+k}$of the forwarding node on the path detects the false report and drops it. At this time, u_{i+k} knows the source CH through the received report and stores the false report detection count value that corresponds to the source CH (Fig. 6[c]). If the attack occurs continuously and the count value exceeds the threshold of a certain CH, the forwarding node encrypts the message of the key re-distribution request and sends it to the BS. Because the BS that received the message knows all of the keys, it decrypts this message and stores the $CH_{ID}$and KI_S. The BS selects a node with the same secret key as the KI_S and checks the source cluster in the CH_{ID} value. The reason for this is that a compromised node also has the same secret key that is used to load the authentication key that is used for detection. The BS executes the key re-distribution scheme by collecting the authentication keys of suspected nodes (Fig. 6[d]). The aggregated authentication key message is encrypted and transmitted to the next hop-forwarding node, u_i, of the source CH. This is because there are authentication keys that are stolen from the attacker among the collected authentication keys. The u_j that received the message decrypts the message and stores the authentication keys in its memory. If attackers try to inject a false report using the same stolen key again, the false report is detected quickly at the next node, u_j.

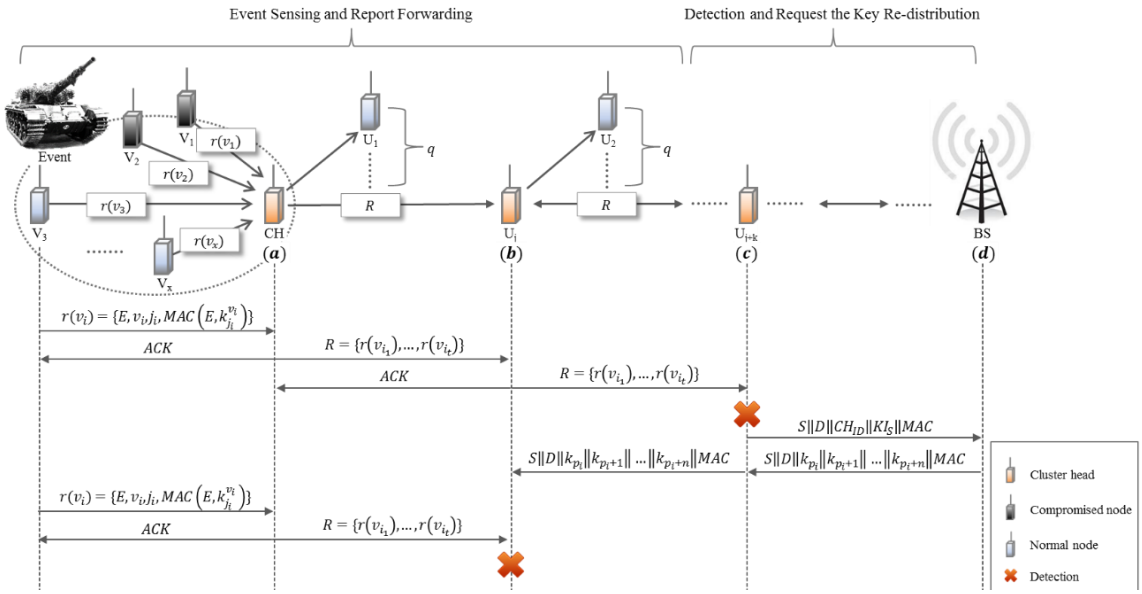

Figure 6:- Detailed execution phases of the proposed scheme

System modeling:-

An experimental frame module is a coupled model that, when coupled to a model, generates input external events, monitors its running, and processes its output. To experiment with the model WSN, we couple it with the experimental frame component EF to form the digraph model EF-WSN. The EF-WSN is described below.

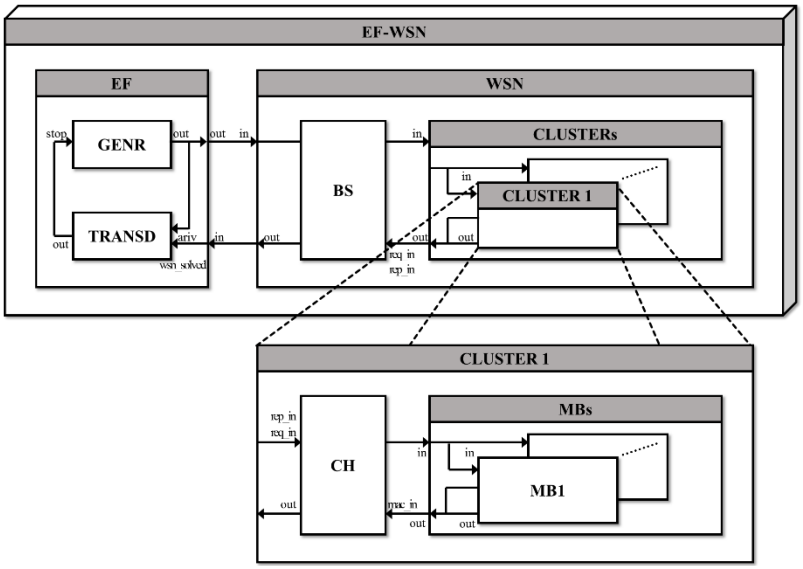

Figure 7:- System structure of the proposed scheme

Figure 7 shows the system structure of the proposed scheme. The system structure is the same as the actual environment of the WSN.

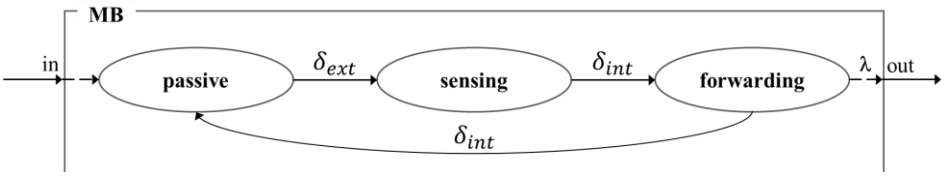

Figure 8:- State diagram of the MB model

Figure 8 shows the state diagram of the MB model. The MB model has three states. The passive state is the wait state. When a message arrives at the in port, it changes from the passive state to the sensing state, which detects the event by an external transition function. The sensing state changes to a forwarding state by an internal transition function. The output function generates an output value (MACs) and sends it to the CH. Finally, it returns to the passive state by an internal transition function.

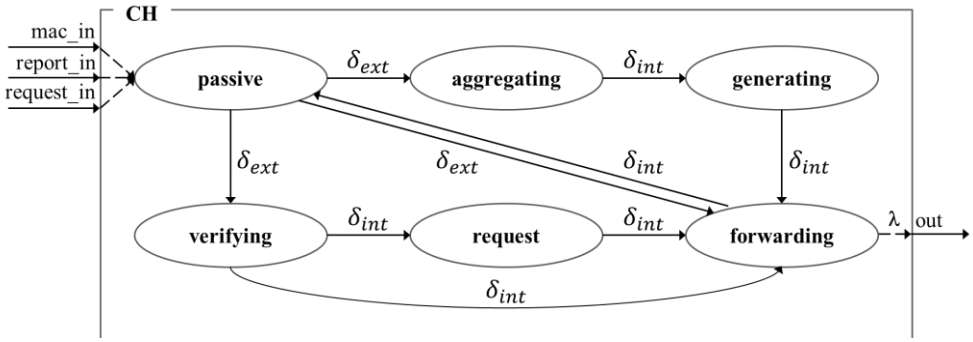

Figure 9:- State diagram of the CH model

Figure 9 shows the state diagram of the CH model. The CH model has six states. When a message arrives at the mac_in port, it changes to an aggregating state by an external transition function and changes to a generating state by an internal transition function. Then, it changes to the forwarding state by an internal transition function. When the CH receives the MAC from the MB, the process generates the report type and sends it to the next forwarding node. When a message arrives at the report_in port, it changes to a verifying state by an external transition. At this time, if the count value of the forwarding node exceeds the threshold value, it changes to a request state; otherwise, it changes to a forwarding state by an internal transition function. This is the process that requests execution of the proposed method. When a message arrives at the request_in port, it changes to a forwarding state. This is the process that forwards the request message to the BS.

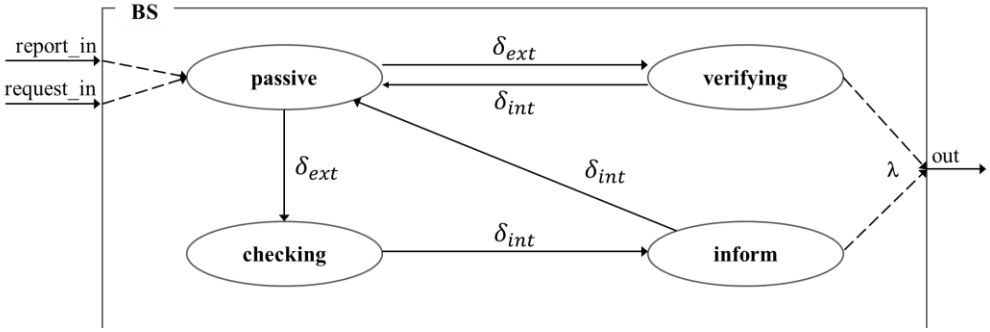

Figure 10:- State diagram of the BS model

Figure 10 shows the state diagram of the BS model. The BS model has four states. When a message arrives at the report_in port, it changes to a verifying state by an external transition function. This corresponds to report verification and generates an output value with the output function. When a message arrives at the request_in port, it changes to a checking state by an external transition function. In this process, the BS checks the suspected nodes by utilizing the message information of key re-distribution. It changes to an inform state by an internal transition function. This is the process that executes key re-distribution.

Performance Evaluation:-

Experimental environment:-

The experimental environment is configured as follows: 1,000 nodes are randomly disseminated at sensor fields, which are 1,000 x 1,000 m^2 in size. Among these nodes, 900 nodes are normal nodes and 100 nodes are CHs. The BS is positioned in the bottom middle (x, y = 500, 1000) of the sensor field. Each node is randomly loaded with two y-keys and one z-key in each key pool of size w. The energy required to transmit one byte is 16.25μJ and the energy required to receive one byte is 12.5μJ. The energy required to verify an MAC is 75μJ and the energy required to

generate an MAC is 15μJ[15]. The size of a report is 36bytes and an MAC is one byte. Experiments with the proposed scheme are conducted using a proper threshold, (α), which is referenced when transmitting the message to the BS at a forwarding node. Through simulation, we select an appropriate threshold (α = 20).False report injection attacks occur depending on the FTR of the 10 clusters that are selected. Target nodes that can be compromised are normal nodes and CHs in the selected cluster. No packet loss occurs in the communication of the node and 1,000 events are randomly run at the selected clusters.

Experimental results:-

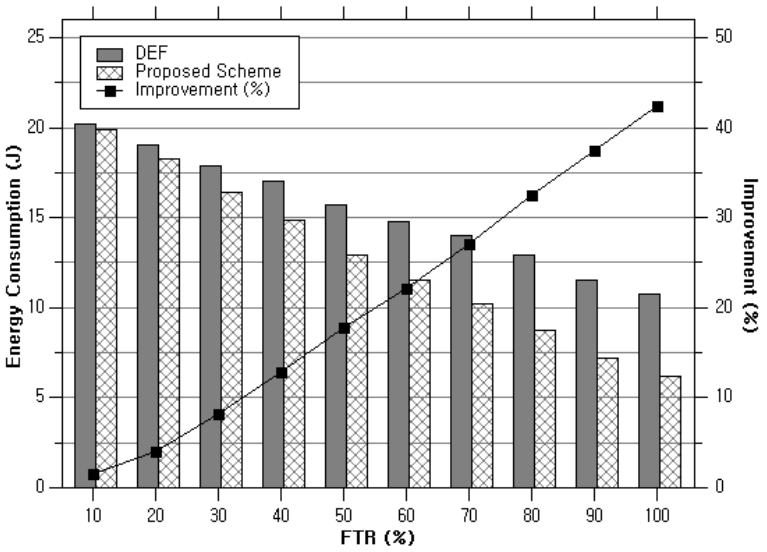

Figure 11. Energy consumption versus the FTR

Figure 11 shows the total energy consumption of the network according to the FTR. In both schemes, the number of false reports that were detected previously at the BS increased as a function of the FTR; therefore, the total energy consumption is reduced. The energy difference between the two schemes is due to the reduced number of hop counts of false reports, which is caused by the proposed key re-distribution phase. Thus, the unnecessary energy loss is reduced. As the FTR increases, we see that the difference in the energy consumption of both schemes increases. The reason for this is that the number of false reports that are detected at the early hop stage is much greater than the existing scheme.

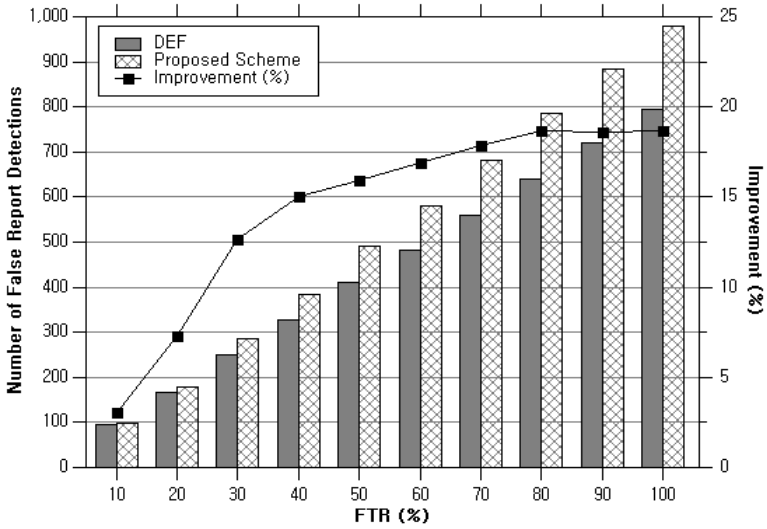

Figure 12:- Number of dropped false reports versus the FTR

Figure 12 shows the number of false reports that were detected. Both schemes have greater than 80% performance. The proposed scheme seems to show a slight improvement in the performance. The reason for this is as follows. When a false report is not detected at the intermediary forwarding node and reaches the BS, the BS counts the false report according to the source CH. If the count value exceeds the threshold, the BS executes key re-distribution in the next node of the source cluster (similar to a forwarding node). The detection power is improved because the false report that was forwarded until the BS is dropped at the next node of the source CH.

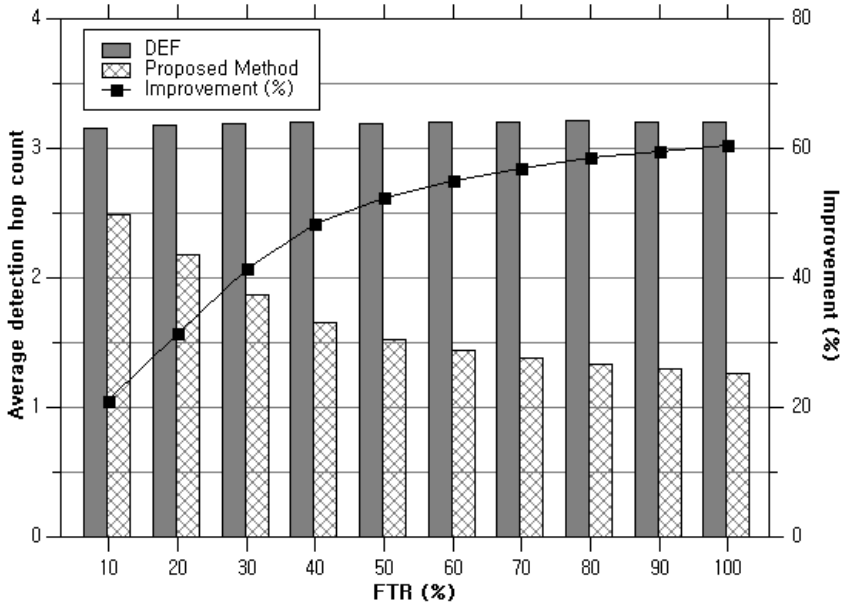

Figure 13:- Average hop count of the false report versus the FTR

Figure 13 shows the average hop count of the false report. The difference between the two schemes is due to the earlier detection of the hop compared to the DEF as well as to the detection of a false report that was not detected in the existing scheme at the next node of the source CH. The average hop count of a false report for DEF, however, is constant. This is the case because DEF randomly distributes the key and the proposed key re-distribution is executed. Therefore, the average hop counts of the proposed schemes differ.Table 1 shows these results quantitatively.

Table 1:- Experimental results quantitatively

FTR (%)	Energy consumption (J)		Efficiency (%)	Detecting Performance (EA)		Efficiency (%)
	Existing scheme	Proposed scheme		Existing scheme	Proposed scheme	
10	20.20	19.89	1.51	96	99	3.03
20	19.05	18.28	4.05	166	179	7.26
30	17.90	16.43	8.18	249	285	12.63
40	17.02	14.84	12.80	327	385	15.06
50	15.73	12.94	17.74	412	490	15.92
60	14.78	11.51	22.13	483	581	16.87
70	14.02	10.21	27.14	560	682	17.89
80	12.93	8.73	32.48	640	787	18.68
90	11.51	7.20	37.42	719	883	18.57
100	10.74	6.18	42.45	795	978	18.71

Conclusions:-

Wireless sensor networks are exposed to impairments such as false report injection attacks. DEF can effectively defend against this attack. We designed a modified DEF based WSN model using a DEVS formalism and we simulated the key re-distribution algorithms of DEF. As shown in the experimental results, we achieved the saving an energy and increasing a detection performance. Through this model, we reflects the operation of the actual WSN, resulting in more accurate experimental values.

Acknowledgements:-

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (No. NRF-2015R1D1A1A01059484)

References:-

1. Y. Wang, G. Attebury and B. Ramamurthy, "A survey of security issues in wireless sensor networks," 2006.
2. S. S. Iyengar and R. R. Brooks, Distributed Sensor Networks: Sensor Networking and Applications. CRC press, 2016.
3. J. Yick, B. Mukherjee and D. Ghosal, "Wireless sensor network survey," Computer Networks, vol. 52, pp. 2292-2330, 2008.
4. S. H. Lee, S. Lee, H. Song and H. S. Lee, "Wireless sensor network design for tactical military applications: Remote large-scale environments," in MILCOM 2009-2009 IEEE Military Communications Conference, 2009, pp. 1-7.
5. V. C. Gungor and G. P. Hancke, "Industrial wireless sensor networks: Challenges, design principles, and technical approaches," IEEE Trans. Ind. Electron., vol. 56, pp. 4258-4265, 2009.
6. N. A. Pantazis, S. A. Nikolidakis and D. D. Vergados, "Energy-efficient routing protocols in wireless sensor networks: A survey," IEEE Communications Surveys & Tutorials, vol. 15, pp. 551-591, 2013.
7. S. M. Nam and T. H. Cho, "A fuzzy rule-based path configuration method for LEAP in sensor networks," Ad Hoc Networks, vol. 31, pp. 63-79, 2015.
8. M. Akram and T. H. Cho, "Energy efficient fuzzy adaptive selection of verification nodes in wireless sensor networks," Ad Hoc Networks, 2016.
9. Z. Yu and Y. Guan, "A dynamic en-route filtering scheme for data reporting in wireless sensor networks," IEEE/ACM Transactions on Networking (ToN), vol. 18, pp. 150-163, 2010.
10. D. J. Park and T. H. Cho, "Key Re-distribution Scheme of Dynamic Filtering Utilizing Attack Information for Improving Energy Efficiency in WSNs," Journal of Korean Institute of Intelligent Systems, vol. 26, pp. 113-119, April, 2016.
11. S. Jeba and B. Paramasivan, "False data injection attack and its countermeasures in wireless sensor networks," European Journal of Scientific Research, vol. 82, pp. 248-257, 2012.
12. W. Ye, J. Heidemann and D. Estrin, "An energy-efficient MAC protocol for wireless sensor networks," in INFOCOM 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE, 2002, pp. 1567-1576.
13. B. P. Zeigler, Object-Oriented Simulation with Hierarchical, Modular Models: Intelligent Agents and Endomorphic Systems. Academic press, 2014.
14. S. Mittal, J. L. Risco and B. P. Zeigler, "DEVS-based simulation web services for net-centric T&E," in Proceedings of the 2007 Summer Computer Simulation Conference, 2007, pp. 357-366.
15. F. Ye, H. Luo, S. Lu and L. Zhang, "Statistical en-route filtering of injected false data in sensor networks," IEEE J. Select. Areas Commun., vol. 23, pp. 839-850, 2005.